

Oracle® Application Server Portal

開発者ガイド

10g (9.0.4)

部品番号 : B14107-01

2004 年 7 月

Oracle Application Server Portal 開発者ガイド, 10g (9.0.4)

部品番号 : B14107-01

原本名 : Oracle Application Server Portal Developer's Guide, 10g (9.0.4)

原本部品番号 : B13922-01

原本協力者 : Joan Carter, Tugdual Grall, Helen Grembowicz, Peter Moskovits, Sergei Pecherskyy, Frank Rovitto, Ingrid Snedecor, Sue Vickers, Vanessa Wang, Gareth Bryan, Pankaj Mittal, Lei Oh, Alistair Wilson

Copyright © 2004 Oracle Corporation. All rights reserved.

制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

このドキュメントの情報は、予告なしに変更される場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S.

Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation, and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションへの用途を目的としておりません。このプログラムをかかるとして使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性（**redundancy**）、その他の対策を講じることは使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle は Oracle Corporation およびその関連会社の登録商標です。その他の名称は、Oracle Corporation または各社が所有する商標または登録商標です。

目次

はじめに	xiii
対象読者	xiv
このマニュアルの構成	xiv
関連ドキュメント	xvi
表記規則	xvi
推奨ブラウザ	xvii

第 I 部 ポートレットの概要

1 ポートレットの概要

ポータル開発の概要	1-2
ポートレットの概要	1-2
ポートレットの構造	1-5
ポートレット・リソース	1-6
複雑な設定が不要なポートレット	1-7
事前に構築されたポートレットのその他のソース	1-8
Web クリップング	1-9
OmniPortlet	1-13
ポートレット・ビルダー	1-14
プログラムのなポートレット	1-15
使用するツールの決定	1-16
まとめ	1-16

第II部 ポートレット・テクノロジー

2 ポートレット・テクノロジーのマトリックス

ポートレット・テクノロジーのマトリックス	2-2
一般的な適性	2-5
Web クリップング	2-5
Web クリップングを使用して構築できるポートレットの例	2-5
OmniPortlet	2-6
OmniPortlet で作成できるポートレットの例	2-6
Java ポートレット	2-6
Java を使用して構築できるポートレットの例	2-6
ポートレット・ビルダー	2-7
ポートレット・ビルダーを使用して構築できるポートレットの例	2-7
PL/SQL ポートレット	2-7
PL/SQL を使用して構築できるポートレットの例	2-7
必要な専門知識	2-8
Web クリップング	2-8
OmniPortlet	2-8
Java ポートレット	2-9
ポートレット・ビルダー	2-9
PL/SQL ポートレット	2-9
デプロイ・タイプ	2-10
Web プロバイダ	2-11
WSRP プロバイダ	2-12
データベース・プロバイダ	2-12
プロバイダ・アーキテクチャ	2-14
キャッシュ・スタイル	2-16
Web クリップング、OmniPortlet およびポートレット・ビルダー	2-17
Java ポートレット	2-17
PL/SQL ポートレット	2-17
開発ツール	2-18
Web クリップング、OmniPortlet およびポートレット・ビルダー	2-18
Java ポートレット	2-18
PL/SQL ポートレット	2-18

ポートレット作成スタイル	2-19
OmniPortlet および Web クリップング	2-20
Java ポートレット	2-20
ポートレット・ビルダー	2-20
PL/SQL ポートレット	2-20
ユーザー・インタフェースの柔軟性	2-21
Web クリップング	2-21
OmniPortlet およびポートレット・ビルダー	2-21
Java ポートレットおよび PL/SQL ポートレット	2-21
Web サイトからコンテンツを取得する機能	2-22
Web クリップング	2-22
OmniPortlet	2-22
Java ポートレット	2-22
PL/SQL ポートレット	2-22
コンテンツをインラインでレンダリングする機能	2-23
Web クリップング	2-23
OmniPortlet	2-23
Java ポートレット	2-24
ポートレット・ビルダー	2-24
PL/SQL ポートレット	2-24
チャート作成機能	2-25
Web クリップング	2-25
OmniPortlet	2-25
Java ポートレット	2-25
ポートレット・ビルダー	2-25
PL/SQL ポートレット	2-25
パブリック・ポートレット・パラメータのサポート	2-26
プライベート・ポートレット・パラメータのサポート	2-27
OmniPortlet、Web クリップングおよびポートレット・ビルダー	2-27
Java ポートレットおよび PL/SQL ポートレット	2-27
イベントのサポート	2-28
Web クリップング、OmniPortlet および Java ポートレット	2-28
ポートレット・ビルダーおよび PL/SQL ポートレット	2-28
ユーザー権限に基づいてポートレットの表示 / 非表示を切り替える機能	2-28
Web クリップングおよび OmniPortlet	2-28
Java ポートレット	2-29

ポートレット・ビルダー	2-29
PL/SQL ポートレット	2-29
多言語サポート	2-29
Web クリップング、OmniPortlet、Java ポートレットおよび PL/SQL ポートレット	2-29
ポートレット・ビルダー	2-29
ページ区切りサポート	2-30
OmniPortlet	2-30
Java ポートレットおよび PL/SQL ポートレット	2-30
ポートレット・ビルダー	2-30
シングル・サインオンおよび外部アプリケーション統合	2-30
Web クリップング	2-30
OmniPortlet	2-30
Java ポートレット	2-31
PL/SQL ポートレット	2-31

第 III 部 ポートレットの構築

3 OmniPortlet を使用したポートレットの構築

OmniPortlet の概要	3-3
タイプ	3-4
ソース	3-5
スプレッドシート	3-6
SQL	3-7
XML	3-8
Web サービス	3-9
Web ページ	3-10
フィルタ	3-12
表示	3-13
レイアウト	3-14
表レイアウト	3-15
チャート・レイアウト	3-16
ニュース・レイアウト	3-19
箇条書きレイアウト	3-21
フォーム・レイアウト	3-22
イベント	3-24

パラメータとイベント	3-25
ポートレット・パラメータとイベント	3-25
ページ・パラメータとイベント	3-26
OmniPortlet の使用	3-27
ポータル・ページへの OmniPortlet の追加	3-27
SQL データソースに基づくポートレットの定義	3-28
Web サービスに基づくポートレットの定義	3-31
既存の Web ページに基づくポートレットの定義	3-34
既存の OmniPortlet のレイアウトの変更	3-41
パラメータとイベントの使用	3-43
既存の OmniPortlet へのパラメータの追加	3-44
既存の OmniPortlet へのイベントの追加	3-46
ページ上でのポートレット・パラメータとイベントの関連付け	3-47
まとめ	3-50

4 Web クリップングを使用したコンテンツ・ベースのポートレットの構築

ページへの Web クリップング・ポートレットの追加	4-3
Web クリップング・ポートレットに表示する Web ページのセクションの選択	4-5
Web クリップング・ポートレットのプロパティの設定	4-10
ポートレットへの外部アプリケーションの統合	4-11
例：ページへの Portal Center 情報の追加	4-13
演習：ページへの Web クリップング・ポートレットの追加	4-13
演習：Portal Center でのクリップングの選択	4-14
演習：Web クリップング・ポートレットのプロパティの設定	4-16
Web クリップングに関する現在の制限	4-20
Web クリップングのトラブルシューティング	4-21
テスト・ページを使用したプロバイダの状態の確認	4-21
接続に関する問題の解決	4-21
プロキシ・サーバーの構成	4-23
プロキシ認証	4-23
ロギング・レベルの設定	4-24

5 Java ポートレットの構築

Java ポートレット作成のガイドライン	5-2
共有画面モード (JPS の場合は表示モード)	5-3
ポートレットのレンダリングに関する HTML ガイドライン	5-3
ポートレットのレンダリングに関する Cascading Style Sheet ガイドライン	5-4
編集モード (JPS および OracleAS Portal)	5-5
編集モード・オプションに関するガイドライン	5-5
編集モードのボタンに関するガイドライン	5-5
カスタマイズ値のレンダリングに関するガイドライン	5-6
デフォルト編集モード (JPS および OracleAS Portal)	5-6
デフォルト編集モード・オプションに関するガイドライン	5-7
デフォルト編集モードのボタンに関するガイドライン	5-7
カスタマイズ値のレンダリングに関するガイドライン	5-7
プレビュー・モード (JPS および OracleAS Portal)	5-8
プレビュー・モードに関するガイドライン	5-8
全画面モード (OracleAS Portal)	5-9
全画面モードに関するガイドライン	5-9
ヘルプ・モード (JPS および OracleAS Portal)	5-9
ヘルプ・モードに関するガイドライン	5-9
情報モード (JPS および OracleAS Portal)	5-10
情報モードに関するガイドライン	5-10
リンク・モード (OracleAS Portal)	5-10
リンク・モードに関するガイドライン	5-10
Java ポートレット仕様および WSRP の概要	5-11
WSRP と JPS の関連	5-12
Oracle JDeveloper を使用した JPS 準拠のポートレットの構築	5-14
Oracle JDeveloper ポータル・アドインのインストール	5-14
JPS 準拠のポートレットの構築	5-15
ポートレットの作成	5-15
ポートレット・ロジックの追加	5-26
アプリケーション・サーバーへのポートレットのデプロイ	5-26
Oracle Application Server Containers for J2EE への接続の確立	5-26
WAR ファイルのデプロイ	5-28
ポートレットの登録と表示	5-29
ローカルの OracleAS Portal インスタンスへの登録	5-29
portal.standards.com への登録	5-34
ポートレットの追加	5-35

Oracle JDeveloper を使用した PDK-Java ポートレットの構築	5-35
Oracle JDeveloper ポータル・アドインのインストール	5-35
PDK-Java ポートレットの構築	5-35
ポートレットとプロバイダの作成	5-36
ポートレット・ロジックの追加	5-44
ポートレットとプロバイダの検証	5-45
アプリケーション・サーバーへのデプロイ	5-46
Oracle Application Server Containers for J2EE への接続の確立	5-47
WAR ファイルのデプロイ	5-48
ポートレットの登録と表示	5-50
レンダリング・モードの追加	5-56
前提	5-56
追加の表示モードの実装	5-56
XML プロバイダ定義の更新	5-57
ポートレットの表示	5-57
ポートレットのカスタマイズ	5-59
前提	5-61
「編集」 ページおよび「デフォルトの編集」 ページでのカスタマイズの実装	5-61
生成されたコードの確認	5-61
生成されたコードの変更	5-62
「表示」 ページでのカスタマイズの実装	5-64
XML プロバイダ定義内の設定情報	5-65
ポートレットの表示	5-65
パラメータの引渡しとイベントの送信	5-66
前提	5-66
ポートレットへのパラメータの追加	5-66
イベントの送信	5-69
イベント・ポートレットの作成	5-69
セッション情報へのアクセス	5-74
前提	5-75
セッション記憶域の実装	5-75
ポートレットの表示	5-78
ポートレットのセキュリティの実装	5-78
前提	5-78
ポートレットのセキュリティ機能	5-78
認証	5-78
認可	5-78
通信セキュリティ	5-79

シングル・サインオン	5-80
パートナ・アプリケーション	5-80
外部アプリケーション	5-81
アプリケーションによる認証がない場合	5-82
OracleAS Portal アクセス制御リスト (ACL)	5-82
ポートレットのセキュリティ・マネージャ	5-83
ポートレットの表示	5-84
独自のセキュリティ・マネージャの実装	5-85
OracleAS Portal のサーバー・セキュリティ	5-85
メッセージ認証	5-86
HTTPS 通信	5-87
SSL の構成	5-88
LDAP (Oracle Internet Directory) セキュリティ	5-88
Oracle Internet Directory セキュリティの実装	5-89
ポートレットの表示	5-92
キャッシュを使用したポートレットのパフォーマンスの向上	5-93
前提	5-95
キャッシュのアクティブ化	5-95
有効期限ベースのキャッシュの追加	5-96
失効化ベースのキャッシュの追加	5-97
プロバイダ・サブレットの構成	5-97
OracleAS Web Cache の失効ポートの定義	5-98
XML プロバイダ定義の構成	5-100
手動によるキャッシュの失効化	5-101
妥当性チェック・ベースのキャッシュの追加	5-102
多言語ポートレットの作成	5-103
前提	5-103
ポートレットの国際化	5-103
ポートレット・コンテンツの翻訳の提供	5-103
ポートレット属性の翻訳の提供	5-106
ポートレットの表示	5-110
Oracle JDeveloper を使用した Struts ポートレットの作成	5-110
OracleAS Portal および Apache Struts のフレームワーク	5-110
Model-View-Controller の概要	5-110
Apache Struts の概要	5-112
OracleAS Portal と Struts の統合	5-113
Oracle Struts ポートレット	5-113
まとめ	5-114

Struts ポートレットの作成	5-115
Struts ポートレットの作成	5-115
ポートレット・アクションをホスティングするための新しいフローとビューの作成 ...	5-117
新しい JSP の作成	5-117
ポートレットの作成	5-118
ポートレットを拡張して Portal のビジネス・ロジックを追加	5-119
プロバイダの登録	5-119
まとめ	5-119

第 IV 部 付録

A ポートレット・ビルダーを使用したポートレットの構築

ウィザードによるポートレットの構築	A-2
OracleAS Portal でのスキーマの作成	A-2
スキーマの作成	A-3
データベース・オブジェクトに対する権限の付与および取消し	A-5
1 つ以上のロールへのスキーマの登録	A-8
ローカルで構築されたポートレットに対するプロバイダの作成	A-10
プロバイダの公開	A-11
OracleAS Portal のウィザードを使用したポートレットの作成	A-14
宣言によるポートレットの構築	A-16
宣言によるフォームの構築	A-19
宣言によるレポートの構築	A-36
interMedia リッチ・コンテンツに対するフォームおよびレポートの構築	A-62
宣言によるチャートの構築	A-64
宣言による値リストの構築	A-88
ポートレット・ビルダーのコンポーネントの編集	A-92
ポートレットの管理	A-94
コンポーネント管理ページへのナビゲート	A-94
ポートレット名の変更	A-95
ポートレットの削除	A-95
ポートレットのコピー	A-96
ポートレットに対する PL/SQL パッケージの生成	A-96

ソース・コードの表示	A-97
ポートレットのパッケージ仕様および本体の表示	A-97
ポートレットのコール・インタフェースの表示	A-98
ポートレットに対するロックの管理	A-98
バージョンの管理	A-99
ポートレットのセキュリティ管理	A-100
ポートレットに対するアクセス権限の付与	A-102
ポートレットに対するアクセス権限のプロバイダからの継承	A-102
個々のユーザーに対するアクセス権限の付与	A-103
ポートレットに対するテスト・ランの実施	A-105
フル・ページとしてのコンポーネントの実行	A-106
ポートレットとしてのコンポーネントの実行	A-106
カスタマイズ・フォームを介したコンポーネントの実行	A-107
ポートレットのカスタマイズ・フォームを介したポートレットとしてのコンポーネントの実行 ...	A-107
バッチ・モードでの実行	A-109
バッチ・ジョブのための <code>init.ora</code> パラメータの設定	A-109
既存のコンポーネントへの「バッチ」ボタンの追加	A-110
OracleAS Portal スキーマの参照	A-111
追加機能のコーディング	A-113
バインド変数の使用	A-114
フォームのアイテムに対するイベント・ハンドラの記述	A-115
フォームのアイテムに対する JavaScript イベント・ハンドラの記述	A-115
フォームのボタンに対する PL/SQL イベント・ハンドラの記述	A-116
PL/SQL によるフォームの値の取得および設定	A-117
PL/SQL によるフォームまたはレポートの Cookie の取得または設定	A-119
ページ・パラメータを介した値の定義	A-120
共有コンポーネントによる外観の作成	A-125
共有コンポーネントに対するアクセス権限の付与	A-126
JavaScript によるフィールド・レベルおよびフォーム・レベルの妥当性チェック	A-127
フィールド・レベルまたはフォーム・レベルの妥当性チェックを実行する JavaScript の 作成に関するガイドライン	A-128
共有コンポーネント・プロバイダの下に JavaScript を作成する方法	A-129
JavaScript のフォームへの追加	A-129
色定義の作成	A-130
イメージ定義の作成	A-131
フォント定義の作成	A-132

ユーザー・インタフェース・テンプレートの使用方法	A-133
構成済ユーザー・インタフェース・テンプレートの構築	A-133
非構成済ユーザー・インタフェース・テンプレートの構築	A-135
UI テンプレートの使用を可能にするためのページ・グループの構成	A-139
UI テンプレートのページへの適用	A-141
例：チャートおよびレポートの構築	A-144
演習：チーム詳細レポートの構築	A-144
演習：平均給与チャートの構築	A-147
演習：チーム賞与レポートの構築	A-148

用語集

索引

はじめに

このマニュアルでは、多様なツールとテクノロジーを使用して Oracle Application Server Portal (OracleAS Portal) のポータルを構築する方法について説明します。選択可能な各種テクノロジーの概要、ユーザーの要件に最適なテクノロジーの選択、ポータルの構築とデプロイに適切なツールの使用などについても説明します。

対象読者

このマニュアルは、主にポータル開発者を対象としていますが、ページ設計者にも有用です。

最初にポートレット・テクノロジーを理解したうえで選択し、次にそのテクノロジーを使用してポートレットを構築するための各手順が説明されています。自分に最も関係のある章を見つけるには、「このマニュアルの構成」を参照してください。

OracleAS Portal での様々な権限、およびこれらの権限が実行可能なタスクに与える影響については、『Oracle Application Server Portal ユーザーズ・ガイド』を参照してください。

ポータル開発者 ポータル開発者は、すべての Portal DB プロバイダを作成し、すべての共有コンポーネントを管理できるグローバル権限を付与されたユーザーです。ポータル開発者の主なタスクは、ポートレットを構築し、ページ設計者や他のユーザーがそのポートレットを各自のページに組み込めるようにすることです。OracleAS Portal には、ポートレットを構築するための多様なツールとテクノロジーが用意されているため、ポータル開発者は、プログラミングに関する十分な知識がなくても作業できます。

ページ設計者 ページ設計者（ページ管理者とも呼ばれます）は、ページの管理権限を持つユーザーです。この権限を持つユーザーは、ページに対するすべての処理を実行し、そのページの下にサブページを作成できます。ページ設計者は、ページ・レイアウト（領域の構成）の設計、および他のユーザーへのページ権限の割当て（どのユーザーがページにコンテンツを追加できるかの決定など）を行うことがあります。

ページがテンプレートに基づいている場合は、そのページに対してページ設計者が制御できる範囲が制限されることがあります。

このマニュアルの構成

このマニュアルの構成は次のとおりです。

第 I 部「ポートレットの概要」

ポートレットの概要を説明します。

第 1 章「ポートレットの概要」

各種ポートレットについて説明し、ポートレット開発を従来の Web ページ開発と比較します。

第 II 部「ポートレット・テクノロジー」

ポートレット構築テクノロジーを選択する際の考慮事項および各ポートレット・テクノロジーを使用した場合の利点について説明し、ユーザーが自分の目的に最も適したテクノロジーを決定できるようにします。

第2章「ポートレット・テクノロジーのマトリックス」

ポートレットの機能、特性、テクノロジーおよびツールの要約を記載します。これらはマトリックス形式で記載されており、ユーザーが自分のニーズに最も適したポートレット構築テクノロジーを決定できるように設計されています。マトリックスでは、横軸上に OracleAS Portal で使用できるテクノロジーとツールをリストし、縦軸上でそれぞれの機能と特性を要約しています。

第III部「ポートレットの構築」

このマニュアルに記載されているポートレット・テクノロジーの使用方法について説明する章が含まれています。

第3章「OmniPortlet を使用したポートレットの構築」

OmniPortlet ウィザードの様々なタブとコントロールを使用する方法、およびプラグブルな拡張機能をインストールして OmniPortlet の機能を拡張する方法について説明します。

第4章「Web クリップングを使用したコンテンツ・ベースのポートレットの構築」

Web クリップング・スタジオを使用し、Web クリップングに基づいてポートレットを作成する方法について説明します。

第5章「Java ポートレットの構築」

JSR-168 ポートレット標準に基づいて Java ポートレットを作成する方法、Oracle Portal Developer Kit-Java を使用して Java ポートレットを構築する方法、および Struts アプリケーションからポートレットを作成する方法について説明します。

第IV部「付録」

付録のサポート情報が記載されています。

付録A「ポートレット・ビルダーを使用したポートレットの構築」

ウィザードを使用してポートレットを作成する方法、および様々な種類のポートレットを作成、編集、管理および実行する手順について説明します。

用語集

このマニュアルおよび他の OracleAS Portal マニュアルで使用される用語の定義を記載します。

関連ドキュメント

詳細は、OracleAS Portal ドキュメント・セットの次のマニュアルを参照してください。

- 『Oracle Application Server Portal リリース・ノート』
- 『Oracle Application Server Portal ユーザーズ・ガイド』
- 『Oracle Application Server Portal 構成ガイド』
- 『Oracle Application Server Portal エラー・メッセージ・ガイド』

Oracle Application Server ドキュメント・セットの次のマニュアルも役立ちます。

- 『Oracle Application Server 10g 概要』
- 『Oracle HTTP Server 管理者ガイド』
- 『Oracle Application Server Web Cache 管理者ガイド』

表記規則

このマニュアルで使用される表記規則は、次のとおりです。

規則	意味
太字	太字は、強調および画面に表示される項目名を表すために使用されます。
大文字	大文字は、プロシージャ名を示します。
<>	山カッコは、ユーザーが入力する名前を表します。
[]	大カッコは、カッコ内の項目を任意に選択できることを表します。
.	垂直の省略記号は、例に直接関係しない情報が省略されていることを示します。

推奨ブラウザ

OracleAS Portal を使用する場合は、次の Web ブラウザのいずれかを使用することをお勧めします。

- Netscape 4.79
- Microsoft Internet Explorer 5.01 または 5.5 (Service Pack 1) (Windows NT および Windows 2000 の場合)
- Microsoft Internet Explorer 6.0 (Windows XP の場合)

推奨バージョンより前のバージョンのブラウザを使用すると、JavaScript エラーが発生する可能性があります。

キャッシュ設定

使用しているブラウザで常に正しいポータル・コンテンツが表示されるように、ブラウザのキャッシュ設定を次のようにしてください。

Internet Explorer を使用している場合：

1. メニューから「ツール」→「インターネット オプション」を選択します。
2. 「全般」タブが選択されていることを確認します。
3. 「インターネット一時ファイル」セクションで「設定」ボタンをクリックします。
4. 「保存しているページの新しいバージョンの確認」ラジオ・グループで、「ページを表示するごとに確認する」を選択します。
5. 「OK」をクリックします。
6. 「OK」をクリックします。

Netscape を使用している場合：

1. メニューから「編集」→「設定」を選択します。
2. 「詳細」ノードを展開します。
3. 「キャッシュ」をクリックします。
4. 「キャッシュしたドキュメントとネットワーク上のドキュメントとの比較」ラジオ・グループで「毎回」を選択します。
5. 「OK」をクリックします。

イメージ設定

次のようにしてイメージが自動的にロードされるように設定してください。

Internet Explorer を使用している場合：

1. メニューから「ツール」→「インターネット オプション」を選択します。
2. 「詳細設定」タブをクリックします。
3. 「マルチメディア」ノードのオプションまでスクロールし、「**画像を表示する**」を選択します。
4. 「OK」をクリックします。

Netscape を使用している場合：

1. メニューから「編集」→「設定」を選択します。
2. 「詳細」をクリックします。
3. 「自動的に画像を読み込む」チェック・ボックスを選択します。
4. 「OK」をクリックします。

帯域幅の低い接続でパフォーマンスを向上させるために、この設定が無効になっていることがあります。ただし、イメージが自動的にロードされないときに発生する一般的な問題として、一度ログアウトすると、ブラウザを閉じて再起動しなければ、再度ログインできないことがあります。したがって、この設定は常に有効にしておくことをお勧めします。

第 I 部

ポートレットの概要

第 I 部の内容は次のとおりです。

- [第 1 章「ポートレットの概要」](#)

1

ポータルレットの概要

この章では、ポータルレットおよびポータルレットの構造について説明し、OracleAS Portal で使用できる様々なポータルレット構築ツールの概要を示します。

- [ポータル開発の概要](#)
- [ポータルレットの概要](#)
- [ポータルレットの構造](#)
- [ポータルレット・リソース](#)

ポータル開発の概要

OracleAS Portal を使用すると、複数の関連性のないデータソースからの情報を 1 つの編成されたビューに表示できます。このビューがポータル・ページで、ポートレットと呼ばれるコンポーネントを 1 つ以上格納できます。各コンポーネントは、それぞれのコンテンツを異なるデータソースから取得できます。

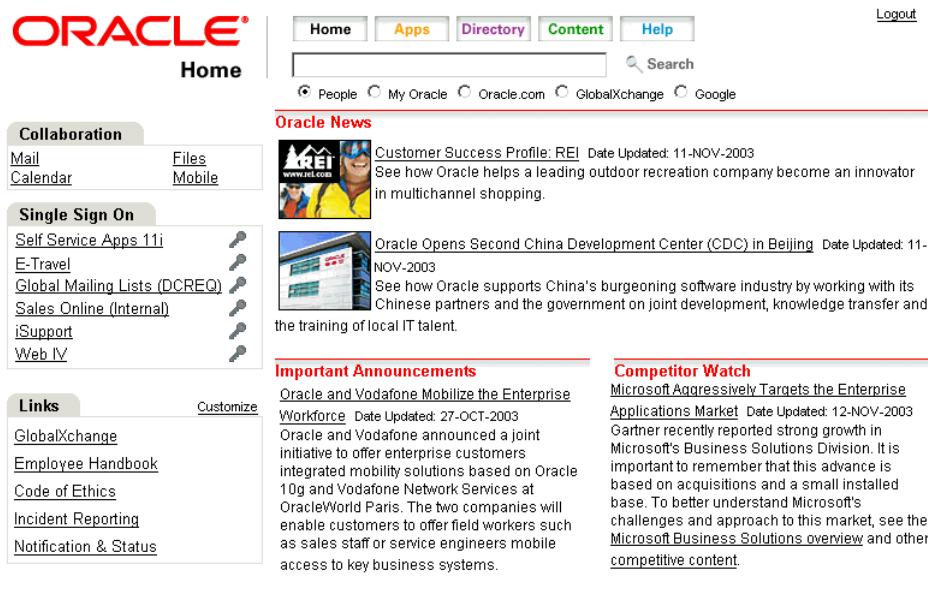
OracleAS Portal には、ポートレットを開発してポータル・ページに追加するために必要なツールがすべて用意されています。Portal のツールは、ビジネス開発の初心者から経験豊かな IT プログラマーに至る広範な開発技能をサポートします。ポートレットは、ポータル・ユーザー・インタフェースを介した宣言型でも、Oracle Application Server Portal Developer Kit (PDK) と呼ばれる Portal の Application Program Interface (API) のコレクションを介したプログラムでも開発できます。さらに、OracleAS Portal に対しては外部の他の開発ツールを使用してポートレットを開発し、PDK およびプロバイダと呼ばれる Portal エンティティによって統合できます。プロバイダの詳細は、[第 2 章「ポートレット・テクノロジーのマトリックス」](#)を参照してください。

この章では、ポートレットを定義し、事前に構築されたポートレットのソースとポートレット構築用のリソースをリストして説明します。また、そのジョブに対する最適なリソースを示します。

ポートレットの概要

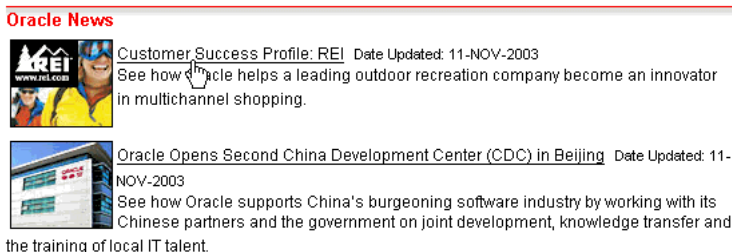
ポートレットは再利用可能でプラグgableな Web コンポーネントで、多数の異なるソースからコンテンツを描画できます。典型的なポートレットは、Web コンテンツのサマリーを表示するポートレットです。

図 1-1 My Oracle ホーム・ページのポートレット



たとえば、ポータルには、リンクされたニュース記事の見出しを表示するニュース・フィード・ポートレットを設定できます。これらの見出しには、記事の内容を説明するセンテンスがそれぞれ含まれています (図 1-2)。

図 1-2 My Oracle ホーム・ページの Oracle ニュース・ポートレット



ユーザーは、リンクされた見出しをクリックして、外部のニュース・サービスでホスティングされている記事のフル・テキストを取得します (図 1-3)。このポートレットには、ソースでのニュース記事の追加と削除にあわせて、見出しを自動的に変更するという動的な性質があります。

図 1-3 ポートレット・リンクからのコンテンツ・ターゲット

MyOracle Oracle.com Contact In the Know

In the Know

ORACLE In the Know e-zine

In the Know
PeopleSoft Special Edition

In the Know Home

Top Stories
Customer Success Profile: REI

In Depth Story
IDC: Critical to Oracle's Global Teamwork

FYI
Oracle UK Offers Green Transport Alternatives to Employees

In the Know Standard Edition Archives

Customer Success Profile: REI

 Occasionally, *In the Know* will highlight reference customers. Recreational Equipment, Inc., (REI), is one of the latest customer approved success stories and it highlights how the company has become an innovator in multichannel shopping.

REI is the first and largest store to offer a broad selection of outdoor gear, expert advice and in-depth information about products and outdoor recreation online. The company's online sales represent 15% of its total sales revenue, with the Web site receiving up to 2.5 million hits a day. REI uses information from its online database to analyze buying trends, access product information, order inventory and manage in-store computer kiosks.

REI.com REI-OUTLET.com REI Adventures Stores & Events



REI realized it could raise its online availability, ease Web troubleshooting and improve customer service by consolidating its four disparate databases servers into one clustered database. In 2002, it chose Oracle9i Database with Real Application Clusters (RAC).

ポートレットは、複数のソースから発信されたデータのシームレスな単一のビューを提供します。共通のページに様々なポートレットを配置できるため、ユーザーは単一のソースとして利用できます。実際には、コンテンツは複数のソースから導出されています。

ポートレットでは、他の Web サイトからの抜粋の表示、主要情報のサマリーの生成、検索の実行および多様なデータソースからの情報を収集したコレクションへのアクセスなどを実行します。

ポートレットの構造

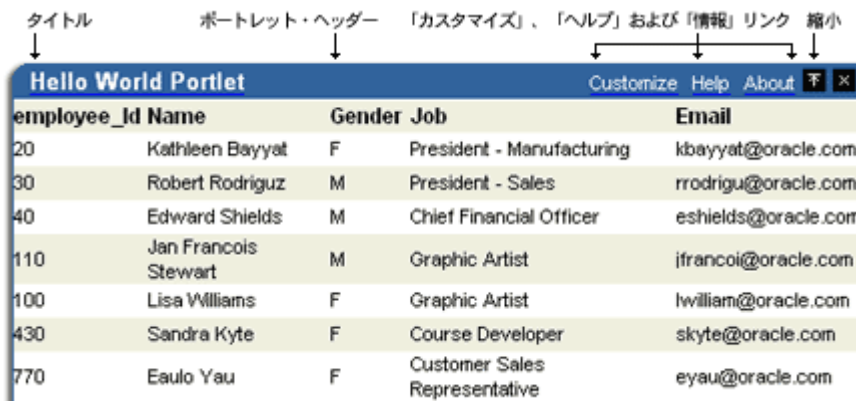
ページ上のポートレットは、HTML 表のセルでレンダリングされます。ポートレットには、HTML、書式設定したテキスト、イメージ、HTML フォームの要素など、様々な種類のコンテンツを表示できます。

図 1-4 に、ポートレット・タイトルがヘッダーに含まれているポートレットの構造を示します。ポートレット・タイトルにはハイパーリンクを作成できます。このリンクによって、ユーザーがタイトルをクリックすると、ポートレットがフル・ブラウザ・ページに表示されます。ポートレットには、レイアウトをページ上の他のポートレットと区別できるように、境界線を含めることもできます。

すべてのポートレットには、「カスタマイズ」、「ヘルプ」および「情報」の3つのリンクがあります。ポートレット開発者は、これらのリンクをページ設計者に公開し、ページ設計者は、これらのリンクをオンまたはオフにできます。「カスタマイズ」リンクをクリックすると、多数のオプションが表示されます。このオプションで、エンド・ユーザーはポートレットの各種の属性をパーソナライズできます。「ヘルプ」リンクをクリックすると、ポートレットのエンド・ユーザーを支援するために作成するヘルプ・テキストを含むウィンドウが表示されます。「情報」リンクをクリックすると、ポートレットのコンテンツを記述するために作成するウィンドウが表示されます。

各ポートレットには、標準の縮小アイコンも含まれています。エンド・ユーザーは、このアイコンをクリックして、ページ上のポートレットを縮小または拡大できます。

図 1-4 ポートレットの構造



ポートレット・リソース

ポートレット・リソースには、事前に構築されたポートレットが多数含まれています。これらのポートレットは、複雑な設定なしに、**OracleAS Portal**、**Oracle E-Business Suite** およびサード・パーティのソースも含めた多くのソースから使用できます。ポートレット・リソースには、ポートレット構築ツールも含まれています。これらのツールは、**Portal** のユーザー・インタフェースを介して利用することも、**PDK** や他の **Oracle Tools** から利用することもできます。これらのツールには、異なる開発者のロールをターゲットにした異なる製品機能がそれぞれ用意されています。

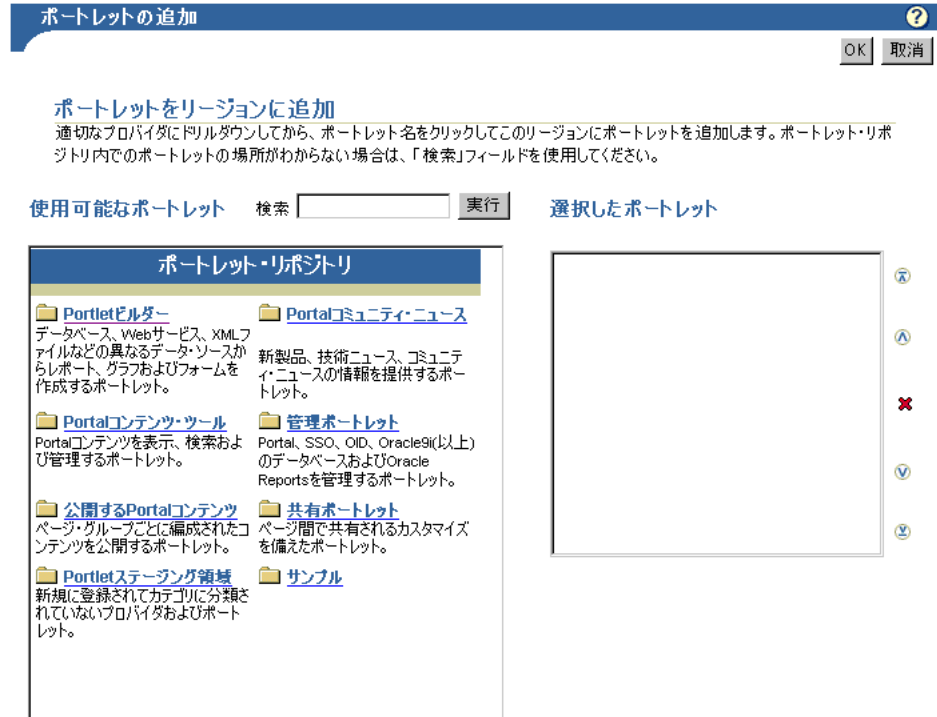
この項では、様々なポートレット・リソースについて説明し、リソースの使用に必要な専門知識のレベルと最適な使用例を示します。この項の内容は、次のとおりです。

- [複雑な設定が不要なポートレット](#)
- [事前に構築されたポートレットのその他のソース](#)
- [Web クリップング](#)
- [OmniPortlet](#)
- [ポートレット・ビルダー](#)
- [プログラムのなポートレット](#)

この項では、様々なポートレット・リソースの概要を説明します。各ツールとその利点の詳細は、[第2章「ポートレット・テクノロジーのマトリックス」](#)を参照してください。

複雑な設定が不要なポータル

図 1-5 ポータル・リポジトリ



概要

複雑な設定が不要なポータルとは完全に開発された登録済のポータルのことで、OracleAS Portal をインストールすると、ポータル・リポジトリから即時に使用できます (図 1-5)。このポータルには、「検索」、「保存された検索」、「お気に入り」、「通知」などが含まれます。

注意: OracleAS Portal の事前に構築されたポータルの詳細は、『OracleAS Portal ユーザーズ・ガイド』を参照してください。

対象ユーザー

複雑な設定が不要なポートレットは、エンド・ユーザーおよびページ設計者に最適なポートレットですが、あらゆる専門知識レベルのユーザーが使用できます。

使用する場合

複雑な設定が不要なポートレットは、そのポートレットが提供する機能がニーズを満たしている場合や、すぐに利用できるカスタマイズのレベルで目的のタスクを十分に完了できる場合に使用します。

ユーザー・インタフェースの変更が必要な場合や希望する機能のために複雑な設定が必要な場合など、ポートレットの拡張やカスタマイズが必要なときには、別の方法を検討してください。

事前に構築されたポートレットのその他のソース

概要

事前に構築されたポートレットのその他のソースには、パートナ・ポートレット、統合化ソリューションおよび Portal Knowledge Exchange が含まれます。

パートナ・ポートレットは、オラクル社と、様々なタイプの主要なシステム・インテグレータ、ソフトウェア・ベンダーおよびコンテンツ・プロバイダとの提携によって提供されます。これらのポートレットには、Portal Center (<http://portalcenter.oracle.com>) にある Portal Catalog を介してアクセスできます。次のような用途例のポートレットがあります。

- Point-to-Point 運転方向の生成
- 広範囲なソースからの IT 情報へのアクセス
- ニュース、株式および天気に関するサマリー情報の表示

Portal Center では、Microsoft Exchange、Lotus Notes、SAP、IMAP、SMTP などの一般的なアプリケーションの基本機能が必要な顧客に便利な統合化ソリューションも提供しています。

Portal Center でもアクセス可能な Portal Knowledge Exchange は、Portal Developer Services により提供されています。コミュニティ・メンバーは、ポートレット・サンプル、ヒント、ホワイト・ペーパー、サンプル・コードなどを含むポータル専門知識を交換します。メンバーは、Portal Center 上に与えられた個人フォルダを使用して、ポートレット・コードやポータル開発に関する見通しをアップロードし、他の開発者による投稿をダウンロードして評価できます。

対象ユーザー

完全に開発されたダウンロード可能なポートレットは、OracleAS Portal での Web プロバイダとデータベース・プロバイダのダウンロード、インストールおよび登録方法を理解しているエンド・ユーザーやページ設計者に最適なポートレットです。これらのポートレットは、すべての経験レベルで使用可能です。

使用する場合

複雑な設定が不要なポートレットと同様に、他のソースによって事前に構築されたポートレットは、そのポートレットが提供する機能がニーズを満たしている場合や、すぐに利用できるカスタマイズのレベルで目的のタスクを十分に完了できる場合に使用します。

ユーザー・インターフェースの変更が必要な場合や希望する機能のために複雑な設定が必要な場合など、ポートレットの拡張やカスタマイズが必要なときには、別の方法を検討してください。

Web クリップング

概要

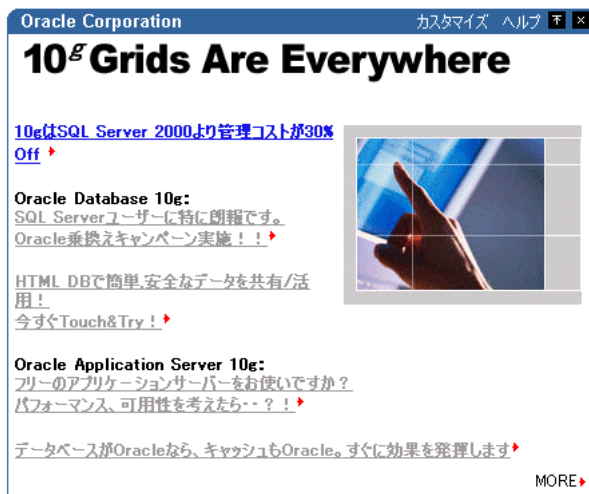
Web クリップングは、既存の HTML ページのすべてまたは一部をクリップし、ポートレットとして再利用するための汎用ポートレットです。基本および HTML フォーム・ベースのサイトはクリップ可能です。

Web クリップング・ポートレットを作成する場合、ポータル・ページの設計者は、Web ブラウザを使用して必要なコンテンツを含む Web ページにナビゲートします。ユーザーは、Web クリップング・スタジオを使用して、ターゲット・ページのビジュアル・レンダリングをドリルダウンし、必要なコンテンツのみを選択できます (図 1-6 および図 1-7)。

図 1-6 Web クリップング・スタジオを使用した Web コンテンツの選択



図 1-7 クリップしたコンテンツを Portal のポートレットとしてレンダリング



注意： 前述の例（[図 1-6](#) および [図 1-7](#)）のイメージは、それぞれのページがリフレッシュされるたびに更新されます。この機能はクリップしたポートレット内で維持されます。したがって、2つの図のイメージは同じではありません。

Web クリップング・ポートレットは次の機能をサポートします。

- 様々なスタイルのログイン・メカニズムを介したナビゲーション
これには、フォーム・ベースと JavaScript ベースの送信、および Cookie ベースのセッション管理を使用した HTTP の Basic 認証と Digest 認証が含まれます。
- クリップングのファジー・マッチング
ソース・ページ内部で Web クリップングの順序が変更されたり、文字のフォント、サイズまたはスタイルが変更された場合でも、Portal では正確に識別され、ポートレット・コンテンツとして配信されます。
- 広範囲な Web コンテンツの再利用
これには、HTTP GET メソッドと POST（フォーム送信）メソッドを使用して取得された、HTML 4.0.1、JavaScript、アプレットおよびプラグイン対応コンテンツで作成されたページの基本サポートが含まれます。
- 外部 Web サイトの情報の使用
これには、外部アプリケーションとの統合が含まれます。この統合によって Oracle Application Server Single Sign-On を利用して、認証されている外部 Web サイトのコンテンツをクリップできます。
- インライン・レンダリング

Web クリップング・ポートレットは、ポートレットのコンテキスト内にリンクを表示するように設定できます。この設定によって、ユーザーが Web クリップング・ポートレット内でリンクをクリックすると、同じポートレット内に結果が表示されます。この機能は、内部および外部の Web サイトで使用できます。

注意： Oracle Application Server Portal Developer Kit (9.0.4.0.2) からは、Web クリップング・プロバイダを使用すると、ユーザーは、既存のポートレットを URL ベースのポートレットから Web クリップング・ポートレットに移行できます。また、外部アプリケーションとも統合できます。詳細は、Portal Center (<http://portalcenter.oracle.com>) を参照してください。

対象ユーザー

Web クリップpingは、既存の Web ページを利用してポートレットを迅速に開発するページ設計者に最適です。Web クリップping・ポートレットは、ポートレット・リポジトリでアクセスできます。したがって、アクセス権を持つすべてのユーザーが使用できます。

使用する場合

Web クリップpingは、既存の Web ページのコンテンツをコピーし、ポータルでポートレットとして公開する場合に使用します。クリップしたポートレット内で情報の表示方法を変更する場合は、別の方法を検討してください。つまり、UI やアプリケーション・フローを制御する必要なしに、Web ベースのアプリケーションにアクセスします。高レベルの制御が必要な場合は、Web クリップpingのかわりに OracleAS Portal OmniPortlet の Web ページ・データソースを使用してください。

Web クリップping・ポートレットの使用は、次の例のような場合に検討します。

- 株式チャート・ポートレット。株式市場の日次実績チャートを、自社の財務顧問の Web サイトから表示するポートレットを作成する場合。この情報は、自社でプロキシを使用している場合でも、外部 Web サイトからクリップできます。
- パーソナライズされた天気ポートレット。天気情報を主要なインターネット天気サイトから表示するポートレットを作成し、必要な郵便番号を指定して、ユーザーがそのポートレットをパーソナライズできるようにする場合。
- Web メール・ポートレット。ユーザーがそれぞれの機密 Web メール・アカウントにポートレット経由でアクセスし、そのポートレット内に表示する受信ボックスにアクセスする場合。

OmniPortlet

図 1-8 表形式を使用した OmniPortlet

Webサービス・ポートレット				
"RESEARCH"部門の社員のリスト				
EMPNO	ENAME	JOB	MGR	HIREDATE
7876	ADAMS	CLERK	7788	1987-05-23
7902	FORD	ANALYST	7566	1981-12-03
7566	JONES	MANAGER	7839	1981-04-02
7788	SCOTT	ANALYST	7566	1987-04-19
7369	SMITH	CLERK	7902	1980-12-17

概要

OracleAS Portal の OmniPortlet は宣言型のポートレット構築ツールで、これを使用すると、XML ファイル、カンマで区切られた値ファイル（スプレッドシートなど）、Web サービス、データベース、Web ページおよび SAP データソースを含めた多様なデータソースに対してポートレットを構築できます。OmniPortlet ユーザーは、データに対して事前に作成されたレイアウトを選択することもできます。事前に作成されたレイアウトには、表、ニュース、箇条書き、フォームまたはチャートが含まれます。

注意： OmniPortlet の詳細は、Portal Center (<http://portalcenter.oracle.com>) を参照してください。

対象ユーザー

OmniPortlet は、ページの設計者や開発者に最適なポートレットです。

使用する場合

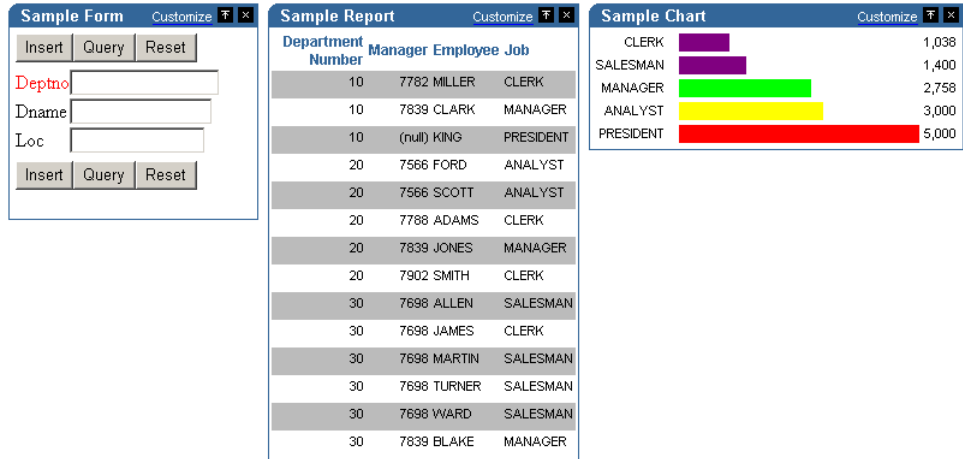
OmniPortlet は、多様なレイアウトを使用した多様なデータソースに対してポートレットを迅速に構築する場合に使用します。ポートレットの設計や機能を完全に制御する場合は、別の方法を検討してください。

OmniPortlet の使用は、次の例のような場合に検討します。

- RSS ニュース・フィード・ポートレット。ユーザーに対してライブのニュース情報をスクロール表示するポートレットを作成する場合。データの発信元は、Oracle Technology Network Headlines などの Really Simple Syndication ニュース・フィードです。ニュース・ソースへのハイパーリンクを含むポートレットも必要です。
- 販売チャート・ポートレット。自社の販売結果に関する最新情報を表す場合。円グラフでもデータを表示します。自社の販売情報はリモートのリレーショナル・データベースに保存されます。
- SAP ポートレット。会社の SAP システムの情報を表示する場合。会社の SAP Business Suite への負荷を最小化するために、システムから取得した情報は1日分ごとに各ユーザー・ベースでキャッシュする必要があります。

ポートレット・ビルダー

図 1-9 ポートレット・ビルダーのサンプルのフォーム、レポートおよびチャート



概要

OracleAS Portal には、複数のポートレット構築ウィザードが含まれています。これらのウィザードには、Portal ナビゲータの「プロバイダ」タブを使用してアクセスできます。これらのウィザードは、チャート、レポート、フォーム、カレンダーおよび値リストの構築に使用できます。

使用する場合

可能なかぎり、ポートレット・ビルダーのかわりに **OmniPortlet** を使用することをお勧めします。**OmniPortlet** には柔軟性があり、データとレイアウトを切り離すことができるため、ポートレット全体を再作成せずに、レポートをチャートに変更できます（ポートレット・ビルダーの場合は再作成が必要です）。また、**OmniPortlet** には、多数の異なるポータルを同時にデプロイするためのオプションがより多く用意されています。**OracleAS Portal** では、今後もポートレット・ビルダーをポートレット構築オプションとしてサポートします。ただし、新機能と拡張機能は **OmniPortlet** ツールが対象となります。

ポートレット・ビルダーの詳細は、[付録 A 「ポートレット・ビルダーを使用したポートレットの構築」](#) を参照してください。

プログラムのなポートレット

概要

OracleAS PDK には、プログラムのなポートレットの構築に使用できる一連のポートレット構築 API が含まれています。

注意： これらの API の詳細は、Portal Center (<http://portalcenter.oracle.com>) を参照してください。

対象ユーザー

このツールは経験と知識の豊富な IT 開発者に最適なツールです。

使用する場合

PDK は、非常に特化したビジネス・ルールやロジックがあり、カスタマイズした認証、動的結果の精度の高い処理およびユーザー・インタフェースの完全な制御が必要な場合に使用します。さらに、次の場合に PDK を使用します。

- 最初からポートレットを構築していて、そのポートレットの全機能を完全に制御する必要がある場合。
- Java または PL/SQL の知識がある場合。
- 機能する開始ポイントが必要な場合。
- PDK と OracleAS Portal プロバイダの構成をよく理解している場合。

複雑な設定が不要な宣言型ツールがニーズに対応していない場合は、このツールの使用を検討してください。

注意： PDK-PL/SQL については、このマニュアルでは詳しく説明していません。PDK-PL/SQL 固有の情報は、Portal Center (<http://portalcenter.oracle.com>) の「Developer Services」エリアを参照してください。

Oracle Application Server Portal Developer Kit で作成した Java ポートレットの使用は、次の例のような場合に検討します。

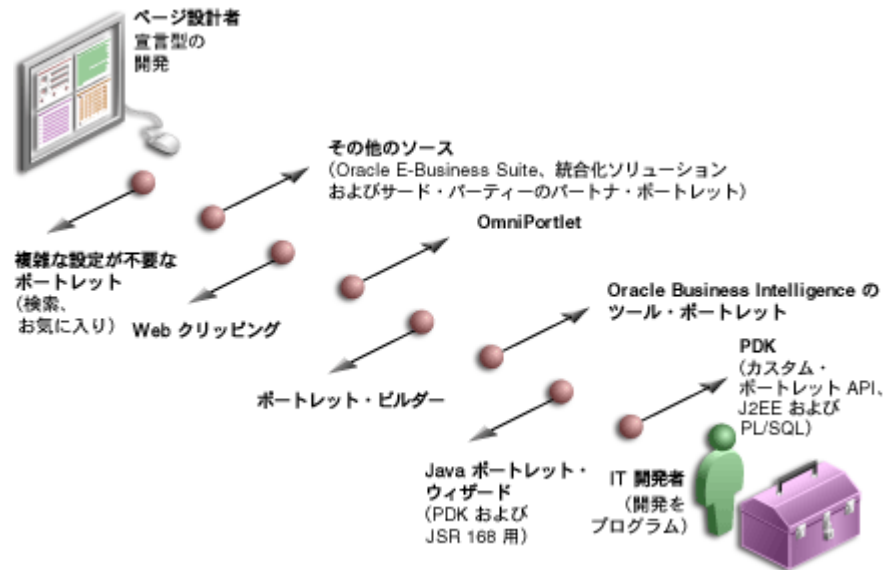
- ディスカッション・フォーラム・ポートレット。自社の JSP ベースのディスカッション・フォーラム・アプリケーションを OracleAS Portal に統合するポートレットを作成する場合。ディスカッション・フォーラム・ポストはリレーショナル・データベースに格納されます。このポートレットの外観は、自社のインターネット Web サイトの外観に厳密に従う必要があります。

- 電子メール・ポートレット。ユーザーが電子メールを自社のイントラネット・ポータルから送信できるポートレットを作成する場合。ユーザーがLDAPサーバー上のアドレス帳を使用できるように、電子メール・ポートレットを会社のLDAPサーバーに統合する必要があります。

使用するツールの決定

図 1-10 に、前項で説明したポートレット・リソースの範囲を示します。範囲の片方はページ設計者を対象とし、もう片方はポートレット開発者を対象にしています。自分の専門知識に最も近いユーザー・タイプに従ってツールを選択できます。

図 1-10 ページ設計者から経験豊かな開発者まで使用できるポートレット・リソース



まとめ

この章では、ポートレットの概要と OracleAS Portal で使用可能な様々なテクノロジーについて説明しました。これらのツールとテクノロジーの詳細は、[第 2 章「ポートレット・テクノロジーのマトリックス」](#)を参照してください。

第II部

ポートレット・テクノロジー

第II部の内容は次のとおりです。

- [第2章「ポートレット・テクノロジーのマトリックス」](#)

ポートレット・テクノロジーのマトリックス

この章では、ポートレットの機能、特性、テクノロジーおよびツールについて説明し、ユーザーが自分のニーズに最も適したポートレット構築テクノロジーを決定できるようにします。この章の内容は、次のとおりです。

- ポートレット・テクノロジーのマトリックス
- 一般的な適性
- 必要な専門知識
- デプロイ・タイプ
- キャッシュ・スタイル
- 開発ツール
- ポートレット作成スタイル
- ユーザー・インタフェースの柔軟性
- Web サイトからコンテンツを取得する機能
- コンテンツをインラインでレンダリングする機能
- チャート作成機能
- パブリック・ポートレット・パラメータのサポート
- プライベート・ポートレット・パラメータのサポート
- イベントのサポート
- ユーザー権限に基づいてポートレットの表示 / 非表示を切り替える機能
- 多言語サポート
- ページ区切りサポート
- シングル・サインオンおよび外部アプリケーション統合

ポートレット・テクノロジーのマトリックス

表 2-1 「ポートレット構築テクノロジーの比較マトリックス」では、横軸上に OracleAS Portal で使用できるテクノロジーとツールをリストし、縦軸上でそれぞれの機能と特性を要約しています。このマトリックスには、このマニュアルで詳細に説明する OmniPortlet、Web クリッピング、Java ポートレット (PDK-Java) および標準などのツールやテクノロジーが掲載されています。ポートレット・ビルダーについては付録で説明し、PL/SQL ポートレット (PDK-PL/SQL) については、このマトリックスでのみ説明します。

注意： これらはポートレットを構築するための主なツールですが、他の Oracle 製品 (Oracle Reports や Oracle Discoverer) など、その他のツールやテクノロジーもあります。これらのツールについては、このマニュアルでは説明しません。

この章の他の項では、表 2-1 にリストした特性の詳細を説明します。この表で機能と特性すべてに簡単に目を通した後、後続の項で詳細情報を参照してください。

表 2-1 ポートレット構築テクノロジーの比較マトリックス

Web クリッピング	OmniPortlet	PDK-Java	標準	ポートレット・ビルダー	PDK-PL/SQL
一般的な適性					
単純なウィザード・ベースのツール。他の Web サイトの Web コンテンツをポータル内に取得して表示する際に役立ちます。	ブラウザからアクセス可能なウィザード・ベースのツール。広範囲なデータソースからデータを取得して表示できます。	OracleAS Portal 専用に構築されたポートレットの API。最も柔軟性のあるプログラムによるアプローチです。	他のベンダーのポータルで動作するポートレット。Oracle では、WSRP と JSR-168 の両方がサポートされています。	ブラウザからアクセス可能なウィザード・ベースのツール。データベース中心の単純なアプリケーションまたはポートレットに最適です。	OracleAS Portal 専用に構築されたポートレットの API。最も柔軟性のあるプログラムによるアプローチです。
必要な専門知識					
専門知識は不要。	サポートされる 1 つ以上のデータソース、およびポートレット、ページ・パラメータ、イベントの概念に関する基本的な理解。	Java、サーブレット、JSP の知識。	Java、サーブレット、JSP の知識。	リレーショナル・データベース概念の基本的な理解。必要に応じて SQL、PL/SQL。	SQL、PL/SQL、PL/SQL Web ツールキット。
サポートされるデータソース (詳細は、「 必要な専門知識 」を参照)					
ネットワーク上で HTTP または HTTPS を介してアクセス可能なすべての Web サイト。	CSV、XML、Web サービス、SAP、SQL、Web サイト、JCA。	制限なし。	制限なし。	SQL (ローカル DB または DB リンクを介したリモート DB)。	SQL (ローカル DB または DB リンクを介したリモート DB)。

表 2-1 ポートレット構築テクノロジーの比較マトリックス (続き)

Web クリッピング	OmniPortlet	PDK-Java	標準	ポートレット・ビルダー	PDK-PL/SQL
デプロイ・タイプ					
Web プロバイダ	Web プロバイダ	Web プロバイダ	WSRP	データベース・プロバイダ	データベース・プロバイダ
キャッシュ・スタイル					
有効期限ベースのキャッシュ、失効化ベースのキャッシュ (カスタマイズ時に自動的に失効)。	有効期限ベースのキャッシュ、失効化ベースのキャッシュ (カスタマイズ時に自動的に失効)。	有効期限ベースのキャッシュ、妥当性チェック・ベースのキャッシュ、失効化ベースのキャッシュ、ESI。	妥当性チェック・ベースのキャッシュ、有効期限ベースのキャッシュ。	有効期限ベースのキャッシュ。	有効期限ベースのキャッシュ、妥当性チェック・ベースのキャッシュ、失効化ベースのキャッシュ。
開発ツール					
ブラウザ - ウィザード。	ブラウザ - ウィザード。	Oracle JDeveloper - Java ポートレット・ウィザード (または、ウィザードを使用しない他の Java 開発環境)。	Oracle JDeveloper - Java ポートレット・ウィザード (または、ウィザードを使用しない他の Java 開発環境)。	ブラウザ - 必要に応じて PL/SQL 開発環境。	PL/SQL 開発環境。
ポートレット作成スタイル					
その場で開発。	その場で開発。	ポートレットはその場で構築しません。ポートレットのページへの追加、デフォルトの編集およびパーソナライズを行います。	ポートレットはその場で構築しません。ポートレットのページへの追加、デフォルトの編集およびパーソナライズを行います。	開発後に追加、およびその場で開発。	ポートレットはその場で構築しません。ポートレットのページへの追加、デフォルトの編集およびパーソナライズを行います。
ユーザー・インタフェースの柔軟性					
適用外。	制限あり。	柔軟性が高い。	柔軟性が高い。	制限あり。	柔軟性が高い。
Web サイトからコンテンツを取得する機能					
あり。これが本来の機能です。	あり。Web データソースを使用します。	あり。java.net パッケージを使用します。	あり。java.net パッケージを使用します。	なし。	あり。UTIL_HTTP パッケージを使用します。
コンテンツをインラインでレンダリングする機能					
あり (Web クリッピング 9.0.4.0.2 以上でサポートされます)。	URL リライティングはサポートされませんが、パブリック・ポートレット・パラメータとイベントを使用することで実現できます。	プライベート・ポートレット・パラメータを使用します。	サーブレットおよび JSP を組み込みます (PortletContext.getRequestDispatcher() メソッドを使用)。	レポートおよびチャートのページ区切りがインラインでレンダリングされます。	プライベート・ポートレット・パラメータを使用します。

表 2-1 ポートレット構築テクノロジーの比較マトリックス (続き)

Web クリップング	OmniPortlet	PDK-Java	標準	ポートレット・ビルダー	PDK-PL/SQL
チャート作成機能					
適用外。	あり。2D チャートと 3D チャート。	BI Bean を使用します。	BI Bean を使用します。	HTML チャート。	プログラムによる HTML チャート。
パブリック・ポートレット・パラメータのサポート					
あり (Web クリップング 9.0.4.0.2 以上でサポートされます)。	あり。	あり。	なし。	あり。	あり。
プライベート・ポートレット・パラメータのサポート					
適用外。	適用外。	あり。	あり。	なし。	あり。
イベントのサポート					
あり。	あり。	あり。	ポートレットのプライベート・イベント (アクション)。	なし。	なし。
ユーザー権限に基づいてポートレットの表示 / 非表示を切り替える機能					
なし。ただし、UI を介して公開されないセキュリティ・マネージャの適用が可能です。	なし。ただし、UI を介して公開されないセキュリティ・マネージャの適用が可能です。	あり。セキュリティ・マネージャを使用します。	あり。サーブレット・セキュリティ・モデルは、PortletRequest.isUserInRole() や PortletRequest.getUserPrincipal() などのメソッドを使用してサポートされます。	あり。	あり。セキュリティ API を使用します。
多言語サポート					
適用外。	あり。	あり。	あり。	なし。	あり。
ページ区切りサポート					
適用外。	なし。	プログラムによるサポート。	プログラムによるサポート。	あり。	プログラムによるサポート。

表 2-1 ポートレット構築テクノロジーの比較マトリックス (続き)

Web クリップング	OmniPortlet	PDK-Java	標準	ポートレット・ビルダー	PDK-PL/SQL
シングル・サインオンおよび外部アプリケーション統合					
Web クリップング 9.0.4.0.2 以上で外部アプリケーション統合がサポートされます。	Basic 認証は、データソースに認証が必要な場合にサポートされます。	外部アプリケーション統合がサポートされます。LDAP 統合は、ポートレットが LDAP サーバーと同じファイアウォールの内側で実行される場合にサポートされます。	なし (カスタム・ユーザー属性を介して実行可能で)。LDAP 統合がサポートされます。	なし (OracleAS Portal リポジトリ内で実行されるため、SSO 統合は必要ありません)。	SSO は、mod_oso を使用することで有効になります。

一般的な適性

この項では、各ポートレット構築テクノロジーをその用途の特性 (ウィザード・ベースまたはプログラムなど) の観点から説明します。

Web クリップング

Web クリップングは、単純なウィザード・ベースのツールで、他の Web サイトの Web コンテンツをポータル内に取得して表示する際に役立ちます。Web クリップングでは、専門知識は不要です。

Web クリップングを使用して構築できるポートレットの例

- 株式チャート・ポートレット
- パーソナライズされた天気ポートレット
- Web メール・ポートレット

OmniPortlet

広範囲なデータソースからデータを取得して表示するために、簡単に使用できるウィザード・ベースのツールを探している場合は、**OmniPortlet** を検討してください。**OmniPortlet** は、ブラウザで完全に稼働します。実行する必要があるのは、ポータル・ページに **OmniPortlet** をドロップし、使用可能なデータソースのリストから、次に示すようなデータソースを選択する操作のみです。

- スプレッドシート
- SQL
- XML
- Web サービス
- Web ページ

OmniPortlet では、追加の開発ツールや高度な専門知識がなくても、再利用可能で高性能なポートレットを構築できます。

OmniPortlet で作成できるポートレットの例

- RSS ニュース・フィード・ポートレット
- 販売チャート・ポートレット
- SAP Business Suite ポートレット

Java ポートレット

ウィザード・ベースのポートレット構築ツールではニーズを満たせない場合は、Java を使用してプログラムでポートレットを構築できます。2003 年に、Java ポートレット API が Java Community Process によって標準化されました。Java Specification Request (JSR) 168 標準に対して構築されたポートレットは、異なるポータル・プラットフォーム間で相互運用できます。Oracle JDeveloper プラグインの 1 つである Java ポートレット・ウィザードは、Java ポートレットを開始する際に役立ちます。

注意： Java でポートレットを構築する場合は、ポートレットの機能を完全に制御できます。たとえば、外観や動作方法を制御できます。

Java を使用して構築できるポートレットの例

- ディスカッション・フォーラム・ポートレット
- 電子メール・ポートレット

ポートレット・ビルダー

ポートレット・ビルダーは、データ駆動ポートレットを作成するためのウィザード・ベースのツールで、データは Oracle データベースに常駐します。インタラクティブなフォームを構築して、データベース・レコードを挿入、更新および削除できます。柔軟性のあるレポートおよび HTML 棒グラフを作成して、データベースの情報を表示できます。また、ポートレット・ビルダーでは、動的リンクを使用して、データ駆動ポートレット間でパラメータを渡したり、ナビゲートすることもできます。

ポートレット・ビルダーを使用して構築できるポートレットの例

- データ入力ポートレット
- パートナの動的リスト・ポートレット
- 販売結果ポートレット

PL/SQL ポートレット

Java ポートレットと同様に、PL/SQL ポートレットは、組込みのポートレットではニーズを満たすことができない Web アプリケーションを構築するために、柔軟なアプローチを提供します。たとえば、アプリケーションで特別なビジネス・ルールやロジックを実装する必要がある場合、またはカスタム設計された認証要件を満たす場合などです。PL/SQL ポートレットは、通常、SQL および PL/SQL を使用してデータ集約型の操作を実行する必要があります。OracleAS Portal には、プログラムによるプロバイダ登録、オブジェクト・レベルの権限管理、ユーザー・インタフェース制御、多言語サポートなど、豊富な PL/SQL API が用意されています。

たとえば、情報プロバイダはカスタム・ポートレットを作成し、OracleAS Portal を介してアプリケーションをユーザーに表示できます。開発者は、OracleAS Portal Developer Kit (PDK) の仕様に従って独自のポートレットを構築し、プロバイダを OracleAS Portal に登録します。PDK を使用すると、ニーズに適したポートレットを開発できます。

PL/SQL を使用して構築できるポートレットの例

- コンテンツ・アップロード・ポートレット
- サイト・マップ・ポートレット
- 高度なデータ入力およびレポート・ポートレット

必要な専門知識

ポートレット構築ツールでは、ポートレットの開発技能を必要としない場合も、高度な専門知識があることを前提としている場合もあります。この項では、ポートレット構築ツールを、それぞれのツールを効果的に使用するために必要な知識レベルの観点から説明します。

Web クリップング

Web クリップングは、専門知識を必要としないツールです。ただし、クリップした Web ページのコンテンツをパラメータ化する場合は、パブリック・ポートレット・パラメータおよびページ・パラメータを理解している必要があります。

OmniPortlet

OmniPortlet では、ポートレット内で利用するデータソースに関する基本的な知識が必要です。

データソース	データソースについて必要な知識
スプレッドシート	ポートレットに表示するデータが含まれたスプレッドシートを指す URL。
SQL	データソースへの接続情報およびデータベースからデータを取得する SQL 問合せ。
XML	XML ソースの位置、必要に応じて XSL フィルタおよび XML スキーマのアドレス。
Web サービス	WSDL URL、Web サービスのメソッド、必要に応じて XSL フィルタの URL と XML スキーマの URL。
Web ページ	Web ページ・データソースは、Web クリップングと同じ環境で使用されます。したがって、専門知識は不要です。
J2EE Connector Architecture	OmniPortlet ウィザードの「タイプ」ページには表示されませんが、J2EE Connector Architecture (JCA) 1.0 アダプタも使用できます。JCA では、ERP システム (Oracle Financials、SAP、PeopleSoft など) に保持されている企業データを格納および取得するためのメカニズムが提供されます。

Java ポートレット

Java ポートレットを構築するには、少なくとも J2EE のサブセットに関する知識が必要です。HTML、Java サーブレットおよび XML に関する知識が必須ですが、JSP の経験も役立ちます。実行するタスクによって、Java に関するその他の知識が必要になる場合もあります。Java ポートレットを使用すると、(Java 言語でサポートされている) すべてのデータソースにアクセスできます。

ポートレット・ビルダー

ポートレット・ビルダーを使用する場合は、リレーショナル・データベースの概念をよく理解している必要があります。実現する内容によって、SQL または PL/SQL (あるいはその両方) の知識が必要になる場合もあります。ポートレット・ビルダーを使用すると、ローカル・データベース (Oracle Application Server インフラストラクチャ) から、またはデータベース・リンクを使用したリモート・データベースからデータを消費できます。

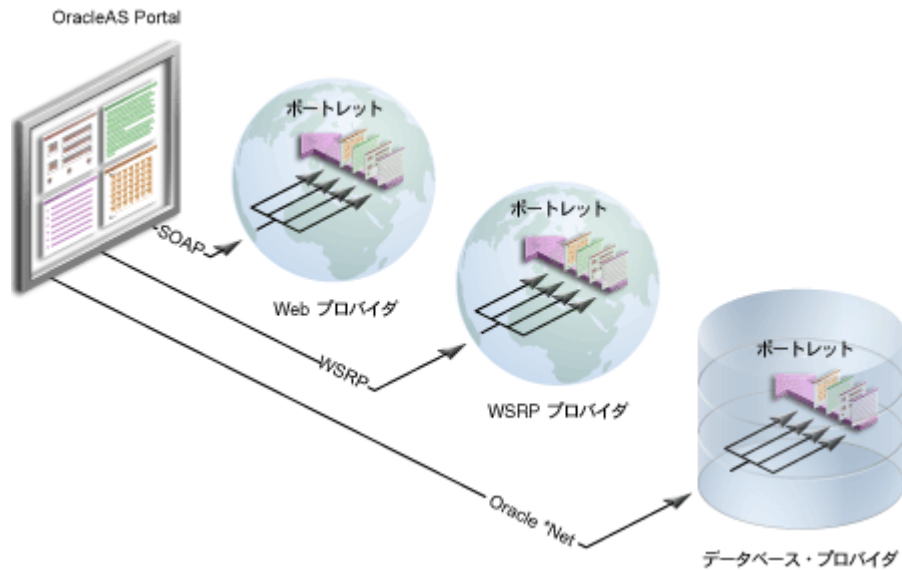
PL/SQL ポートレット

PL/SQL ポートレットを構築するには、SQL 文を作成し、SQL*Plus または Oracle データベースに接続できる同様の開発ツールを使用して PL/SQL プログラム・ユニットをコード化およびデバッグする方法を理解する必要があります。また、ポートレット・コンテンツを生成するために、HTML および PL/SQL Web ツールキットに関する知識も必要です。PL/SQL Server Pages (PSP) のコーディングの経験が必要な場合もあります。

デプロイ・タイプ

図 2-1 に示すとおり、ポートレットは、Web プロバイダ、WSRP プロバイダおよびデータベース・プロバイダの3つのプロバイダ・タイプを介して OracleAS Portal にデプロイできます。Web プロバイダは、J2EE アプリケーション・サーバーにデプロイされます。このサーバーは多くの場合リモートで、HTTP を介した Simple Object Access Protocol (SOAP) を通じて OracleAS Portal と通信します。OASIS 標準である Web Services for Remote Portlets (WSRP) が、OracleAS Portal の開発者プレビュー版でサポートされています。データベース・プロバイダは、PL/SQL で実装され、OracleAS Portal がインストールされている Oracle データベースにデプロイされます。

図 2-1 ポートレット・プロバイダの概要



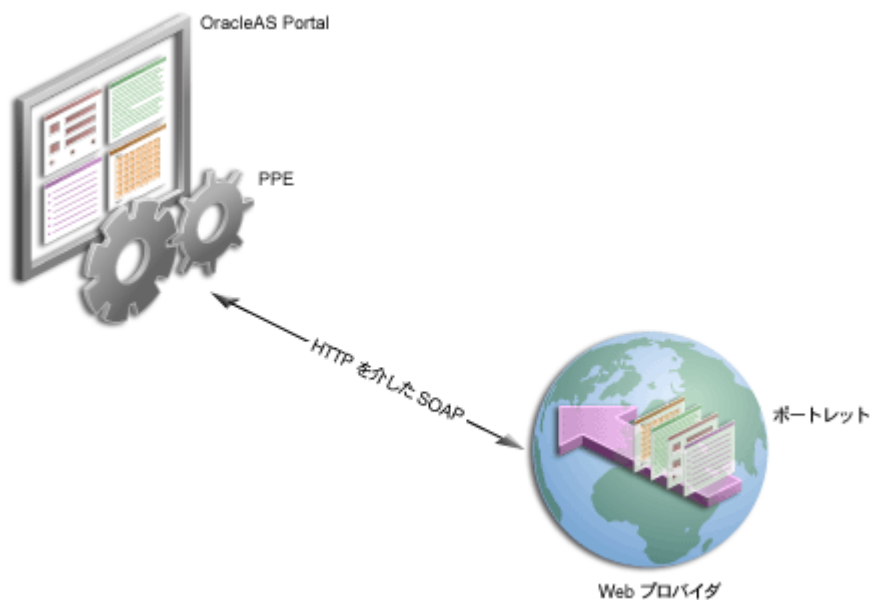
Web プロバイダ

Web プロバイダは、最も一般的に使用され、柔軟なタイプのプロバイダです。Web プロバイダは、OracleAS Portal と同じアプリケーション・サーバー、リモート・アプリケーション・サーバーまたはネットワーク上の任意の場所に常駐します。Web プロバイダは、事実上、どの Web テクノロジを使用しても実装できます。ただし、Oracle Application Server Portal Developer Kit では、Web プロバイダの構築タスクを簡素化する Java フレームワークが提供されます。

Web プロバイダは、デプロイ、定義および OracleAS Portal との通信に関し、XML、SOAP、HTTP または J2EE などのオープン標準を使用します。また、Web プロバイダは、J2EE コンテナへのデプロイが可能のため、OracleAS Portal リポジトリ・データベースに追加の負荷がかかりません。

注意： ほとんどの場合、Web プロバイダはデータベース・プロバイダより有益ですが、システム・レベルのカスタマイズ（デフォルトの編集）およびエンド・ユーザーによるパーソナライズのエクスポートとインポートがサポートされていません。

図 2-2 Web プロバイダ



ポートレットを開発し Web プロバイダとして公開する場合は、次のような利点があります。

- ポートレットをリモートでデプロイします。
- 既存の Web アプリケーション・コードを利用してポートレットを作成します。
- プロバイダの仕様を宣言します。
- データベース・プロバイダに比べ、豊富な機能があります。
- Web プロバイダのポートレットは、標準の Java テクノロジ（サーブレットや JSP など）を使用して開発できます。

Web プロバイダを使用してポートレットを公開するには、ポートレットを管理し、SOAP を使用して OracleAS Portal と通信できるプロバイダを作成する必要があります。Web プロバイダを使用したポートレットの公開方法については、Portal Studio (<http://portalstudio.oracle.com>) にアクセスし、「**General**」をクリックして「An Overview of Writing Portlets for Web Providers」を参照してください。

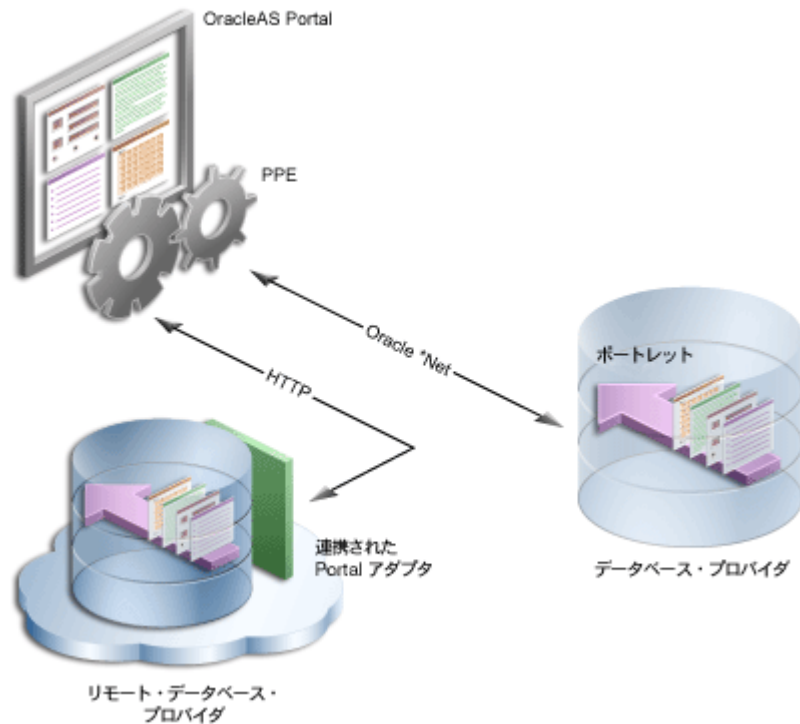
WSRP プロバイダ

Web サービスの標準である WSRP を使用すると、標準対応のコンテナと任意の WSRP ポータル間での相互運用が可能になります。アーキテクチャの視点からみると、WSRP は、Web プロバイダによく似ています。

データベース・プロバイダ

1 つ以上の PL/SQL ポートレットを所有するデータベース・プロバイダを作成することもできます。データベース・プロバイダとその PL/SQL ポートレットは、Oracle Application Server Metadata Repository データベースに常駐し、PL/SQL パッケージとして実装されます。リモート・サーバー上のデータベース・プロバイダにアクセスする場合は、連携された Portal アダプタを使用できます。詳細は、Portal Studio (<http://portalstudio.oracle.com>) の「**General**」ページにある「Understanding the Federated Portal Adapter」を参照してください。

図 2-3 データベース・プロバイダ



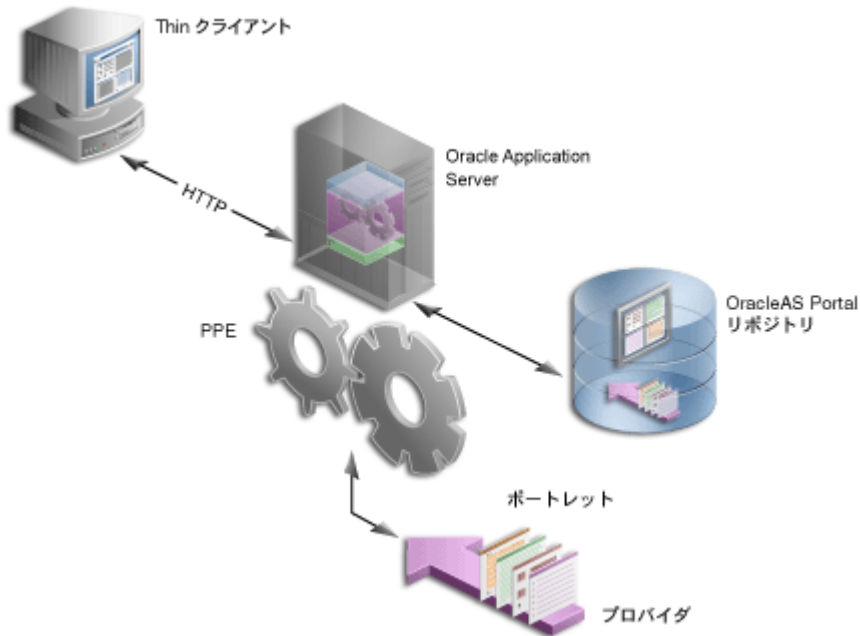
データベース・プロバイダは、PL/SQL を使用してデータ集約型の操作を実行する必要がある場合は理想的です。OracleAS Portal ユーザー・インタフェース、または PDK で提供される PL/SQL API を使用してフォームまたはチャートを構築している場合がその例です。

データベース・プロバイダを使用した PL/SQL ポートレットの公開方法については、Portal Studio にアクセスし、「**General**」をクリックして「An Overview of Writing Portlets for Database Providers」を参照してください。

注意： ポートレットの詳細は、Portal Center (<http://portalcenter.oracle.com>) の「Documentation」ページを参照してください。

プロバイダ・アーキテクチャ

図 2-4 プロバイダ・アーキテクチャ



この図は、ポートレット・プロバイダの基本的なアーキテクチャを示しています。ユーザーが Web ブラウザでポータル・ページを表示するとき、リクエストの流れは次のようになります。

1. ユーザーは、Web ブラウザのアドレス・フィールドに URL を入力することで、Web ブラウザからポータル・ページをリクエストします。
2. Oracle Application Server の中間層に常駐する Parallel Page Engine (PPE) は、ポータル・ページのレイアウト情報 (ページ・メタデータとも呼ばれます) を OracleAS Portal リポジトリから取得します。

注意： PPE は、リクエストされたポータル・ページをページ・メタデータに基づいて構築する役割を果たします。

3. PPE は、ポートレット・コンテンツのすべてのプロバイダを問い合わせます。
4. プロバイダは、ポートレットが HTML コードまたは XML コード形式のポートレット・コンテンツを生成するように、それぞれのポートレットに対して必要なコールを実行します。
5. プロバイダは、ポートレット・コンテンツを PPE に戻します。
6. PPE は、ポータル・ページを組み立て、Oracle Application Server は、このページを Web ブラウザに戻します。

注意： ポートレットおよびプロバイダ・アーキテクチャの詳細は、Portal Center (<http://portalcenter.oracle.com>) にアクセスし、「Developer Services」の下にある「Portal Studio/PDK」をクリックしてください。

Web クリッピング、OmniPortlet および Java ポートレットは、Web プロバイダを介して OracleAS Portal と通信します。OracleAS Portal をインストールすると、Web クリッピング および OmniPortlet を使用する準備が整います。つまり、これらのプロバイダは、OracleAS Portal に最初から登録されています。Java ポートレットのプロバイダは、明示的に登録する必要があります。

注意： Web クリッピングおよび OmniPortlet の開発は、急速に進んでいます。これらのポートレットの最新バージョンは、OTN からダウンロードできます。これらのツールの最新バージョンを使用する場合は、OC4J にデプロイし、Web プロバイダとして OracleAS Portal に登録する必要があります。

ポートレット・ビルダーで構築されたデータ駆動ポートレットは、データベース・プロバイダを介して OracleAS Portal と通信します。ポートレット・ビルダーのプロバイダは自動的に登録されるため、OracleAS Portal に明示的に登録する必要はありません。

PL/SQL ポートレットは、データベース・プロバイダを介して OracleAS Portal と通信します。データベース・プロバイダを明示的に登録する必要があります。

キャッシュ・スタイル

キャッシュは、ポータルのパフォーマンスを向上させるための重要な役割を果たします。OracleAS Portal では、ページ、ポートレット、スタイル、ページ・メタデータのキャッシュなど、様々なレベルのキャッシュがサポートされています。ポートレットのキャッシュは、正確な情報をタイムリな方法でユーザーに配信するために重要です。OracleAS Portal で使用できるすべてのポートレット構築テクノロジーがキャッシュをサポートしています。

OracleAS Portal では、ユーザーによるページおよびポートレットのカスタマイズがサポートされているため、ページの表示はユーザーによって異なる可能性があります。OracleAS Portal のキャッシュは、ユーザーごとに異なるコンテンツを使用できるように設計されています。したがって、ポートレットなどのポータル・オブジェクトは、ユーザー・レベルとシステム・レベルの2つのレベルでキャッシュできます。

- ユーザー・レベルのキャッシュは、特定のユーザーを対象とします。格納されたキャッシュ・エントリは、そのユーザー固有のものであり、他のユーザーはアクセスできません。ユーザー・レベルのキャッシュに適したものとして、電子メールまたは株式チャッカー・ポートレットなど、カスタマイズがサポートされたポートレットがあります。
- システム・レベルのキャッシュでは、複数のユーザーが1つのキャッシュ・エントリを共有できます。したがって、すべてのユーザーのオブジェクトのコピーをキャッシュする必要はありません。システム・レベルのキャッシュに適したコンテンツの例として、カスタマイズが不可能な新規ポートレットまたはカスタム構築されたナビゲーション・ポートレットがあります。

キャッシュを使用せずに、Web クリッピング、OmniPortlet およびポートレット・ビルダーで様々なデータソースにアクセスすると、時間がかかる場合があります。キャッシュを有効にする場合は、OracleAS Portal または OracleAS Web Cache に対して、ポートレット・コンテンツのコピーを保持するように指示します。ポートレットがリクエストされ、そのコンテンツが以前にキャッシュされている場合、ポートレットは、データソースへの問合せとそのコンテンツの再生成に時間を費やす必要がありません。以前にキャッシュされたポートレット・コンテンツが単に戻されます。

- **有効期限ベースのキャッシュ:** ポートレット・コンテンツが静的である場合、またはコンテンツの表示が最新であることが重要でない場合は、有効期限ベースのキャッシュを使用できます。有効期限ベースのキャッシュを使用する場合は、キャッシュ期間を指定する必要があります。
- **妥当性チェック・ベースのキャッシュ:** 妥当性チェック・ベースのキャッシュは、頻繁な変更または予測不能な変更がある動的コンテンツを含むポートレットに使用できます。ポートレットは、コンテンツをキャッシュ・キーに関連付け、そのキーの値をコンテンツとともに戻します。ポートレット・コンテンツがリクエストされると、ポートレットは、キャッシュ・キーに基づいて、現在のコンテンツが有効かどうかを判断します。ポートレット・コンテンツが有効な場合は、キャッシュされたコンテンツを使用できる（つまり、コンテンツが有効である）ことを示すレスポンスを戻します。有効でない場合は、新しいポートレット・コンテンツを生成し、そのコンテンツの新しいキャッシュ・キーとともに戻します。

- **失効化ベースのキャッシュ**：失効化ベースのキャッシュは、最も複雑ですが、最も柔軟性のあるキャッシュ形式です。このキャッシュは、有効期限ベースのキャッシュの効率性と、キャッシュ・コンテンツを任意の時点で失効化する機能を組み合わせたものです。OracleAS Web Cache 内のオブジェクトは、明示的に失効化されるまで有効とみなされます。

Web クリップング、OmniPortlet およびポートレット・ビルダー

Web クリップング、OmniPortlet およびポートレット・ビルダーで構築されたポートレットには、ポートレットがキャッシュされる期間を指定できます（有効期限ベースのキャッシュ）。さらに、Web クリップングおよびOmniPortletで構築されたポートレットは、エンド・ユーザーがポートレットをパーソナライズすると、自動的にリフレッシュされます。

Java ポートレット

Java ポートレットでは、有効期限ベース、妥当性チェック・ベース、失効化ベースの3種類のキャッシュがサポートされています。Java ポートレットでは、失効化ベースのキャッシュを、有効期限ベースまたは妥当性チェック・ベースのキャッシュのいずれかと組み合わせることができます。

また、ポートレットのすべてのコンテンツをキャッシュすることに加えて、Edge Side Include (ESI) を使用してポートレットのフラグメントをキャッシュできます。

PL/SQL ポートレット

Java ポートレットと同様に、PL/SQL ポートレットでも、有効期限ベース、妥当性チェック・ベース、失効化ベースの3種類のキャッシュがサポートされています。

開発ツール

この項では、ポートレット構築に使用できる開発ツールの種類について説明します。たとえば、OmniPortlet はブラウザ・ベースのウィザードで構築され、Java ポートレットは Oracle JDeveloper などのツールで構築されます。

Web クリップング、OmniPortlet およびポートレット・ビルダー

OmniPortlet、Web クリップングおよびポートレット・ビルダーでは、ブラウザ・ベースのウィザードを開発ツールとして使用します。

Java ポートレット

Java ポートレットを構築する場合、唯一の要件は JDK です。ただし、プロフェッショナルな統合開発環境 (IDE) である Oracle JDeveloper を使用することをお勧めします。他の IDE も検討できますが、PDK には、Java ポートレット・ウィザードを組み込む Oracle JDeveloper プラグインが含まれているため、Java ポートレットの開発作業を最小限にできます。

Java ポートレット・ウィザードでは、JSR168 と PDK-Java の両方のポートレットに対する開始用のスケルトンおよびファイル構造が生成されます。スケルトンに追加する必要があるのは、独自のビジネス・ロジックのみです。また、JDeveloper を使用してアプリケーションをパッケージ化し、OracleAS Containers for J2EE (OC4J) などの J2EE コンテナにデプロイできます。ポートレット・プロバイダのテストにも使用できます。デプロイする予定のプラットフォームのバージョンと一致する場合は、開発用 Java ポートレット・ランタイム環境として Oracle JDeveloper に付属している、統合されたスタンドアロン OC4J を使用できます。

PL/SQL ポートレット

PL/SQL ポートレットの開発時に、OracleAS Portal PL/SQL API をコールすることで、OracleAS Portal にアクセスする PL/SQL プログラム・ユニットを作成できます。このアクセスを可能にするには、OracleAS Portal がインストールされているデータベースと同じデータベースにプロバイダのスキーマを作成して、プロバイダおよびポートレット PL/SQL パッケージを格納します。このプロバイダのスキーマには、OracleAS Portal の PL/SQL API に対する実行権限が付与されている必要があります。

データベース・プロバイダおよび PL/SQL ポートレットの開発を容易にするために、PL/SQL ジェネレータを使用できます。PL/SQL ジェネレータは、データベース・プロバイダとその PL/SQL ポートレットに関するインストール可能な PL/SQL コードを作成する、ホスティング対象のユーティリティです。PL/SQL ジェネレータは、プロバイダおよびポートレットの定義を XML ファイル形式で受信する Web アプリケーションです。プロバイダおよびポートレットで使用する XML タグの構文は、PDK-Java で Web プロバイダを定義するために使用する XML タグのサブセットです。PL/SQL ジェネレータは、SQL*Plus から実行できる SQL スクリプトを出力します。このスクリプトには、プロバイダおよびポートレット・パッケージをインストールするための SQL コマンドが含まれています。

ホスティング対象の PL/SQL ジェネレータは、<http://portalstudio.oracle.com> の Portal Studio Web サイトから入手できます。

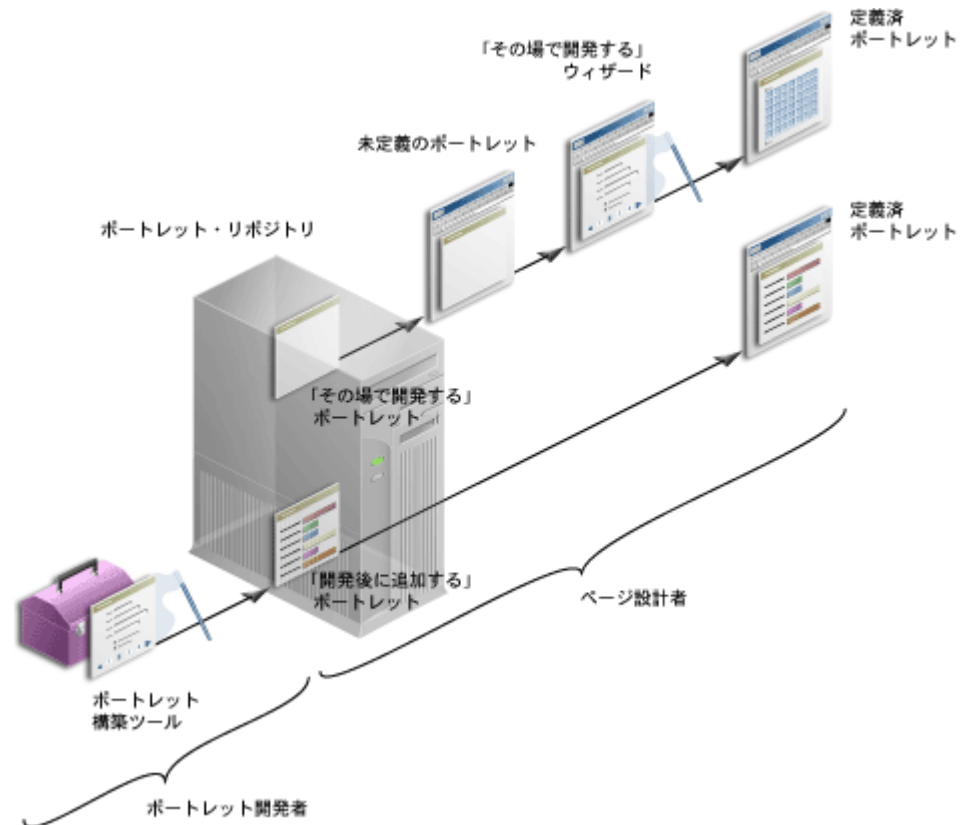
ポートレット作成スタイル

OracleAS Portal では、次の図に示すとおり、2種類のポートレット作成スタイルがサポートされています。

- 「その場で開発する」
- 「開発後に追加する」

次の図では、「開発後に追加する」ポートレットの作成は、通常、ポートレット開発者のタスクで、「その場で開発する」ポートレットの作成は、ページ設計者の責任であることも示しています。

図 2-5 ポートレット作成スタイル



OmniPortlet および Web クリッピング

OmniPortlet および Web クリッピングでは、ポートレットの作成に同じアプローチを提供します。最初に、ポータル・ページにポートレットを追加し、次に、そのページのポートレットをその場で定義します。

Java ポートレット

Java ポートレットでは、その場での開発が提供されない傾向にあります。Java ポートレットには、デフォルトの編集およびパーソナライズを簡単に追加できます。

注意： 広範囲のコーディングによって、その場で開発する Java ポートレットを作成できます。たとえば、Web クリッピングおよび OmniPortlet は、ともに Java ポートレットです。

ポートレット・ビルダー

ポートレット・ビルダーでは、最初にポートレットを定義します。以前に定義されたポートレットをポートレット・リポジトリで使用できるため、それらのポートレットをページに追加できます。ただし、単純なポートレットの場合は、OmniPortlet および Web クリッピングと同様に、ポートレット・ビルダーでもその場での開発が提供されます。

注意： ポートレット・ビルダーのその場での開発テクノロジーで構築されたポートレットは、ナビゲータを使用して構築されたポートレットに比べいくつか制限があります。

PL/SQL ポートレット

Java ポートレットと同様に、PL/SQL ポートレットでは通常、「開発後に追加する」作成方法に従います。PL/SQL ポートレットをその場で開発するには、広範囲のコーディングが必要です。たとえば、ポートレット・ビルダーで提供されているその場で開発する単純なポートレットは、PL/SQL で作成されています。

ユーザー・インタフェースの柔軟性

この項では、ポートレット構築ツールをユーザー・インタフェースの制御の観点から説明します。

Web クリップング

Web クリップングでは、その本来の機能により、常にリモート Web サイトのコンテンツが表示されます。したがって、UI の柔軟性はこのポートレットの要件ではありません。

OmniPortlet およびポートレット・ビルダー

OmniPortlet およびポートレット・ビルダーによるポートレット構築は、生産性が非常に高い一方、ユーザー・インタフェースに関してはいくつかの制限があります。

OmniPortlet では、表、チャート、ニュース、箇条書き、フォームなどのいくつかのレイアウト・スタイルから選択できます。レイアウト・スタイルによっては、ポートレットの外観を微調整できます。たとえば、表の行の背景に対して標準または代替のどちらかを選択すること、3 種類のチャート（棒、円および折れ線）から選択すること、あるいは箇条書きスタイル（黒丸、白丸、四角形、番号など）を指定することができます。また、列ラベルとその文字位置も指定できます。

Java ポートレットおよび PL/SQL ポートレット

Java ポートレットおよび PL/SQL ポートレットでは、ポートレットのユーザー・インタフェースを完全に制御できます。ポートレットでは、OracleAS Portal ページのレンダリング・ルールに準拠している HTML コンテンツを自由に生成できます。

Web サイトからコンテンツを取得する機能

この項では、ポートレット構築ツールを、他のソースからコンテンツを組み込む機能の観点から説明します。

Web クリッピング

リモート Web サイトのコンテンツを、ソース位置で表示されているとおりに表示するポートレットを作成する必要がある場合、使用する最適なツールは Web クリッピングです。Web クリッピングでは、ソース HTML ページの変更をある程度許容できます。クリップされた表がソース・ページ内のある場所から別の場所に移動した場合、Web クリッピング・エンジンは、内部の「ファジー・マッチ」アルゴリズムを使用してその表を再度検出できます。Web クリッピングで構築されたポートレットでは、リモート Web サイトへのセッションも管理できます。また、Web クリッピングでは、エンド・ユーザーによる HTML フォーム値のパーソナライズがサポートされています。

OmniPortlet

別の使用例は、リモート Web サイトでの表示方法には関係なくデータのみに関心がある場合です。取得したデータは処理し（書式設定、フィルタ処理など）、表、チャートまたはニュース形式でポートレットに表示します。この目的には、OmniPortlet が最適な選択です。OmniPortlet は、Web ページのデータをその Web データソースを使用して抽出する強力なツールです。

Java ポートレット

Java ポートレットでは、他の Java アプリケーションと同様に、下位レベルの Java ネットワーク API を常に利用して、リモート Web サイトからコンテンツを取得し、処理できます。不必要な開発作業を避けるために、Web クリッピングまたは OmniPortlet が実用的な選択肢でないことを必ず確認してから、Java を選択してください。

PL/SQL ポートレット

PL/SQL ポートレットでは、UTL_HTTP パッケージのプロシージャとファンクションを使用して、インターネット上のデータにアクセスするために Web サービスと通信できます。このパッケージは、SQL および PL/SQL から HTTP コールアウトを実行します。また、このパッケージは、HTTPS と呼ばれる Secure Sockets Layer プロトコル (SSL) 上の HTTP を、直接または HTTP プロキシを介してサポートします。他のインターネット関連のデータ・アクセス・プロトコル（ファイル転送プロトコル (FTP) や Gopher など）も、これらのプロトコルをサポートしている HTTP プロキシ・サーバーを使用することでサポートされます。

コンテンツをインラインでレンダリングする機能

ユーザーは、リンクまたはフォームのボタンなど、ポートレット内のアクティブ要素を使用してリモート URL にナビゲートできます。たとえば、ニュース・ポートレットでは、ハイパーリンクをクリックして、関心のあるニュースの詳細情報が表示されているニュース・サイトにナビゲートできます。この場合、ユーザーはポータル・ページを離れます。つまり、ブラウザ内のポータル・ページはニュース・サイトに置き換わります。

ただし、リクエストされたコンテンツを同じポートレット・コンテナ内にレンダリングすることで、ユーザーをポータル・ページのコンテキスト内に留まらせることが必要な場合があります。この場合は、同じポートレットの境界内のポータル・ページ上に、詳細なニュースを表示する必要があります。

Web クリップング

Web クリップングでは、この機能を実現するために URL リライティングがサポートされています。URL リライティングでは、ソース Web サイトを起点とするリンクを処理し、そのリンクを変更（リライト）することで、目的の機能を実現できます。

次の3つのオプションから選択できます。

- リンクをリライトしないように選択できます。この場合、リンクをクリックすると、ユーザーは Portal を離れ、クリップを提供している Web サイトに移動します。
- また、クリップに認証が必要かどうかによって、Login Server またはインラインを選択できます。Login Server を選択して、Web クリップングに対してポートレット内の URL をリライトするように指示すると、ユーザーは、必ず最初に外部 Web サイトにログインした後、先に進みます。
- インライン・オプションを選択すると、ポートレット内のすべての URL がポータル・ページを指すようにリライトされ、Web クリップング・ポートレット内でのブラウズはすべて Portal 内に留まります。

OmniPortlet

OmniPortlet では、URL リライティングは直接提供されませんが、パブリック・ポートレット・パラメータとイベントを使用することで、インライン・レンダリング機能を実現できます。イベントは、OmniPortlet が常駐しているポータル・ページにマップする必要があります。

Java ポートレット

Java ポートレット内のリンクとボタンは完全に制御できるため、インライン・レンダリング機能を簡単に実装できます。ページの URL にプライベート・ポートレット・パラメータを追加する必要があります。

ポートレット内で **Struts** を使用する場合、コンテンツは、**PDK-Struts** 統合フレームワークによって常に同じポートレット・コンテナにレンダリングされます。

ポートレットが複数の **JSP** で構成される場合（サーベイ内またはウィザード内の複数のステップなど）、ポートレットでは特別なパラメータを使用して、コンテンツのレンダリングに使用する **JSP** を実行時に指定できます。

ポートレット・ビルダー

ポートレット・ビルダーで構築されたポートレットでは、固有のインライン・レンダリングはサポートされていません。ただし、**SQL** ベースのレポートおよびチャート内にリンクを構築し、リンクが特定のポータル・ページを指すようにできます。必要に応じて、パラメータをポータル・ページに渡し、ポートレット・パラメータにマップすることもできます。

PL/SQL ポートレット

Java ポートレットと同様に、**PL/SQL** ポートレット内のアクティブ要素は完全に制御できます。したがって、プライベート・ポートレット・パラメータを実装することで、インライン・レンダリング機能をプログラムで実現できます。

チャート作成機能

この項では、ポートレット構築ツールをチャート作成機能の観点から説明します。

Web クリップング

Web クリップングでは、その本来の機能により、チャートを含む HTML コンテンツを取得および表示できますが、チャート作成機能はサポートされていません。

OmniPortlet

OmniPortlet では、棒、折れ線および円の 3 種類のチャートがサポートされています。OmniPortlet のチャートは、動的に生成されるイメージで、必要に応じてイベント対応のハイパーリンクが含まれます。

Java ポートレット

Java ポートレットでは、Oracle Business Intelligence (BI) Bean を使用して、さらに高度なチャート・ポートレットをプログラムで作成できます。

注意： Oracle Reports および Oracle Discoverer のポートレットでは、専門的なグラフを作成するために BI Bean が使用されます。

ポートレット・ビルダー

ポートレット・ビルダーでは、HTML ベースの棒グラフ・ポートレットを構築できます。多数の機能の中で、バーの色と方向を指定できます。

PL/SQL ポートレット

PL/SQL ポートレットでは、広範囲のコーディングによって HTML ベースのチャート作成機能を実現できます。

パブリック・ポートレット・パラメータのサポート

OracleAS Portal には、ページ・パラメータ、パブリック・ポートレット・パラメータおよびプライベート・ポートレット・パラメータの 3 種類があります。

- **ページ・パラメータ** : ページ・パラメータは、ページに値を渡すために使用します。ページ・パラメータを使用すると、ページがコールされた場所およびページを表示しているユーザーに応じて、ページに表示される情報を変更できます。また、ページ・パラメータを使用して複数のポートレットに同じ値を渡すと、ページ上のポートレットを同期化できます。このように、ポートレットとページ・パラメータを組み合わせることで、ページ上のポートレットを再利用および調整できる機能が提供されます。この機能がないと、異なるパラメータ値を使用するために、ポートレットを個別にコード化することが必要になります。
- **パブリック・ポートレット・パラメータ** : パブリック・ポートレット・パラメータは、ポートレットに値を渡すために使用します。ポートレット・パラメータを使用すると、ポートレットに表示される情報を、特定のページまたは特定のユーザーに対して固有にできます。ポートレット・パラメータは、ポートレット開発者によって作成され、ユーザー・インタフェースを介してページ設計者に公開されます。ページ設計者は、ポートレットをページに追加した後、パブリック・ポートレット・パラメータに値を割り当てることで、ポートレットに表示される情報をそのページ固有にできます。
ページ設計者は、特定の値（定数）、システム変数（ポータル・ユーザー名など）またはページ・パラメータを指定することで、パブリック・ポートレット・パラメータに値を割り当てることができます。ポートレットは実行時に、開発者が指定したソースから値を受信します。このように、ページ設計者は、パラメータのソースを完全に制御し、開発者は、ポートレットに送信したデータの使用方法を完全に制御できます。
- **プライベート・ポートレット・パラメータ** : プライベート・ポートレット・パラメータを使用すると、ポートレットに内部ナビゲーションを実装できます。ページがリクエストされるたびに、ポートレットにパラメータを渡すことができます。プライベート・ポートレット・パラメータは、同じポートレット・インスタンス内で排他的に渡すことができます。

パブリック・ポートレット・パラメータがサポートされているポートレットを使用すると、ページ設計者は、ポートレット・インスタンスごとにポートレットのデータ入力を調整できます。この場合、ポートレット開発者は、ポートレット・ロジックに集中でき、ページ設計者は、ポートレットを簡単に再利用してページとポートレット間の相互作用を処理できます。

この章で説明した 5 つのポートレット構築テクノロジー (OmniPortlet、Web クリップング、Java ポートレット、ポートレット・ビルダーおよび PL/SQL ポートレット) のすべてが、パブリック・ポートレット・パラメータをサポートしています。OmniPortlet、Web クリップングおよびポートレット・ビルダーでは、ウィザード・インタフェースを介して完全なサポートが提供されます。Java ポートレットに対しては、プログラムまたは Java ポートレット・ウィザードを使用してパブリック・ポートレット・パラメータのサポートを追加できま

す。PL/SQL ポートレットでは、プログラムでのみパブリック・パラメータがサポートされます。

注意： JSR 168 標準では、パブリック・ポートレット・パラメータの概念は提供されません。Java ポートレットでパブリック・ポートレット・パラメータを利用する場合は、PDK-Java を使用する必要があります。

プライベート・ポートレット・パラメータのサポート

この項では、ポートレット構築ツールをプライベート・パラメータのサポートの観点から説明します。

OmniPortlet、Web クリッピングおよびポートレット・ビルダー

OmniPortlet、Web クリッピングおよびポートレット・ビルダーでは、ポートレット開発者に対しプライベート・ポートレット・パラメータへのアクセスが提供されません。

Java ポートレットおよび PL/SQL ポートレット

Java ポートレットおよび PL/SQL ポートレットでは、プライベート・ポートレット・パラメータを使用して内部ナビゲーションを実装できます。

注意： PL/SQL ポートレットでは、プライベート・パラメータとパブリック・パラメータを同時にサポートできません。PL/SQL ポートレットのコーディング前に、どちらのパラメータ・タイプをサポートするかを決定する必要があります。

イベントのサポート

イベントは、Portal ページを起動するためのユーザー・アクションで、ポートレット開発者が定義します。ユーザー・アクションには、ポートレット内のリンクまたはボタンに対するクリックが含まれます。ページ設計者は、ページ上のポートレットでイベントが発生したときに実行する内容を指定します。イベントの発生時、ページ設計者は、現在のページを再表示したり、必要に応じてそのページのパラメータに値を渡して、ユーザーを別のポータル・ページにナビゲートすることもできます。

Web クリップング、OmniPortlet および Java ポートレット

Web クリップング、OmniPortlet および Java ポートレットでは、イベントがサポートされています。

ポートレット・ビルダーおよび PL/SQL ポートレット

ポートレット・ビルダーおよび PL/SQL ポートレットでは、イベントはサポートされていません。

ユーザー権限に基づいてポートレットの表示 / 非表示を切り替える機能

この項では、ポートレット構築ツールを認証機能のサポートの観点から説明します。

Web クリップングおよび OmniPortlet

Web クリップングおよび OmniPortlet を使用してポータル・ページ上に構築されたポートレットは、セキュリティ・マネージャを使用して表示 / 非表示を動的に切り替えることができます。Web クリップングおよび OmniPortlet では、セキュリティ・マネージャはユーザー・インタフェースを介して公開されませんが、XML プロバイダの定義ファイルを編集することで適用できます。

Java ポートレット

PDK では、Java ポートレットに対するセキュリティ・マネージャがいくつか提供されています。次に例を示します。

- **グループ・セキュリティ・マネージャ**: グループ・セキュリティ・マネージャを使用すると、特定のグループのメンバーであるユーザーにポートレットが表示されますが、メンバーでないユーザーには表示されません。
- **認証レベル・セキュリティ・マネージャ**: 認証レベル・セキュリティ・マネージャを使用すると、ユーザーの認証レベルに基づいてポートレットへのアクセスを制御できます。たとえば、ポートレットを、パブリック・ユーザーに対して非表示にし、認証されたユーザーに対して表示します。

JSR 168 ポートレットでは、標準のサブレット・メカニズムがサポートされています。

ポートレット・ビルダー

ポートレット・ビルダーでは、ポートレットへのアクセスを制御する宣言型ユーザー・インタフェースが提供されます。

PL/SQL ポートレット

PDK には、PL/SQL ポートレット内のコンテンツの表示 / 非表示を切り替える機能を実装するセキュリティ API が用意されています。

多言語サポート

この項では、ポートレット構築ツールを他の言語に対するサポートの観点から説明します。

Web クリップング、OmniPortlet、Java ポートレットおよび PL/SQL ポートレット

Web クリップング、OmniPortlet、Java ポートレットおよび PL/SQL ポートレットでは、ポータル・ユーザーが選択した言語で原文の情報が表示されます。

ポートレット・ビルダー

ポートレット・ビルダーで構築されたポートレットでは、英語のみがサポートされています。

ページ区切りサポート

比較的大きいレコードをポートレットに表示する必要がある場合は、ページ区切りサポートを使用すると便利です。

OmniPortlet

OmniPortlet では、ページ区切りはサポートされていません。

Java ポートレットおよび PL/SQL ポートレット

Java ポートレットおよび PL/SQL ポートレットでは、ページ区切りをプログラムで実装できます。

ポートレット・ビルダー

ポートレット・ビルダーには、ページ区切りの組込みサポートがあります。

シングル・サインオンおよび外部アプリケーション統合

この項では、ポートレット構築ツールを外部アプリケーションに対する認証の観点から説明します。

Web クリッピング

Web クリッピングの外部アプリケーション・フレームワークとの統合では、外部 Web サイトへのパスワードの保存を完全に自動化するメカニズムが提供されます。実行する必要があるのは、プロバイダ登録時に外部アプリケーション ID を Web クリッピング・プロバイダに関連付ける操作のみです。

OmniPortlet

OmniPortlet では、データソースがパスワードで保護されている場合、接続情報を格納できます。データソースへのアクセスに対する接続情報は、すべてのユーザーで共有するか、またはユーザーごとに保存できます。OmniPortlet では、データベース接続情報に加え、HTTP Basic 認証に使用するユーザー名とパスワードのペアも格納できます。接続情報は、Oracle データベースにある OmniPortlet の保護データ・リポジトリに格納されます。

Java ポートレット

Java ポートレットは、Oracle Internet Directory (OID) などの LDAP サーバーに加え、外部アプリケーション・フレームワークともプログラムで統合できます。

PL/SQL ポートレット

Oracle HTTP Server 上の認証モジュールである `mod_osso` を使用して、シングル・サインオンを可能にする PL/SQL ポートレットを構築できます。`mod_osso` は、パートナー・アプリケーションを統合するために初期のリリースで使用されていた Single Sign-on SDK の単純な代替モジュールです。`mod_osso` を使用すると、Single Sign-On サーバーへの単一のパートナー・アプリケーションとして機能することで、認証処理が簡素化されます。

PL/SQL ポートレットは、外部アプリケーション・フレームワークとプログラムで統合できます。

第 III 部

ポートレットの構築

第 III 部の内容は次のとおりです。

- 第 3 章「[OmniPortlet](#)を使用したポートレットの構築」
- 第 4 章「[Web クリッピング](#)を使用したコンテンツ・ベースのポートレットの構築」
- 第 5 章「[Java](#) ポートレットの構築」

OmniPortlet を使用したポートレットの構築

OmniPortlet は、Oracle Application Server Portal のサブコンポーネントです。このポートレットを使用すると、ページ設計者やコンテンツ作成者は、様々なレイアウトを使用した異なるデータソースのデータを簡単に公開できます。

データベースに格納する単純なポートレットの構築には、ポートレット・ビルダーの使用を検討してください。他のポートレット開発ツールの詳細は、第2章「ポートレット・テクノロジーのマトリックス」を参照してください。

OmniPortlet Web プロバイダを使用すると、ページ設計者やコンテンツ作成者は次の操作を実行できます。

- 複数ソース（CSV、XML、SQL など）のデータの表示
- 表示するデータのソート
- 多様なレイアウト（箇条書きリスト、チャート、フォームなど）を使用したデータの書式設定
- ポートレット・パラメータの使用
- ポートレット・イベントの起動
- ページ・ビューアへのカスタマイズ可能な設定の公開

OmniPortlet プロバイダは OracleAS Portal の一部としてインストールされます。最新リリースへのアップグレードは、Portal Studio (<http://portalstudio.oracle.com>) から Oracle Application Server Portal Developer Kit (PDK) の一部として実行できます。OmniPortlet プロバイダのインストール、構成および登録に関する指示は、JPDK および Portal Tools が格納されている `pdksoftware.zip` ファイルに用意されています。

注意： ポータルの開発の詳細は、第1章「ポートレットの概要」を参照してください。プロバイダと他のポートレット・テクノロジーの詳細は、第2章「ポートレット・テクノロジーのマトリックス」を参照してください。

この章では、OmniPortlet Web プロバイダの概要を説明し、様々なタイプの OmniPortlets の定義手順を示します。この章の内容は、次のとおりです。

- [OmniPortlet の概要](#)
- [パラメータとイベント](#)
- [OmniPortlet の使用](#)

OmniPortlet の概要

OmniPortlet の目標は、ページ設計者や開発者が、Web ベースのウィザードを使用して、多様なデータソースから様々なレイアウトのデータを迅速かつ簡単に公開できるようにすることです。OmniPortlet は、スプレッドシート（カンマで区切られた値）や XML、さらに既存の Web ページのアプリケーション・データなど、あらゆる種類のデータソースに使用できます。

パーソナライズされたデータを表示する場合は、問合せの結果をフィルタリングして、取得したデータを絞り込み、保護されたデータのアクセスに使用する資格証明情報をパラメータ化できます。複雑な設定が不要な OmniPortlet には、表、チャート、ニュース、箇条書きリストおよびフォームなど、最も一般的なポートレット用のレイアウトが用意されています。

OmniPortlet は、Oracle Application Server 10g に付属しているため、OmniPortlet は「ポートレット・ビルダー」フォルダの OracleAS Portal リポジトリから追加できます。

OmniPortlet を Oracle Application Server Portal Developer Kit の一部としてダウンロードしてある場合は、使用する前に登録する必要があります。Web プロバイダの登録方法は、Portal Center のドキュメントを参照してください。登録後は OmniPortlet を任意のポータル・ページに追加できます。

注意： ポータル・ページの構築方法の詳細は、『Oracle Application Server Portal ユーザーズ・ガイド』を参照してください。

OmniPortlet ウィザードには合計 6 つのタブがあります。OmniPortlet を最初に定義するときには、データソース・タイプ、ソース、フィルタ、表示およびレイアウトを設定します。ウィザードのこれらの設定ステップを終えた後は、Portal ページの「デフォルトの編集」アイコンをクリックしてウィザードを再開できます。ウィザードの再開時には、「ソース」、「フィルタ」、「表示」および「レイアウト」の各タブで定義を変更でき、「イベント」タブでイベント・パラメータを設定できます。

注意： OmniPortlet をデフォルトの編集モードで変更する場合は、開発者またはエンド・ユーザーがポートレットをカスタマイズしていると、変更内容がポートレットの表示モードで表示されないことがあります。デフォルトの編集モードでポートレットを変更した場合、エンド・ユーザーがその変更内容を表示するには、「カスタマイズ」ページで「デフォルトに戻す」オプションを選択する必要があります。

この項では、6 つのタブの概要（次の表を参照）と各タブの詳細な説明を示します。各タブのオプションの詳細についてはオンライン・ヘルプも参照できます。オンライン・ヘルプは、製品内の「ヘルプ」リンクをクリックするとアクセスできます。

表 3-1 Omniportlet ウィザード

タブ	説明
タイプ	データソース・オプションを提供します。ポートレットの初期定義の場合のみ表示されます。
ソース	使用する Web サービスの URL など、選択したデータソースのオプションを提供します。
フィルタ	結果の絞り込みを可能にする OracleAS Portal レベルのソート・オプションを提供します。
表示	ポートレットのヘッダーとフッターのテキスト、レイアウト・スタイルおよびキャッシュを表示するオプションを提供します。
レイアウト	レイアウトをカスタマイズするための詳細オプションを提供します。
イベント	イベントをポートレットに追加するためのオプションを提供します。表示されるのは、ポートレットがウィザードのデフォルトの編集モードで定義された場合のみです。

タイプ

図 3-1 OmniPortlet ウィザードの「タイプ」タブ



- スプレッドシート - カンマで区切られた値(CSV)を含むテキスト・ファイル
- SQL
- XML
- Webサービス
- Webページ - 既存のWebコンテンツをデータソースとして使用

注意： 追加のデータソースを Portal Center (<http://portalcenter.oracle.com>) からダウンロードしてインストールすると、そのデータソースが「タイプ」タブに表示されます。

OmniPortlet を最初に起動すると「タイプ」タブが表示され、このタブでデータソースを選択できます。複雑な設定が不要な OmniPortlet は、次のデータソースをサポートしています。

表 3-2 サポートされるデータソース

データソース	説明
スプレッドシート	カンマで区切られた値 (CSV) を含むテキスト・ファイルのデータを表示します。
SQL	SQL を使用したデータベースのデータを表示します。
XML	XML ファイルのデータを表示します。
Web サービス	標準のプロトコルを使用してインターネット上でアクセスできる個々のビジネス・サービスのデータを表示します。
Web ページ	既存の Web コンテンツに基づくデータを表示します。

ソース

データソースを選択すると、OmniPortlet ウィザードの「ソース」タブが表示されます。このタブは、選択したデータソースに対応しているため、そのデータソースが提供するオプションを指定できます。画面の最上部は指定したソース用の領域です。各「ソース」タブには、保護データにアクセスするための接続情報を定義できる「接続」セクションと、ポートレットにパラメータを定義できる「ポートレット・パラメータ」セクションがあります。ポートレット・パラメータはページ・レベルのパラメータにマップできます。

図 3-2 「ソース」タブ: 「接続」セクションと「ポートレット・パラメータ」セクション

接続

保護データにアクセスするには、データソースへの接続情報を指定する必要があります。

接続情報 <None> [接続の編集](#)

- この接続情報をすべてのユーザーが使用する
- ユーザーは接続情報を再入力する必要があります

ポートレット・パラメータ

パラメータは、ポートレットが表示されたときにページからポートレットに渡されます。ページ・プロパティを編集すると、パラメータをページ・レベル・パラメータにマップできます。

パラメータ名	デフォルト値	カスタマイズ可能	ページラブルのカスタマイズ	ページ説明のカスタマイズ
Param1	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param1"/>	<input type="text" value="Description for Paramet"/>
Param2	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param2"/>	<input type="text" value="Description for Paramet"/>
Param3	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param3"/>	<input type="text" value="Description for Paramet"/>
Param4	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param4"/>	<input type="text" value="Description for Paramet"/>
Param5	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param5"/>	<input type="text" value="Description for Paramet"/>

次の各項では、各データソースに固有の「ソース」タブ部分について説明します。

- [スプレッドシート](#)
- [SQL](#)
- [XML](#)
- [Web サービス](#)
- [Web ページ](#)


注意: 「ソース」タブのオプションの詳細は、ページの右上隅にある「ヘルプ」をクリックしてください。

スプレッドシート

スプレッドシートは、小規模なデータ・セットを格納するための一般的な方法です。OmniPortlet では、カンマで区切られた値 (CSV) をデータソースとしてサポートすることによって複数のスプレッドシートを共有できます。「ソース」タブでは、CSV ファイルの位置を指定します。保護されたサーバー上にそのファイルがある場合は、前述の「接続」セクションで接続情報を指定できます。OracleAS Portal によるファイルの読取り時に使用する文字セットも選択できます。

図 3-3 「ソース」タブ：「スプレッドシート」

スプレッドシート



CSV URL

CSV文字セット・エンコーディング


スプレッドシートの最初の行を列名に指定します。

ヒント ##ParamN##の書式(##Param1##など)を使用して、ページからURLにデータを渡すことができます。 [詳細…](#)

SQL

データを格納する最も一般的な場所は、リレーショナル・データベースです。OmniPortletでは、標準の JDBC ドライバを使用できるため、複雑な設定なしで Oracle データベースや任意の ODBC データベースにアクセスできます。ドライバのタイプは、接続情報の構成時に指定できます。

図 3-4 「ソース」タブ：「SQL」



文

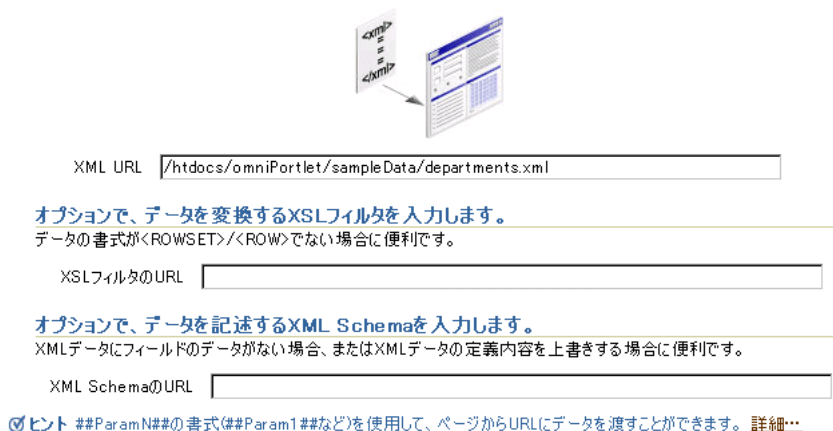
ヒント SQL問合せでテキストのかわりに##ParamN## (##Param1##など)を使用できます。変数(deptno = :p_deptなど)を使用してバインド変数を定義することもできます。 [詳細…](#)

注意： SQL データソースの使用方法は、3-28 ページの「[SQL データソースに基づくポートレットの定義](#)」を参照してください。

XML

XML は、比較的新しいデータの格納方法ですが、イントラネットやインターネット上のデータ・セットへのアクセスを制御するために使用する例が増えています。「ソース」タブでは、使用するデータを格納する XML ファイルの URL を指定できます。

図 3-5 「ソース」タブ：「XML」




XML URL

オプションで、データを変換するXSLフィルタを入力します。
データの書式が<ROWSET>/<ROW>でない場合に便利です。

XSLフィルタのURL

オプションで、データを記述するXML Schemaを入力します。
XMLデータにフィールドのデータがない場合、またはXMLデータの定義内容を上書きする場合に便利です。

XML SchemaのURL

 **ヒント** ##ParamN##の書式(##Param1##など)を使用して、ページからURLにデータを渡すことができます。 [詳細…](#)

指定した XML ファイルは ROWSET/ROW 構造でかまいません。あるいは XML スタイルシート (XSL) を用意してデータを変換できます。次のイメージは、XML データソースの ROWSET/ROW 構造の例を示しています。

図 3-6 XML データソースの ROWSET/ROW 構造

```
<TEAM>
  <EMPLOYEE>
    <DEPTNO>10</DEPTNO>
    <ENAME>KING</ENAME>
    <JOB>PRESIDENT</JOB>
    <SAL>5000</SAL>
  </EMPLOYEE>
  <EMPLOYEE>
    <DEPTNO>20</DEPTNO>
    <ENAME>SCOTT</ENAME>
    <JOB>ANALYST</JOB>
    <SAL>3000</SAL>
  </EMPLOYEE>
</TEAM>
```

この例では、<TEAM> タグが行セットを表し、<EMPLOYEE> タグが行を表しています。

OmniPortlet は、XML ファイルの形式に関係なく、自動的に XML を確認して列名を決定します。この列名は後でレイアウトの定義に使用されます。この情報を自分で指定する場合は、データを記述する XML スキーマへの URL を指定できます。

XML ファイルが HTTP Basic 認証で保護されている保護サーバー上にある場合は、他のデータソースと同様に、このデータソースの接続情報も指定できます。

Web サービス

Web サービスは、SOAP や HTTP などの標準プロトコルを使用してインターネット上でプログラム的にアクセスできる個別のビジネス・サービスです。Web サービスはプラットフォームや言語にとらわれないサービスで、通常、Web サービス・ブローカに登録されています。使用する Web サービスを検索するには、Web サービスを記述し、コール可能なメソッド、必要なパラメータおよび戻されたデータの記述などを指定する WSDL (Web サービス記述言語) への URL を取得する必要があります。

OmniPortlet は、ドキュメントと RPC (リモート・プロシージャ・コール) の 2 つのタイプの Web サービスをサポートしています。指定した WSDL は解析され、コールできる使用可能なメソッドが「ソース」タブに表示されます。

XML データソースと同様に、OmniPortlet では ROWSET/ROW 形式の Web サービス・データが必要です。ただし、データは、XSL ファイルを使用して変換することもできます。OmniPortlet は、WSDL を確認して列名を決定します。ただし、XML スキーマを指定して、戻されたデータ・セットを記述することもできます。

Web サービスが初めての場合は、最初に Oracle Web Services Center (<http://otn.oracle.com/tech/webservices/learner.html>) の『New to Web Services guide』を参照できます。

注意： Web サービス・データソースの使用方法は、3-31 ページの「[Web サービスに基づくポートレットの定義](#)」を参照してください。

図 3-7 「ソース」タブ：「Web サービス」



WSDL URL メソッドの表示

Webサービス・メソッド

このWebサービスで使用できるメソッド パラメータの表示

メソッド・パラメータの値を入力してください。

このメソッドはパラメータをとりません。 テスト

ヒント ##ParamN##の書式(##Param1##など)を使用して、ページからメソッドにデータを渡すことができます。 [詳細…](#)
オプションで、メソッド出力を変換するXSLフィルタを入力します。
 データの書式が<ROWSET>/<ROW>でない場合に便利です。

XSLフィルタのURL テスト

オプションで、メソッド出力を記述するXML Schemaを入力します。
 XMLデータにフィールドのデータがない場合、またはXMLデータの定義内容を上書きする場合に便利です。

XML SchemaのURL

ヒント ##ParamN##の書式(##Param1##など)を使用して、ページからURLにデータを渡すことができます。 [詳細…](#)

Web ページ

OmniPortlet では、既存の Web コンテンツをデータのソースとして使用し、情報をポータルに公開できます。クリップされた Web コンテンツはデータソースとして提供され、レンダリングされます。

Web ページ・データソースでは、スクレイプ機能を組み込むために、Web クリッピング・ポートレットが提供する範囲が拡張されます。次の機能もサポートしています。

- **様々なログイン・メカニズムを介したナビゲーション。** これには、フォーム・ベースと JavaScript ベースの送信、および Cookie ベースのセッション管理を使用した HTTP の Basic 認証と Digest 認証が含まれます。
- **クリッピングのファジー・マッチング。** ソース・ページ内部で Web クリッピングの順序が変更されたり、文字のフォント、サイズまたはスタイルが変更された場合でも、Web ページ・データソースでは正確に識別され、ポートレット・コンテンツとして配信されます。
- **広範囲な Web コンテンツの再利用。** これには、HTTP の GET と POST (フォーム送信) を使用して取得された、HTML 4.0.1 や JavaScript で作成されたページの基本サポートが含まれます。

すべての Web クリップング定義は、Oracle Application Server のインフラストラクチャ・データベースに永続的に格納されます。パスワードなどの保護情報は、Oracle9i の暗号化テクノロジーを使用し、DES (Data Encryption Standard) に基づいて暗号化されたフォームで格納されます。

OmniPortlet ウィザードの「ソース」タブでは、「Web ページの選択」ボタンをクリックして Web クリップング・スタジオを起動できます。Web クリップング・スタジオが起動すると、Oracle Application Server Web Clipping のオンライン・ヘルプを参照できます。

注意： Web ページ・データソースの使用方法は、3-34 ページの「[既存の Web ページに基づくポートレットの定義](#)」を参照してください。

図 3-8 「ソース」タブ：「Web ページ」

Web ページ

Web ページをデータソースとして使用できます。ポートレットが表示されるとき、データは Web ページから抽出されます。Web ページに加えられた変更は自動的にポートレットに反映されます。

Web ページを選択し、データとして使用するページの領域(クリッピング)を指定します。

Web ページの選択

ポートレット・パラメータ

パラメータは、ポートレットが表示されたときにページからポートレットに渡されます。ページ・プロパティを編集すると、パラメータをページ・レベル・パラメータにマップできます。

パラメータ名	デフォルト値	カスタマイズ可能	ページ・ラベルのカスタマイズ	ページ説明のカスタマイズ
Param1	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param1"/>	<input type="text" value="Description for Paramet"/>
Param2	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param2"/>	<input type="text" value="Description for Paramet"/>
Param3	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param3"/>	<input type="text" value="Description for Paramet"/>
Param4	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param4"/>	<input type="text" value="Description for Paramet"/>
Param5	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param5"/>	<input type="text" value="Description for Paramet"/>

フィルタ

データソースを選択し、問合せを指定した後は、OmniPortlet のフィルタリング・オプションを使用してデータをさらに絞り込むことができます。フィルタリングを効率的に使用するには、データソース・レベル（つまり、「ソース」タブ）で可能なかぎりデータをソートしてから、「フィルタ」タブのオプションを使用してデータを簡素化することをお勧めします。たとえば、SQL データソースを使用している場合は、WHERE 句を使用して指定の列から特定のデータのみを戻すことができます。この場合は、「フィルタ」タブをスキップして、ウィザードの「表示」ページに進むことができます。ただし、データソース・レベルにフィルタリング・オプションがない場合は、「フィルタ」タブのオプションを使用してデータをソートできます。

図 3-9 「フィルタ」タブ

フィルタ

このページを使用して、ポートレットに表示するデータの絞り込みと並替えを行います。

条件

データをポートレットに表示するための条件を指定します。別の列に対して条件を指定するには、プラス記号をクリックします。

列	演算子	値	操作
<None>	含む		+

すべてに一致 いずれかに一致

順序

データの表示順序を決定します。

列	順序
順序基準	<None> 昇順
次の順序基準	<None> 昇順
次の順序基準	<None> 昇順

制限

必要に応じて、ポートレットに表示される行数を制限します。

結果を制限しない
 制限 結果

表示

データの指定とソートが完了すると、OmniPortlet に対して表示オプションとレイアウトを選択できます。「表示」タブではヘッダーとフッターのテキストを追加し、「レイアウト」タブで後で調整できるレイアウト・スタイルを選択し、キャッシュを有効にできます。次のレイアウトから選択できます。

- 表
- チャート
- ニュース
- 箇条書き
- フォーム

図 3-10 「表示」タブ

ヘッダーおよびフッターのテキスト

タイトル

ヘッダー・テキスト

ヘッダー・テキストの表示

フッター・テキスト

フッター・テキストの表示

ヒント ##ParamN##の書式(##Param1##など)を使用して、ページからテキストにデータを渡し、##Column Name##の書式(##DEPTNO##など)を使用して、テキストに列の値を渡すことができます。 [詳細...](#)

レイアウト・スタイル

ポートレット・データのレイアウトを選択します。

表
 チャート
 ニュース
 箇条書き
 フォーム

Column1	Column2	Column3
Kathleen Bayyat	President - Manufacturing	100000
Robert Rodriguz	President - Sales	100000
Edward Shields	Chief Financial Officer	100000
Jan Francois Stewart	Graphic Artist	36000
Lisa Williams	Graphic Artist	36000
Sandra Kyte	Course Developer	54000
Eaulo Yau	Customer Sales Representative	39000

キャッシング

ページ・レベル・キャッシュが使用されない場合、ポートレットがキャッシュされていない場合は、ポートレットが生成されるまで常に待たなければなりません。これは、ページ生成のパフォーマンスに悪影響を与えます。

- ポートレット・コンテンツのキャッシュ 分
 ポートレット・コンテンツをキャッシュしない

注意： OmniPortlet で使用できる様々なレイアウト・スタイルの詳細は、ページの右上隅にある「ヘルプ」をクリックしてください。

レイアウト

「レイアウト」タブは、「表示」タブで選択したレイアウト・スタイルによって変わります。このタブでは、ポートレットの外観をさらにカスタマイズできます。OmniPortlet は、チャート・レイアウトのドリルダウン・ハイパーリンクをサポートしています。つまり、ユーザーがチャートの特定部分をクリックすると操作（別の URL へのジャンプなど）が実行されるように、チャートを設定できます。

他のレイアウト・スタイルの場合は、それぞれの列が特定の形式（プレーン・テキスト、HTML、イメージ、ボタンまたはフィールドなど）で表示されるように定義できます。たとえば、イメージへの URL を含むデータソースを選択したとします。このイメージを表示するには、この列の表示に対して「イメージ」を選択できます。それぞれの列も、チャートのハイパーリンクの動作と同じように、操作にマップできます。

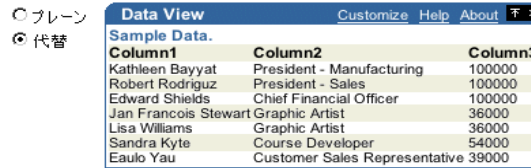
OmniPortlet で使用できるレイアウト・スタイルは、次のとおりです。

- [表レイアウト](#)
- [チャート・レイアウト](#)
- [ニュース・レイアウト](#)
- [箇条書きレイアウト](#)
- [フォーム・レイアウト](#)

表レイアウト

図 3-11 「レイアウト」タブ: 「表スタイル」

表スタイル



列のレイアウト

表に表示する列を選択してください。

名前	列ラベル	列	文字位置	表示内容	操作	URL
Field1	employee_Id	employee_Id	左揃え	テキスト	<None>	
Field2	Name	Name	左揃え	テキスト	<None>	
Field3	Gender	Gender	左揃え	テキスト	<None>	
Field4	Job	Job	左揃え	テキスト	<None>	
Field5	Email	Email	左揃え	テキスト	<None>	

🔗 ヒント ##ParamN##の書式(##Param1##など)を使用して、ページからラベルおよびURLにデータを渡し、#Column Name##の書式(##DEPTNO##など)を使用して列の値をURLに渡すことができます。 [詳細](#)

「表示」タブで選択した表スタイルは、「レイアウト」タブで調整できます。通常、データの1つ以上の列を表に表示する場合は、表レイアウトを使用します。「プレーン」を選択すると、表のすべての行が背景色なしで表示されます。「代替」を選択すると、表の行の1行おきに背景色が表示されます。

「列のレイアウト」セクションでは、ポートレットに表示するデータの列を選択し、次にそのデータの表示形式を選択できます。このセクションでは、ハイパーリンクを表示する列も設定できます。これによって、ユーザーが表内でその列をクリックすると、2番目のWebページが表示されます。

図 3-12 表レイアウトを使用した OmniPortlet の例

EMPNO	ENAME	JOB	MGR	HIREDATE
7876	ADAMS	CLERK	7788	1987-05-23
7902	FORD	ANALYST	7566	1981-12-03
7566	JONES	MANAGER	7839	1981-04-02
7788	SCOTT	ANALYST	7566	1987-04-19
7369	SMITH	CLERK	7902	1980-12-17

注意: OmniPortlet ウィザードの使用方法は、「レイアウト」タブの右上隅にある「ヘルプ」リンクをクリックしてください。

チャート・レイアウト

図 3-13 「レイアウト」タブ: 「チャート」

チャート・スタイル

棒グラフ
 円グラフ
 線グラフ

幅 ピクセル
 高さ ピクセル
 凡例

列のレイアウト

チャートに表示する列を選択してください。

列

グループ
 カテゴリ
 値

チャート・ドリルダウン

ユーザーが棒グラフ、円グラフまたは線グラフをクリックしたときに発生するアクションを選択してください。

操作 URL

チャート・レイアウトを使用すると、データを棒、円または折れ線のグラフで表示できます。「レイアウト」タブで、チャート・スタイルと列のレイアウトを選択します。列のレイアウトを選択した場合は、ラベルの基となるグループまたは列を選択できます。カテゴリはチャートの凡例の作成に使用する値を定義し、値はチャート内のバー、線またはスライスの相対サイズを決定します。

注意: チャート内の情報をグループ化するには、データ・レベル (SQL 問合せ文など) で情報をグループ化する必要があります。

図 3-14 円グラフ・レイアウト用の「レイアウト」タブの例

チャート・スタイル

棒グラフ
 円グラフ
 線グラフ

Data View Customize Help About

Sample Data.

<Value>	<Category>
21.69%	Pres
18.07%	Chief
60.24%	Graphic
	Course
	Sales

10
<Group>

幅 ピクセル
 高さ ピクセル
 凡例

列のレイアウト

チャートに表示する列を選択してください。

列

グループ
 カテゴリ
 値

チャート・ドリルダウン

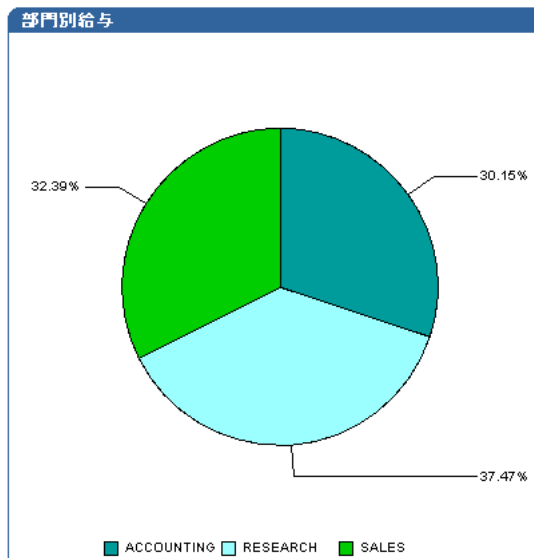
ユーザーが棒グラフ、円グラフまたは線グラフをクリックしたときに発生するアクションを選択してください。

操作
URL

バー、スライスまたは線のそれぞれが別の Web ページにリンクするように、チャートのハイパーリンクを設定できます。たとえば、ポータル・ページにチャート・ポートレットとレポート・ポートレットを表示し、次に、チャートのハイパーリンクを設定して、選択したデータに関する詳細な情報を表示するレポートの行を表示できます。

次のイメージでは、前述のイメージの「レイアウト」タブで選択したオプションの結果が表示されています。チャートの下に、凡例で使用されているカテゴリ（「レイアウト」タブの「部門名」）が表示されています。

図 3-15 円グラフ・レイアウトを使用した OmniPortlet の例



注意： データソースの数値に、書式設定された文字列、カンマまたは通貨（\$32,789.00 など）が含まれている場合、その内容はテキストとみなされ、チャートの生成時に無視されます。数値を正確に読み取るために、数値から書式設定文字を削除してください。

ニュース・レイアウト

図 3-16 「レイアウト」タブ: 「ニュース」

ニュース・スタイル



列のレイアウト

ニュースのヘッダーに表示する列を選択してください。

名前	列	表示内容	操作	URL
Field1	EMPNO	テキスト	<None>	

ニュースのコールアウトに表示する列を選択してください。

名前	列	表示内容	操作	URL
Field2	ENAME	テキスト	<None>	
Field3	JOB	テキスト	<None>	
Field4	MGR	テキスト	<None>	
Field5	HIREDATE	テキスト	<None>	

ニュース・レイアウトを使用すると、簡単な説明のある記事へのリンクをそれぞれ表示できます。このレイアウトによって、ポータル・ページに RSS や RDF などの標準 XML 形式で情報を公開できます。「列のレイアウト」セクションでは、ポートレットの最上部に表示するヘッダーを追加できます。ロゴも追加できます。また、スクロール・レイアウトを使用すると、ユーザーは垂直に移動するポートレットにすべての情報を表示できます。ユーザーがポートレットの特定のデータをクリックすると別の Web ページが表示されるように、URL も入力できます。

注意： OmniPortlet のニュース・レイアウトのスクロール・タイプは、Microsoft Internet Explorer と Netscape 7.0 でサポートされています。Netscape 4.79 では機能しません。

図 3-17 ニュース・レイアウトを使用した OmniPortlet の例

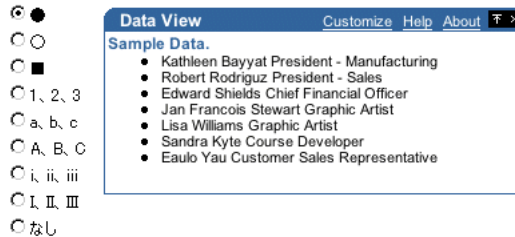
OmniPortlet	
20	Kathleen Bayyat F President - Manufacturing kbayyat@oracle.com
30	Robert Rodriguz M President - Sales rrodrigu@oracle.com
40	Edward Shields M Chief Financial Officer eshields@oracle.com

注意： OmniPortlet ウィザードの使用方法は、「レイアウト」タブの右上隅にある「ヘルプ」リンクをクリックしてください。

簡条書きレイアウト

図 3-18 「レイアウト」タブ：「簡条書き」

簡条書きスタイル



列のレイアウト

簡条書きリストに表示する列を選択してください。

名前	列	表示内容	操作	URL
Field1	EMPNO	テキスト	<None>	
Field2	ENAME	テキスト	<None>	
Field3	JOB	テキスト	<None>	
Field4	MGR	テキスト	<None>	
Field5	HIREDATE	テキスト	<None>	

簡条書きレイアウトを使用すると、簡条書きリストのデータを表示できます。「レイアウト」タブには様々な簡条書きスタイルと番号付き簡条書きスタイルが用意されています。「列のレイアウト」セクションでは、ポートレットでの列の表示方法とユーザーが列をクリックしたときに 2 番目の Web ページを表示するかどうかも選択できます。

図 3-19 簡条書きレイアウトを使用した OmniPortlet の例



注意： OmniPortlet ウィザードの使用方法は、「レイアウト」タブの右上隅にある「ヘルプ」リンクをクリックしてください。

フォーム・レイアウト

図 3-20 「レイアウト」タブ: 「フォーム」

フォームスタイル

垂直
 水平

Data View Customize Help About

Sample Data.

Name:

Job:

Salary:

Commission:

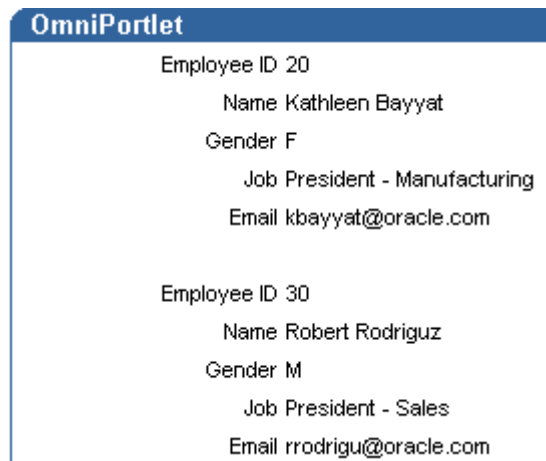
列のレイアウト

フォームに表示する列を選択してください。

名前	ラベル	列	文字位置	表示内容	操作	URL
Field1	EMPNO	EMPNO	左揃え	テキスト	<None>	
Field2	ENAME	ENAME	左揃え	テキスト	<None>	
Field3	JOB	JOB	左揃え	テキスト	<None>	
Field4	MGR	MGR	左揃え	テキスト	<None>	
Field5	HIREDATE	HIREDATE	左揃え	テキスト	<None>	

フォーム・レイアウトは、Name: <name> のように、フォーム内にラベルまたはデフォルト値として表示するデータがある場合に使用できます。この場合は、ポートレット・パラメータとイベントを使用して、選択した行にデータを渡すことができます。

図 3-21 フォーム・レイアウトを使用した OmniPortlet の例



The screenshot shows a web interface for an OmniPortlet. At the top, there is a blue header bar with the text "OmniPortlet". Below the header, the content is organized into two distinct sections, each representing an employee record. The first section is for Employee ID 20, with details: Name Kathleen Bayyat, Gender F, Job President - Manufacturing, and Email kbayyat@oracle.com. The second section is for Employee ID 30, with details: Name Robert Rodriguz, Gender M, Job President - Sales, and Email rrodrigu@oracle.com. The text is left-aligned and presented in a clean, sans-serif font.

OmniPortlet

Employee ID 20
Name Kathleen Bayyat
Gender F
Job President - Manufacturing
Email kbayyat@oracle.com

Employee ID 30
Name Robert Rodriguz
Gender M
Job President - Sales
Email rrodrigu@oracle.com

注意： OmniPortlet ウィザードの使用方法は、「レイアウト」タブの右上隅にある「ヘルプ」リンクをクリックしてください。

イベント

OmniPortlet を作成して Portal ページに戻ると、「デフォルトの編集」アイコンをクリックしてウィザードを再開できます。「イベント」タブが表示され、「ソース」タブで選択したポートレット・パラメータに基づいてイベント・パラメータを識別できます。

図 3-22 OmniPortlet ウィザードの「イベント」タブ

デフォルトの編集: OmniPortlet

適用

イベント出力

イベント出力は、ページ・パラメータとしてターゲット・ページに渡すことができます。

Event1の発生時に、以下を渡す:

Event1 Param1	=	<Null>
Event1 Param2	=	<Null>
Event1 Param3	=	<Null>

Event2の発生時に、以下を渡す:

Event2 Param1	=	<Null>
Event2 Param2	=	<Null>
Event2 Param3	=	<Null>

Event3の発生時に、以下を渡す:

Event3 Param1	=	<Null>
Event3 Param2	=	<Null>
Event3 Param3	=	<Null>

次の項の「パラメータとイベント」では、OmniPortlet のパラメータとイベントについて説明します。

パラメータとイベント

OmniPortlet は、最大 5 つのパラメータを受信し、最大 3 つのイベントを起動できます。各イベントは 1 つ以上のパラメータを送信できます。たとえば、1 つの部門の複数の従業員を表示するチャートを設定できます。ユーザーがチャートの一部（部門名など）をクリックすると、イベントが起動し、従業員に関する情報を表示するページのすべてのポートレットにパラメータが送信されます。その結果、ページ上のすべてのポートレットに、選択した部門の従業員に関する情報が表示されます。

注意： OmniPortlet でパラメータとイベントを使用する場合は、3-27 ページの「OmniPortlet の使用」の順に従ってください。provider.xml ファイルに慣れている場合は、このファイル編集してパラメータとイベントを追加できます。

パラメータとイベントを設定するには、最初に、それらを受け入れるページを有効にしておく必要があります。Oracle Application Server 10g では、パラメータとイベントはデフォルトで有効になっています。したがって、各ポートレットを設定して、必要なパラメータを受け入れ、必要なイベントを起動します。ポートレット・パラメータを設定した後は、ページ・レベルのパラメータとイベントを設定して、複数のポートレットをまとめてリンクできます。この手順は、この章の後半で詳しく説明します。

ポートレット・パラメータとイベント

1 つの OmniPortlet に対して最大 5 つのポートレット・パラメータを定義できます。この定義は次の場所で実行できます。

- OmniPortlet を定義するときに、ウィザードの「ソース」タブで定義します。
- OmniPortlet に対して「デフォルトの編集」を選択するときに、「ソース」タブで定義します。

図 3-23 「ソース」タブ：「ポートレット・パラメータ」セクション

パラメータ名	デフォルト値	カスタマイズ可能	ページ・ラベルのカスタマイズ	ページ説明のカスタマイズ
Param1	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param1"/>	<input type="text" value="Description for Paramet"/>
Param2	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param2"/>	<input type="text" value="Description for Paramet"/>
Param3	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param3"/>	<input type="text" value="Description for Paramet"/>
Param4	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param4"/>	<input type="text" value="Description for Paramet"/>
Param5	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Param5"/>	<input type="text" value="Description for Paramet"/>

パラメータ値によってポートレットに表示されるデータが決定されます。また、パラメータを使用して URL に値を渡したり、ポートレット・テキストに値を埋め込むことができます。

注意： ポートレット・パラメータの詳細は、オンライン・ヘルプを参照してください。オンライン・ヘルプにアクセスするには、**OmniPortlet** ウィザードの「ソース」タブにある「ヘルプ」リンクをクリックします。このオンライン・ヘルプでは、ポートレット・パラメータの詳細と **OmniPortlet** への設定方法を説明しています。

各 **OmniPortlet** は、最大 3 つのイベントを起動するように設定できます。各イベントは最大 3 つのパラメータを渡すことができます。それぞれのパラメータは、**Param1** などのポートレット・パラメータや **Department_No** などのデータソース列の場合があります。イベントは、**OmniPortlet** に対して「デフォルトの編集」を選択するときに、「イベント」タブで設定します。

図 3-24 「イベント」タブ

イベント出力

イベント出力は、ページ・パラメータとしてターゲット・ページに渡すことができます。

Event1の発生時に、以下を渡す:	
Event1 Param1 =	<Null>
Event1 Param2 =	<Null>
Event1 Param3 =	<Null>
Event2の発生時に、以下を渡す:	
Event2 Param1 =	<Null>
Event2 Param2 =	<Null>
Event2 Param3 =	<Null>
Event3の発生時に、以下を渡す:	
Event3 Param1 =	<Null>
Event3 Param2 =	<Null>
Event3 Param3 =	<Null>

ページ・パラメータとイベント

ポータル・ページで各 **OmniPortlet** に対してパラメータとイベントを設定した後は、そのポートレット・パラメータとイベントを同じページの他のポートレットにマップできます。ページ・パラメータとイベントの使用方法は、**OracleAS Portal** のオンライン・ヘルプおよび『**Oracle Application Server Portal ユーザーズ・ガイド**』を参照してください。

OmniPortlet の使用

次の各項では、3つのタイプの OmniPortlet（SQL データソース、Web サービス・データソースおよび Web ページ・データソース）の作成方法を示します。これらの OmniPortlet の作成方法と変更方法に加えて、ポートレット・パラメータとイベントおよびページ・パラメータとイベントを使用して、ポータル・ページに対話機能を追加する方法も説明します。

注意： OmniPortlet の特定のページやタブの詳細は、右上隅の「ヘルプ」リンクをクリックしてください。このヘルプ・トピックでは、選択したページやタブの内容を説明しています。

ポータル・ページへの OmniPortlet の追加

この項では、OmniPortlet インスタンスをポータル・ページに追加する方法を説明します。

OmniPortlet インスタンスをページに追加する手順は、次のとおりです。

1. OmniPortlet を追加するページの編集モードで、「**ポートレットの追加**」アイコンをクリックします。
2. 「ポートレットの追加」ページの「使用可能なポートレット」リストで、「ポートレット・ビルダー」リンクをクリックします。
3. 「OmniPortlet」リンクをクリックします。

このリンクが見つからない場合は、検索フィールドを使用して OmniPortlet を検索します。

4. 「OK」をクリックします。OmniPortlet の新規インスタンスがポータル・ページに表示されます。

図 3-25 ポータル・ページの OmniPortlet インスタンス



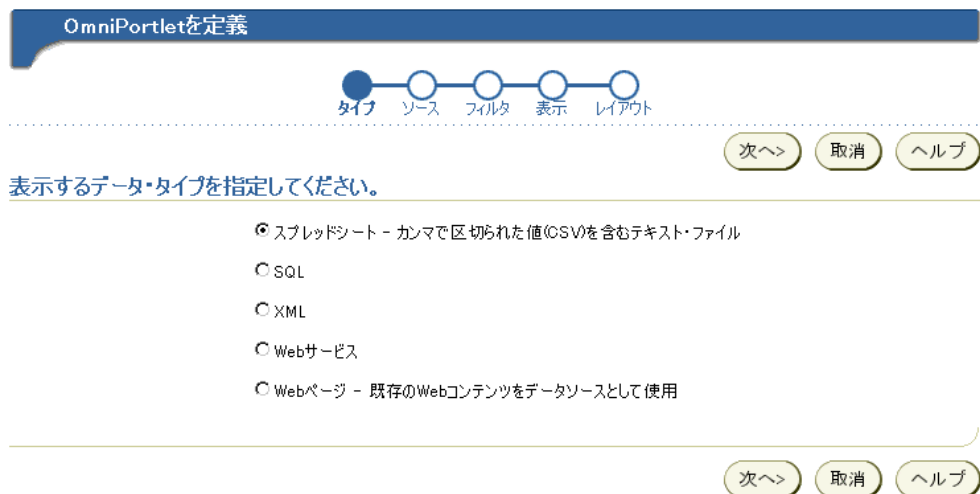
SQL データソースに基づくポートレットの定義

この項では、OmniPortlet を使用して、SQL 文を使用しているデータベースからデータを公開する方法を説明します。

ポートレットに対して SQL データソースを選択する手順は、次のとおりです。

1. 新規 OmniPortlet インスタンスで、「定義」リンクをクリックします。
2. OmniPortlet ウィザードの「タイプ」タブでは、使用するデータソースを選択できます。「SQL」ラジオ・ボタンを選択してから、「次へ」をクリックします。

図 3-26 OmniPortlet ウィザードの「タイプ」タブ



3. 「ソース」タブで、「接続の編集」をクリックして、データソースに接続情報を設定します。
4. 「接続情報」ページの「接続名」フィールドに Scott と入力します。
5. 「ユーザー名」フィールドに、サンプル・データベースのログイン ID の scott を入力します。
6. 「パスワード」フィールドに、ユーザー ID のパスワードの tiger を入力します。
7. 「接続文字列」フィールドに、データベースの接続文字列 (hostname:port:oracle_sid など) を入力します。この場合、hostname は、データベースがインストールされている位置です。
8. Oracle-thin が「ドライバ名」ドロップダウン・リストから選択されていることを確認してから、「テスト」をクリックして、接続情報が正確であることを確認します。

図 3-27 Scott 接続の接続情報

接続名 

カスタマイズ可能

ユーザー名	<input type="text" value="scott"/>	<input checked="" type="checkbox"/>
パスワード	<input type="password" value="*****"/>	<input checked="" type="checkbox"/>
接続文字列	<input type="text" value="localhost:1521:oracle_sid"/>	<input checked="" type="checkbox"/>
ドライバ名	<input type="text" value="Oracle-thin"/>	<input checked="" type="checkbox"/>

注意： 接続が機能しない場合は、データベースに対して適切な情報を入力したかどうかを確認するか、データベース管理者に連絡してください。

9. 接続の「成功」メッセージが戻った後は、「テスト」ダイアログ・ボックスを閉じてから、「OK」をクリックします。

新しい接続名が SQL データソース：ソース・タブに表示されます。

図 3-28 SQL データソース・タブの接続名

接続情報 scott (scott)

この接続情報をすべてのユーザーが使用する


ユーザーは接続情報を再入力する必要があります

10. 「文」フィールドに次のコードを入力します。

```
select dept.deptno "Dept No", dname "Department", sum(sal)
"Salary" from emp,dept
where emp.deptno = dept.deptno
group by dname,dept.deptno
```

11. 「テスト」をクリックして、データソースに対する問合せをテストします。次に「閉じる」をクリックして「テスト」ダイアログ・ボックスを閉じます。

図 3-29 「ソース」タブの SQL 文



```
select dept.deptno "Dept No", dname "Department", sum(sal) "Salary" from
emp, dept
where emp.deptno = dept.deptno
group by dname, dept.deptno
```

文

ヒント SQL 問合せでテキストのかわりに##ParamN## (##Param1##など)を使用できます。変数(deptno = :p_dept
など)を使用してバインド変数を定義することもできます。 [詳細...](#)

12. データベースに対する問合せの妥当性チェックを完了した後は、「次へ」をクリックします。
13. データは SQL 文を使用してすでにソートされているため、「フィルタ」タブの条件を設定する必要はありません。
「次へ」をクリックします。
14. 「表示」タブの「タイトル」フィールドに、「部門別給与」と入力します。
15. 「フッター・テキスト」フィールドに、「SQL データ・ソースからのデータ」と入力します。

図 3-30 「表示」タブのヘッダーおよびフッター・セクション

タイトル	<input type="text" value="部門別給与"/>
ヘッダー・テキスト	<input type="text"/>
	<input checked="" type="checkbox"/> ヘッダー・テキストの表示
フッター・テキスト	<input type="text" value="SQL データ・ソースからのデータ"/>
	<input checked="" type="checkbox"/> フッター・テキストの表示

16. 「次へ」をクリックします。

17. 「レイアウト」タブで、デフォルトの表の定義をそのまま選択し、「完了」をクリックします。OmniPortlet は、ポータル・ページに SQL 問合せの結果を表示します。

図 3-31 SQL データソースの表形式の結果

部門別給与		カスタマイズ
Dept_No	Department	Salary
30	SALES	9400
20	RESEARCH	10875
10	ACCOUNTING	8750

SQLデータソースからのデータ

この項では、OmniPortlet ウィザードを使用してポートレットを作成し、SQL データソースのデータを表レイアウトで表示する方法を説明しました。次の項では、新しい OmniPortlet のレイアウトを変更する方法を手順に従って説明します。

Web サービスに基づくポートレットの定義

この項では、Web サービスに基づいてポートレットを構築する方法を説明します。この例では、Oracle Technology Network (<http://otn.oracle.com>) 上で提供されている Web サービスを使用しますが、OmniPortlet には任意の Web サービスを使用できます。

Web サービスに基づいてポートレットを作成する手順は、次のとおりです。

1. 3-27 ページの「OmniPortlet の使用」の手順に従って、OmniPortlet のインスタンスをポータル・ページに追加します。
2. ページの編集モードで、新しい OmniPortlet インスタンスの「定義」リンクをクリックします。
3. 「タイプ」タブで、「Web サービス」ラジオ・ボタンを選択してから、「次へ」をクリックします。
4. 「ソース」タブで、Web サービス (WSDL) の定義位置を設定します。

「WSDL URL」フィールドに次の URL を入力します。

<http://otn.oracle.com/ws/oracle.otn.ws.scott.OTNDeptEmp?WSDL>

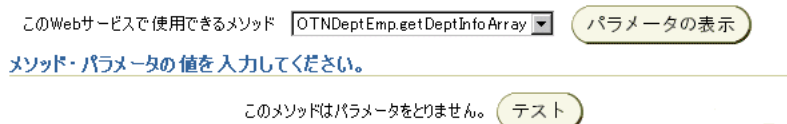
図 3-32 「ソース」タブの「WSDL URL」フィールド

WSDL URL

5. 「**メソッドの表示**」をクリックして、この Web サービスで使用可能なメソッドを表示します。
6. 「**この Web サービスで使用できるメソッド**」ドロップダウン・リストから、「**OTNDeptEmp.getEmpXML**」を選択し、次に「**パラメータの表示**」をクリックして、このメソッドが受け入れるパラメータを表示します。

注意： パラメータを含むメソッドを使用する場合は、そのパラメータがタブのこのセクションに表示されます。パラメータにサンプルの値を入力できます。次に、「**テスト**」をクリックしてサンプルの XML データ、SOAP レスポンスおよび SOAP リクエストを表示します。

図 3-33 「ソース」タブの「Web サービス・メソッド」セクション

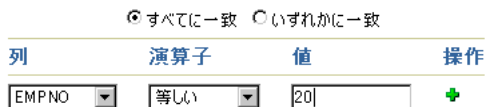


7. 「**次へ**」をクリックします。
8. 「**フィルタ**」タブで、データソースから戻されるデータを制限できます。この例では、Web サービスが EMP 表からすべてのデータを戻すため、戻されるデータを 1 つの部門のデータに制限します。

「**条件**」セクションで、次のように選択して EMPNO = 20 を設定します。

1. 「**列**」ドロップダウン・リストから、「**EMPNO**」を選択します。
2. 「**演算子**」ドロップダウン・リストから、「**等しい**」を選択します。
3. 「**値**」フィールドに、20 と入力します。

図 3-34 「フィルタ」タブの「条件」セクション



9. 「順序」セクションで、結果を従業員の名前別にアルファベット順に表示するように設定します。

「順序基準」の横の「列」ドロップダウン・リストから「ENAME」を選択し、「順序」ドロップダウン・リストに「昇順」が選択されていることを確認します。

図 3-35 「フィルタ」タブの「順序」セクション

	列	順序
順序基準	ENAME	昇順
次の順序基準	<None>	昇順
次の順序基準	<None>	昇順

10. 「完了」をクリックします。

新規ポートレットは、次のように表示されます。

図 3-36 ポータル・ページ上の Web サービス・ポートレット

OmniPortlet					カスタマイズ	閉
EMPNO	ENAME	JOB	MGR	HIREDATE		
7876	ADAMS	CLERK	7788	1987-05-23		
7902	FORD	ANALYST	7566	1981-12-03		
7566	JONES	MANAGER	7839	1981-04-02		
7788	SCOTT	ANALYST	7566	1987-04-19		
7369	SMITH	CLERK	7902	1980-12-17		

この項では、データソースとして Web サービスを使用してポートレットを作成し、OmniPortlet ウィザードの「フィルタ」タブのオプションを使用してデータをソートする方法を説明しました。次の項では、パラメータとイベントを使用して、ポータル・ページに対話機能を追加する方法を説明します。

既存の Web ページに基づくポートレットの定義

この項の手順では、既存の Web ページの情報をポータル・ページに追加する方法を示します。この例では、Oracle Web サイト (<http://www.oracle.com>) の情報を使用します。

Web ページ・データソースを使用してポートレットを作成する手順は、次のとおりです。

1. 3-27 ページの「[OmniPortlet の使用](#)」の手順に従って、OmniPortlet インスタンスをポータル・ページに追加します。
2. ページの編集モードで、新しい OmniPortlet インスタンスの「**定義**」リンクをクリックします。
3. OmniPortlet ウィザードの「**タイプ**」タブで、「**Web サービス**」ラジオ・ボタンを選択してから、「**次へ**」をクリックします。
4. 「**ソース**」タブで、「**Web ページの選択**」をクリックします。
5. 表示された Web クリップング・スタジオ・ページの「**URL ロケーション**」フィールドに <http://www.oracle.com> と入力します。

図 3-37 「URL ロケーション」フィールド

URLロケーション

6. Web クリップング・スタジオを表示するには、「**開始**」をクリックします。
7. ホーム・ページの検索フィールドに検索文字列を入力できます。この例では、omniportlet と入力して、**検索**アイコンをクリックします。

図 3-38 ホーム・ページ (www.oracle.com) での情報の検索



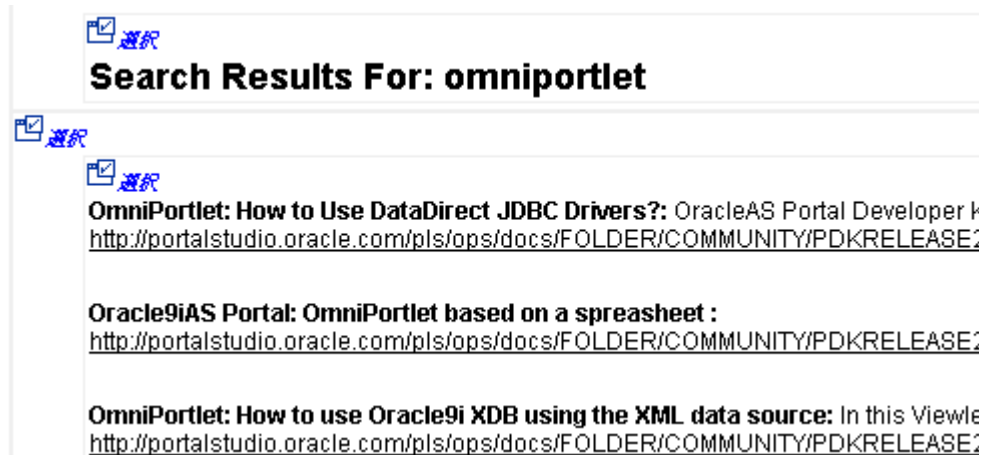
8. 検索結果が Web クリップング・スタジオに表示されます。「セクション」をクリックして、ターゲット Web ページを複数のクリップ可能なセクションに分割します。

図 3-39 ターゲット Web ページのセクション化



9. 検索結果の左上隅の「選択」をクリックして、複数の検索結果がある表を選択します。

図 3-40 検索結果の選択



10. クリップングを選択した後は、データをスクレイプ、つまり、ポートレットに表示する特定のセルを選択することによって、データをさらに絞り込むことができます。「スクレイプ」をクリックします。

図 3-41 クリップング内のデータのスクレイプ



11. スクレイプ・モードでは、各アイテムの横にあるチェック・ボックスをクリックすると、Web クリッピングのテキスト部分を識別できます。これらのアイテムは列レベル、行レベルまたは表レベルで繰り返すことができます。この例では、検索結果の各記事のタイトルと説明を表示します。タイトルと説明を行レベルで繰り返すことができます。したがって、検索で戻された各結果では、すべての結果のタイトルと説明のみが表示されます。通常は、行ごとに繰り返す部分をすべて含むテキスト・アイテムを最初の行で選択します。

この手順では、各アイテムの横のチェック・ボックスをクリックして、必要な出力を選択します。この例では、記事のタイトルと説明を表示します。

図 3-42 タイトルの選択



OmniPortlet: How to use Oracle9i XDB using the XML data source: In this Viewlet you learn how to: - use the XML <http://portalstudio.oracle.com/pls/ops/docs/FOLDER/COMMUNITY/PDKRELEASE2/PDKHOME902/PDKDEMOS2/DEMO...>

図 3-43 説明の選択

In this Viewlet you learn how to: - use the XML data source with Oracle9i - set the connection information for HTTP... <http://portalstudio.oracle.com/pls/ops/docs/FOLDER/COMMUNITY/PDKRELEASE2/PDKHOME902/PDKDEMOS2/DEMO...>

12. 出力名をわかりやすい名前に変更します。次のイメージは、新しい名前の例を示しています。

図 3-44 製品名の変更

NAME	VALUE
タイトル	In this Viewl...
説明	OmniPor...

13. 繰り返す行（この例では、記事のタイトルと説明）のベース・ユニットを定義した後は、繰り返しレベルを指定できます。「データ」セクションで、表の第 1 行目のセルを 1 つ以上選択した場合は、行、列または表の各レベルで選択したセルを繰り返すことができます。「追加」ボタンをクリックすると、繰り返しグループを識別するために、ベース・ユニットの行の全セルにあるチェック・ボックスがハイライト表示されます。この例のベース・ユニットは、「OmniPortlet:How to use Oracle9i XDB using the XML data source」というタイトルの記事です。

注意：（タイトルと説明の選択時に）手動で選択したチェック・ボックスの背景は白色ですが、「追加」ボタンで選択したチェック・ボックスの背景は、灰色（またはハイライト表示）です。

「追加」ボタンをクリックすると、列レベルで開始した繰返し操作が有効になっていることを確認できます（「追加」ボタンを最初にクリックすると1行目の全列がハイライト表示されます）。「追加」ボタンと「削除」ボタンをクリックすると、ハイライト表示されたチェック・ボックスの位置が変化する様子が見えるため、繰返し可能なすべてのアイテムを識別できます。表示するデータの正確な繰返しパターンを見つけるには、「追加」ボタンと「削除」ボタンを数回クリックする必要があります。

図 3-45 「追加」ボタンと「削除」ボタン

Data

Use to identify a cell of data.
data within the clipping. Whe:

追加削除

この例では、「追加」ボタンを2回クリックして、繰返しレベルを行に変更します。第1行目に記事のタイトルと説明が（選択したチェック・ボックスの背景の白色が示すように）そのまま選択されているのがわかります。後続の行でも、記事のタイトルと説明が（選択したチェック・ボックスの灰色の背景が示すように）選択されています。これは、繰返しレベルが行レベルであることを示しています。

図 3-46 ハイライト表示された行レベル（タイトル）のチェック・ボックス

OmniPortlet: How to use Oracle9i XDB using the XML data source: In this Viewlet you learn how to: - use t
<http://portalstudio.oracle.com/pls/ops/docs/FOLDER/COMMUNITY/PDKRELEASE2/PDKHOME902/PDKDEMO:>



OmniPortlet: XML Data Source :
<http://portalstudio.oracle.com/pls/ops/docs/FOLDER/COMMUNITY/PDKRELEASE2/PDKHOME902/PDKDEMO:>



OmniPortlet: How to use the SQL data source: This viewlet show you how you can easily publish data from
<http://portalstudio.oracle.com/pls/ops/docs/FOLDER/COMMUNITY/PDKRELEASE2/PDKHOME902/PDKDEMO:>



OmniPortlet: How to publish and Filter a CSV file:
<http://portalstudio.oracle.com/pls/ops/docs/FOLDER/COMMUNITY/PDKRELEASE2/PDKHOME902/PDKDEMO:>

図 3-47 ハイライト表示された行レベル（説明）のチェック・ボックス

How you can easily publish data from a database using the SQL data source. Also you will learn...
http://portalstudio.oracle.com/pls/ops/docs/FOLDER/COMMUNITY/PDKRELEASE2/PDKHOME902/PDKDEMOS2/DEMOSOURCE/PORTALTOOLS/OMNIPORTLET_SQL_DATASOURC:

http://portalstudio.oracle.com/pls/ops/docs/FOLDER/COMMUNITY/PDKRELEASE2/PDKHOME902/PDKDEMOS2/DEMOSOURCE/PORTALTOOLS/OMNIPORTLET_FILTER_CSV/OMNI:

Events:
<http://portalstudio.oracle.com/pls/ops/docs/FOLDER/COMMUNITY/PDKRELEASE2/PDKHOME902/PDKDEMOS2/DEMOSOURCE/PORTALTOOLS/OMNIPORTLET3/OMNIPORTLETPA:>

14. 「続行」アイコンをクリックします。

図 3-48 「続行」アイコン



- クリッピング属性セクションで、このポートレットのタイトルと説明を更新できます。したがって、ユーザーはこのポートレットに含まれているクリップされた情報を確認できます。また、Web クリッピングのコンテンツ・ソースとの接続を確立するためにポートレットが待機する接続のタイムアウト、つまり接続時間（秒）も変更できます。

注意： Web クリッピング・スタジオの使用方法は、右上隅の「ヘルプ」アイコンをクリックして、オンライン・ヘルプを表示してください。

「タイトル」フィールドに、OmniPortlet Articles と入力します。

- 「説明」フィールドに、Articles and descriptions on Oracle.com と入力します。
- 「タイムアウト(秒)」フィールドに、60 と入力します。

Web クリッピング・スタジオのクリッピング属性セクションは、次のようなイメージになります。

図 3-49 Web クリッピング・スタジオのクリッピング属性セクション

Clipping Attributes

タイトル	<input type="text" value="OmniPortlet Article"/>
説明	<input type="text" value="Articles and descriptions on Oracle.com"/>
タイムアウト(秒)	<input type="text" value="60"/>

- 「OK」をクリックします。
- OmniPortlet ウィザードの「ソース」タブに、新規タイトルと説明が表示されます。Web クリッピング・スタジオで Web クリッピングを編集する場合は、「Web ページの選択」ボタンを再度クリックできます。
「完了」をクリックします。
- 新規 Web ページ・ポートレットがポータル・ページに、次のようなイメージで表示されます。

図 3-50 既存の Web ページ・コンテンツに基づくポートレット



OmniPortlet		カスタマイズ 下 ×
タイトル	説明	
HTML DB: How to make an omniportlet parameter driven:	A brief tutorial describing the steps required to configure a portlet to be driven by a parameter in another portlet....	
Oracle9iAS Portal: OmniPortlet based on a spreadsheet :		
OmniPortlet based on a Web Service :		
OmniPortlet: XML Data Source :		
OmniPortlet: How to use Oracle9i XDB using the XML data source:	In this Viewlet you learn how to: - use the XML data source with Oracle9i\ - set the connection information for HTTP...	
OmniPortlet & Simple Parameter Form portlet : Page Parameters and Events:		
OmniPortlet: How to use the SQL data source:	This viewlet show you how you can easily publish data from a database using the SQL data source. Also you will learn...	
Oracle9iAS Portal - OmniPortlet - Web Page Data Source:		
OmniPortlet: How to publish and Filter a CSV file:		
No Title	A brief tutorial describing the steps required to publish an HTML DB report as a chart using an OmniPortlet....	

既存の OmniPortlet のレイアウトの変更

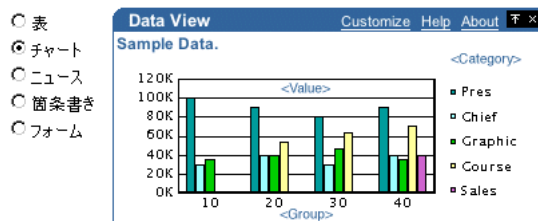
ポートレットを作成した後で、その外観を変更する場合があります。OmniPortlet では、reentrant ウィザードを使用して変更できます。

この項では、前項で作成したポートレットのレイアウトを変更する方法を説明します。

OmniPortlet のレイアウトを変更する手順は、次のとおりです。

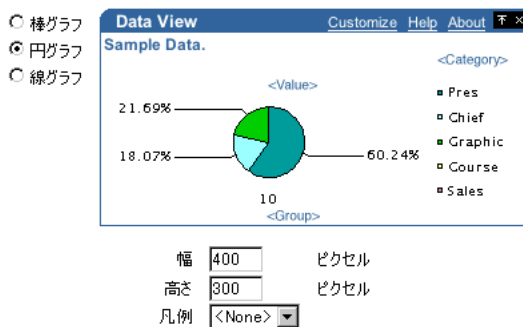
1. OmniPortlet を追加したページの編集モードで、OmniPortlet の「**デフォルトの編集**」アイコンをクリックします。
2. デフォルトの編集モードでは、必要なタブをクリックすると、OmniPortlet の様々な部分を変更できます。
レイアウトを変更するには、最初に「表示」タブをクリックします。
3. 「レイアウト・スタイル」の下に「**チャート**」ラジオ・ボタンを選択し、次に「**レイアウト**」タブをクリックしてレイアウトの定義を変更します。

図 3-51 「表示」タブの「レイアウト・スタイル」セクション



4. 「レイアウト」タブの「チャート・スタイル」セクションで、「**円グラフ**」ラジオ・ボタンを選択し、円グラフを表示します。次に「凡例」ドロップダウン・リストから「一番下」を選択してチャートの凡例を円グラフの下に表示します。

図 3-52 「レイアウト」 タブの「チャート・スタイル」 セクション



- 「列のレイアウト」セクションで、表示または非表示にする列を選択します。

「グループ」ドロップダウン・リストから「<None>」を選択します。その他の列はそのままにしておきます。

図 3-53 「レイアウト」 タブの「列のレイアウト」 セクション

列

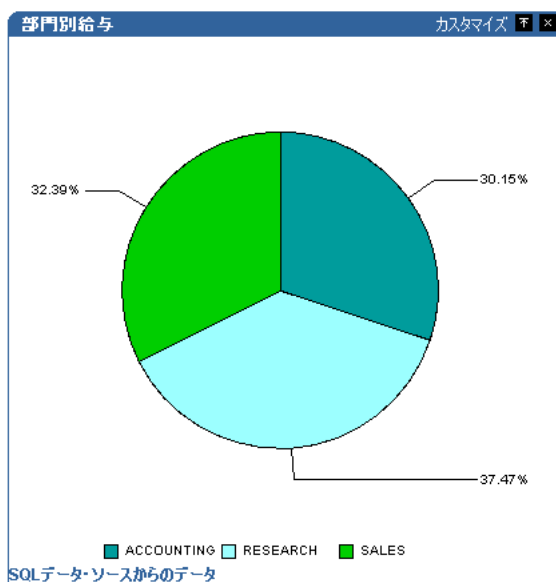
グループ <None>

カテゴリ Department

値 Salary

- 「OK」をクリックして変更内容を受け入れ、ポータル・ページに戻ります。
新規 SQL ポートレットは、次のように表示されます。

図 3-54 円グラフとして表示する SQL データソースの OmniPortlet



この項では、OmniPortlet ウィザードに戻ってポートレットのレイアウトを変更する方法と、ポートレットのレイアウトを表形式レポートから円グラフに変更する方法を説明しました。

パラメータとイベントの使用

値をページやページ上のポートレットに渡す場合は、ページ・パラメータを使用します。ページ・パラメータを受け入れるように構成されているポートレットでは、パラメータとイベントのコンテキスト・ワイヤリングに基づいて、別の操作をトリガーできます。たとえば、株式チャッカーを表示するポートレットと特定の会社に関するニュースを表示するポートレットを表示するページを設定できます。ページ・パラメータは銘柄記号を値として受け入れ、関連する株式情報とニュース情報を適切なポートレットに表示します。

ページ内のパラメータを使用できるのは、「パラメータおよびイベントを有効にする」ページ・グループの設定が有効な場合のみです。ページ・グループ内のパラメータとイベントを有効化する方法は、『Oracle Application Server Portal ユーザーズ・ガイド』の第4章「ページ・グループの使用」を参照してください。

注意： ページ・パラメータとイベントの詳細は、『Oracle Application Server Portal ユーザーズ・ガイド』第7章の「パラメータとイベントの使用」を参照してください。

この項では、ページを構成してページ・パラメータとイベントをサポートする方法を説明します。次にポートレット・パラメータとイベントを設定し、この2つをページ・パラメータとイベントに統合します。

この項の終わりには、3-41 ページの「[既存の OmniPortlet のレイアウトの変更](#)」で作成した円グラフのセクション上をエンド・ユーザーがクリックできるようになります。この操作では、選択した部門に関する情報が 3-31 ページの「[Web サービスに基づくポートレットの定義](#)」で作成した Web サービス・ポートレットに表示されます。

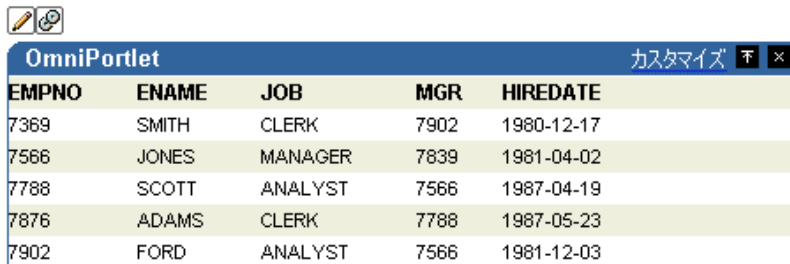
既存の OmniPortlet へのパラメータの追加

この項の手順では、3-31 ページの「[Web サービスに基づくポートレットの定義](#)」で作成した Web サービス・ポートレットにパラメータを追加する方法を示します。

パラメータを受け入れるように OmniPortlet を構成する手順は、次のとおりです。

1. ポータル・ページの編集モードで、Web サービス・ポートレットの左上隅の「**デフォルトの編集**」アイコンをクリックします。

図 3-55 Web サービス・ポートレットの「デフォルトの編集」アイコン



The image shows a small icon with a pencil and a circular arrow, representing the 'Default Edit' function. Below it is a screenshot of a table titled 'OmniPortlet' with a 'カスタマイズ' (Customize) button in the top right corner. The table contains the following data:

EMPNO	ENAME	JOB	MGR	HIREDATE
7369	SMITH	CLERK	7902	1980-12-17
7566	JONES	MANAGER	7839	1981-04-02
7788	SCOTT	ANALYST	7566	1987-04-19
7876	ADAMS	CLERK	7788	1987-05-23
7902	FORD	ANALYST	7566	1981-12-03

2. 表示された「ソース」タブで、「ポートレット・パラメータ」セクションを見つけて、次のパラメータを設定します。
 - Param1 のデフォルト値を 20 に設定します。
 - Param2 のデフォルト値を RESEARCH に設定します。

図 3-56 「ソース」タブの「ポートレット・パラメータ」セクション

パラメータ名	デフォルト値	カスタマイズ可能	ページラベルのカスタマイズ	ページ説明のカスタマイズ
Param1	20	<input type="checkbox"/>	Param1	Description for Paramet
Param2	RESEARCH	<input type="checkbox"/>	Param2	Description for Paramet

3. 画面の最上部の「フィルタ」タブをクリックします。
4. 既存の条件を次のように変更します。

DEPTNO 等しい ##Param1##

図 3-57 「フィルタ」タブの「条件」セクション

すべてに一致
 いずれかに一致

列	演算子	値	操作
DEPTNO	等しい	##Param1##	+

注意： 値を ##Param1## に変更することによって、ポートレットは、DEPTNO 列の値に対して Param1（「ソース」タブで作成した）の値を使用するように設定されます。たとえば、Param1 が 10 の場合、DEPTNO は 10 に設定されます。

5. 「表示」タブをクリックします。
6. 選択した部門に応じて更新するポートレット・ヘッダーにテキストを追加します。
「ヘッダーおよびフッターのテキスト」セクションの「ヘッダー・テキスト」フィールドに、次のように入力します。

##Param2## 部門の社員のリスト

図 3-58 「表示」タブのヘッダーおよびフッター・セクション

タイトル	OmniPortlet
ヘッダー・テキスト	##Param2##部門の社員のリスト
	<input checked="" type="checkbox"/> ヘッダー・テキストの表示
フッター・テキスト	
	<input checked="" type="checkbox"/> フッター・テキストの表示

7. 「OK」をクリックして、Web サービス・ポートレットに対する変更内容を保存します。

この項では、ポートレット・パラメータの設定方法とそのパラメータをポートレットの機能に統合する方法を説明しました。ポートレットには、Department 20 のデータの表示に加えて、使用したパラメータに指定した部門番号の情報が表示され、使用した 2 番目のパラメータに指定した部門に基づいてヘッダー・テキストも更新されます。SQL データソース・ポートレットにイベントを設定した後は、ページ・パラメータをこの項で作成したポートレット・パラメータにマップします。

次の項では、3-28 ページの「SQL データソースに基づくポートレットの定義」で作成した SQL ポートレットにイベントを追加する方法を示します。次に、イベントの設定が完了した後、ポータル・ページで一緒に機能するようにパラメータとイベントを構成します。

既存の OmniPortlet へのイベントの追加

この項の手順では、同じページの別のポートレットにパラメータを設定するときにイベントを起動する方法を示します。

ポートレットにイベントを設定する手順は、次のとおりです。

1. ページが編集モードのときに、「部門別給与」ポートレット（円グラフを表示します）の左上隅にある「**デフォルトの編集**」アイコンをクリックします。
2. 「**レイアウト**」タブをクリックします。
3. 「チャート・ドリルダウン」セクションの「操作」ドロップダウン・リストから「**Event1**」を選択します。これによって、ユーザーが円グラフのセクションをクリックすると、Event1 が発生します。
4. 「**イベント**」タブをクリックします。
5. 「Event1 の発生時に、以下を渡す」で、次のオプションを設定します。
 - Event1Param1 = Dept No
 - Event1Param2 = Department

図 3-59 「イベント」タブの「イベント出力」セクション

Event1 の発生時に、以下を渡す:

Event1 Param1	=	DeptNo	▼
Event1 Param2	=	Department	▼
Event1 Param3	=	<Null>	▼

このセクションでは、円グラフの一部をクリックするとイベントが起動するように SQL OmniPortlet を設定しました。ユーザーがチャートの一部をクリックすると、OmniPortlet は、選択された部門の番号と名前を取得し、その情報を 2 つのイベント・パラメータ (Event1Param1 と Event1Param2) に渡します。

次の項では、3-44 ページの「既存の OmniPortlet へのパラメータの追加」で作成したパラメータとこの項で作成したイベントをページ・パラメータとイベントにそれぞれマップすることによって、ポータル・ページで一緒に動作するように設定します。

ページ上でのポートレット・パラメータとイベントの関連付け

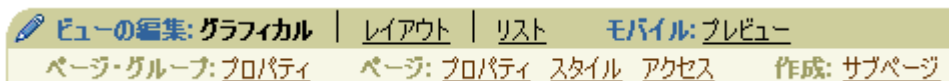
引き渡すデータを制御するポートレットにパラメータを作成し、イベントを設定した後は、ページ・パラメータとイベントを設定して、ポータル・ページでこの2つポートレットを関連付ける必要があります。

この項の手順では、ページ・レベルでパラメータを設定する方法を示します。したがって、Web サービス・ポートレットに設定されたパラメータは、SQL ポートレットに渡され、次に適切なイベントが起動します。

ページ・パラメータとイベントを設定する手順は、次のとおりです。

1. ページの編集モードで、ページの左上隅の「ページ: プロパティ」リンクをクリックします。

図 3-60 「ページ・グループ: プロパティ」リンクの横にある「ページ: プロパティ」リンク



2. 「パラメータ」タブをクリックします。
3. 「新規ページ・パラメータ」セクションの「パラメータ名」フィールドに、dept_no と入力し、「追加」をクリックします。
4. 新規 dept_no パラメータの横にある「ページ・パラメータのプロパティ」セクションの「表示名」フィールドに Department Number と入力し、「デフォルト値」フィールドに 20 と入力します。
5. 前述の2つの手順に従って、dept_name という2つ目のページ・パラメータを作成し、表示名を Department Name に、デフォルト値を RESEARCH に設定します。

図 3-61 「パラメータ」タブの「ページ・パラメータのプロパティ」セクション

名前	表示名	デフォルト値	カスタマイズ可能	説明
✗ dept_no	Department Number	20	<input checked="" type="checkbox"/>	
↑ ✗ dept_name	Department Name	RESEARCH	<input checked="" type="checkbox"/>	

- 「ポートレット・パラメータ値」セクションで、ポートレット・パラメータを含む作成済の Web サービス・ポートレットを検索します。この章の手順を実行するために、新しいページを特別に作成していた場合、このポートレットは、リストにある「ポートレット : OmniPortlet」の 2 番目のインスタンスになります。

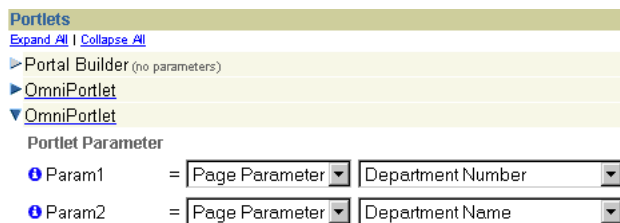
「ポートレット : OmniPortlet」の左にある矢印をクリックすると、そのポートレットのポートレット・パラメータが表示されます。

- Param1 の横のドロップダウン・リストから「ページ・パラメータ」を選択し、次に、表示されたドロップダウン・リストから「Department Number」を選択します。

ポートレット・パラメータ Param1 が前の手順で作成した dept_no ページ・パラメータにマップされます。

- Param2 ポートレット・パラメータを Department Name ページ・パラメータにマップします。

図 3-62 ページの「パラメータ」タブの「ポートレット・パラメータ値」セクション



注意： これで、3-44 ページの「既存の OmniPortlet へのパラメータの追加」で Web サービス・ポートレット用に作成したポートレット・パラメータは、dept_no (Department Number) および dept_name (Department Name) というページ・レベルのパラメータにマップされます。


- 画面の最上部の「イベント」タブをクリックします。
- 「ポートレット : Omniportlet」の最初のインスタンスをクリックします。
- 「ポートレット : Omniportlet」の下の Event1 がハイライト表示されていることを確認します。
- 「このイベントが発生する時」の下の「ページに移動」が選択されていることを確認してから、ドロップダウン・リストの横の「リスト」アイコンをクリックします。

13. リストでページ名を検索し、名前の横の「オブジェクトを戻す」リンクをクリックします。

ページ名が「ページに移動」リストに表示されます。

図 3-63 ポータル・ページの表示をポートレット Event1 に設定

When this event is raised:

- Go to page: 
- Redisplay page: OmniPortlet Hands-On

14. ポータル・ページを選択すると、ページ・パラメータが「ページ入力」セクションに表示されます。

「Department Number」の横のドロップダウン・リストから「イベント出力」を選択し、次に、表示されたドロップダウン・リストから「Event1Param1」を選択します。これで、Event Parameter 1 のコンテンツを表示するように Department Number ページ・パラメータが設定されました。

15. Event Parameter 2 のコンテンツを表示するように Department Name ページ・パラメータを設定します。

「ページ入力」セクションは、次のようなイメージになります。

注意： これで、3-46 ページの「既存の OmniPortlet へのイベントの追加」で SQL データソース・ポートレット用に作成したイベント・パラメータは、ポートレット・パラメータにリンクされているページ・パラメータにマップされます。ページ上の 2 つのポートレットは、リンクされインタラクティブになっています。

図 3-64 ポータル・ページの「イベント」タブの「ページ入力」セクション

Pass data to input parameters of page **OmniPortlet Hands-On:**

Page Input

- ③ Department Number =
- ③ Department Name =

16. 「OK」をクリックして変更内容を受け入れます。

17. 円グラフの一部をクリックして変更内容をテストします。たとえば、「Accounting」セクションをクリックすると、Web サービス・ポートレットでは会計部門に関する情報が自動的に更新されます。

この項では、ページ・パラメータとイベントを使用して1つのポータル・ページ上で複数のポートレットをリンクする方法を説明しました。

まとめ

この章では、OmniPortlet のツールの使用方法とページ上に各種のポートレットを構築する方法を説明しました。さらに、パラメータとイベントを使用してページ上でポートレットを統合し、ポータル・アプリケーションを作成する方法も説明しました。

OmniPortlet の多様なツールの使用方法は、ウィザードの各ページの「ヘルプ」リンクをクリックしてください。Web クリップिंगの使用方法は、[第4章「Web クリップングを使用したコンテンツ・ベースのポートレットの構築」](#)を参照してください。

Web クリッピングを使用したコンテンツ・ベースのポートレットの構築

Web クリッピング・プロバイダは、OracleAS Portal に対するポートレット・プロバイダです。このプロバイダは、Web クリッピング・ポートレットを介して、クリップした Web コンテンツをポートレット・コンテンツとして提供およびレンダリングします。

Web クリッピング・ポートレットでは、Web コンテンツを単一の一元化したポートレットに収集できます。このポートレットを使用すると、大規模な組織に散在する数百の Web サイトのコンテンツを統合できます。

Web クリッピング・ポートレットは次の機能をサポートします。

- **様々なスタイルのログイン・メカニズムを介したナビゲーション。**これには、フォーム・ベースと JavaScript ベースの送信、および Cookie ベースのセッション管理を使用した HTTP の Basic 認証と Digest 認証が含まれます。
- **クリッピングのファジー・マッチング。**ソース・ページ内部で Web クリッピングの順序が変更されたり、文字のフォント、サイズまたはスタイルが変更された場合でも、OracleAS Portal では正確に識別され、ポートレット・コンテンツとして配信されます。
- **広範囲な Web コンテンツの再利用。**これには、HTTP の GET と POST (フォーム送信) を使用して取得された、HTML 4.0.1、JavaScript、アプレットおよびプラグイン対応コンテンツで作成されたページの基本サポートが含まれます。
- **カスタマイズ。**ページ設計者による Web クリッピングのカスタマイズを可能にし、ページの閲覧者がポートレットのカスタマイズ時に変更できる入力パラメータを公開します。また、これらのパラメータは、ページ設計者がページ・パラメータにさらにマップできるポートレット・パラメータにマップされます。
- **外部 Web サイトからの情報の使用。**これには、外部アプリケーションとの統合が含まれます。この統合によって、Oracle Application Server Single Sign-On を利用して、認証されている外部 Web サイトのコンテンツをクリップできます。

-
- **インライン・レンダリング。**このレンダリングによって、Web クリップング・ポートレットは、ポートレットのコンテキスト内にリンクを表示するように設定できます。この設定によって、ユーザーが Web クリップング・ポートレット内でリンクをクリックすると、同じポートレット内に結果が表示されます。この機能は、内部および外部の Web サイトで使用できます。
 - **URL ベースのポートレットからの移行。**URL ベースのポートレットから Web クリップング・ポートレットへの移行を可能にします。詳細は、Portal Center (<http://portalcenter.oracle.com>) を参照してください。

すべての Web クリップング定義は、Oracle Application Server のインフラストラクチャ・データベースに永続的に格納されます。パスワードなどの保護情報は、Oracle の暗号化テクノロジーを使用し、Data Encryption Standard (DES) に基づいて暗号化されたフォームで格納されます。

Web クリップング・プロバイダのインストール、構成および登録の方法については、次のファイルを参照してください。

<http://host:port/portalTools/webClipping/htdocs/help/configuring.webclipping.html>

この章の内容は、次のとおりです。

- [ページへの Web クリップング・ポートレットの追加](#)
- [Web クリップング・ポートレットに表示する Web ページのセクションの選択](#)
- [Web クリップング・ポートレットのプロパティの設定](#)
- [ポートレットへの外部アプリケーションの統合](#)
- [例: ページへの Portal Center 情報の追加](#)
- [Web クリップングに関する現在の制限](#)
- [Web クリップングのトラブルシューティング](#)

ページへの Web クリップング・ポートレットの追加

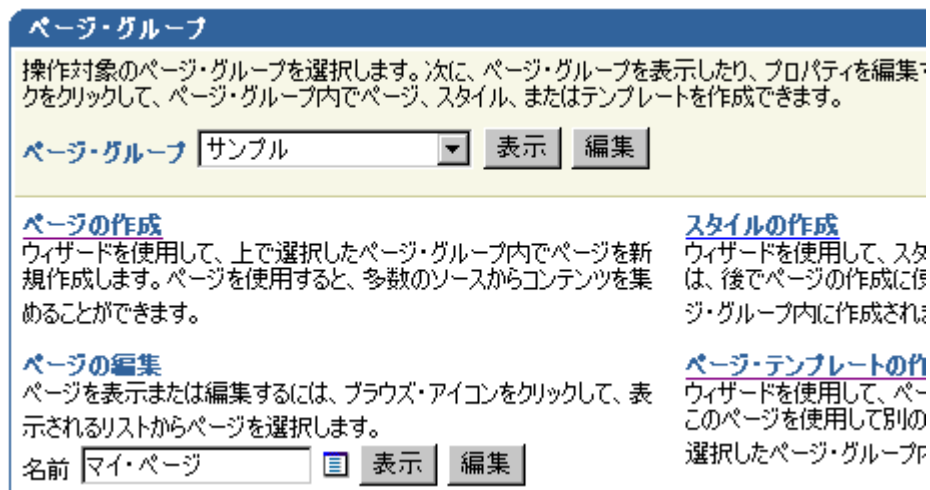
Web ページのコンテンツをポータル・ページに追加するには、Web クリップング・ポートレットをポータル・ページに追加し、Web クリップング・スタジオを使用して目的のコンテンツを含む Web ページにナビゲートしてから、そのページのクリップ部分を選択します。

Web クリップング・ポートレットをページに追加する手順は、次のとおりです。

1. 「ページ・グループ」ポートレットの「ページの編集」セクションで、「ページをブラウズ」アイコンをクリックし、Web クリップング・ポートレットを追加するページを選択します。

デフォルトでは、「ページ・グループ」ポートレットは、「Portal ビルダー」ページの「構築」タブにあります。図 4-1 に、そのページを示します。

図 4-1 ページの選択



2. 「編集」をクリックします。
3. Web クリップング・ポートレットを追加する領域で、「ポートレットの追加」アイコンをクリックします。

図 4-2 に、そのページの一部を示します。

図 4-2 ページへのポートレットの追加



4. 「ポートレットの追加」 ページで、「Web クリップング・ポートレット」 リンクにナビゲートし、それをクリックします。
デフォルトでは、Web クリップング・ポートレットは、ポートレット・リポジトリの「Portal コンテンツ・ツール」 ページにあります。このページが見つからない場合は、検索フィールドを使用してポートレットを検索します。
5. 「OK」 をクリックして、ページに Web クリップング・ポートレットを追加します。
図 4-3 に、ページに追加された Web クリップング・ポートレットを示します。

図 4-3 ページに追加された Web クリップング・ポートレット



Web クリップング・ポートレットに表示する Web ページのセクションの選択

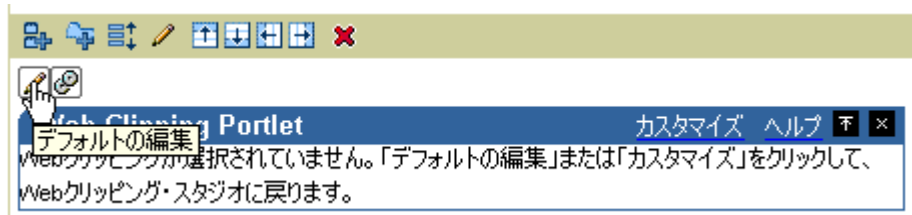
Web クリップング・スタジオを使用すると、次の操作が可能です。

- Web コンテンツのブラウズ
- 選択したターゲット・ページのセクション化
- Web コンテンツの正確なクリップ部分の選択
- ポートレットのクリップされたコンテンツのプレビュー
- ポートレットのクリップされたコンテンツの保存
- ポートレット・プロパティの設定と更新されたポートレット情報の保存

Web クリップング・ポートレットに表示する Web ページのセクションを選択する手順は、次のとおりです。

1. 図 4-4 に示すように、Web クリップング・ポートレットの上にある「デフォルトの編集」アイコンをクリックします。

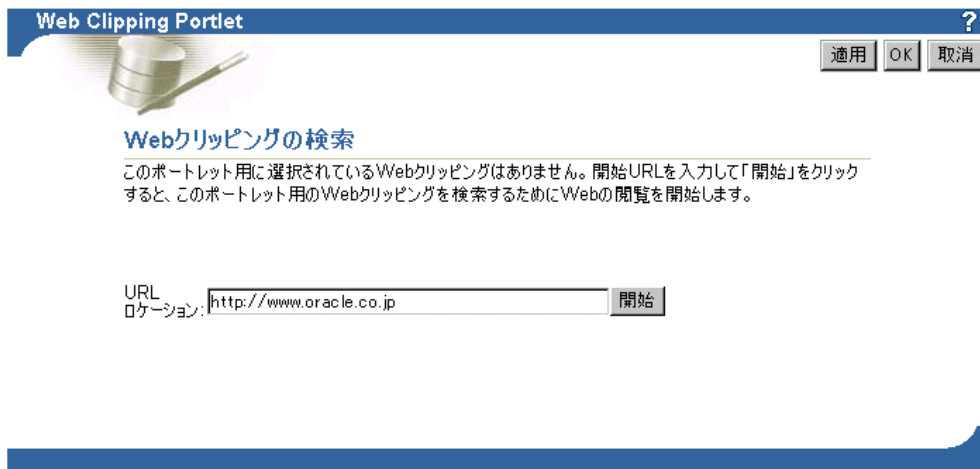
図 4-4 デフォルト設定の編集



「Web クリップングの検索」ページが表示されます。

2. 「URL ロケーション」フィールドに、クリップするコンテンツにリンクする Web ページの開始の位置を入力します。
3. 図 4-5 に示すように、「開始」をクリックします。

図 4-5 URL の指定



Web クリップング・スタジオに指定したページが表示されます。図 4-6 に、Web クリップング・スタジオに表示されたページを示します。

図 4-6 Web クリップング用のコンテンツを含むページのブラウザ



ブラウザ・バーの URL が変化します。変化前の URL は、次のとおりです。

`http://hostname:port/portal/page?_dad=portal&_schema=PORTAL...`

変化後の URL は、次のとおりです。

`http://hostname:port/portalTools/webClipping...`

- クリップするコンテンツを含むページをブラウズします。

この Web ページでハイパーリンクをクリックすると、使用するナビゲーション・リンクが記録されます。

注意： 最終的な Web クリップングに寄与しないブラウズ操作は、破棄されます。表示モードの間は、重要なブラウズ操作のみが後の再生用に記録され、破棄されたリンクは参照されません。

HTTP の Basic 認証または Digest 認証が必要な Web サイトの場合は、ユーザー名とパスワード情報を要求するフォームが表示されます。このエンコードされた認証情報は、ブラウズ情報の一部として記録されます。

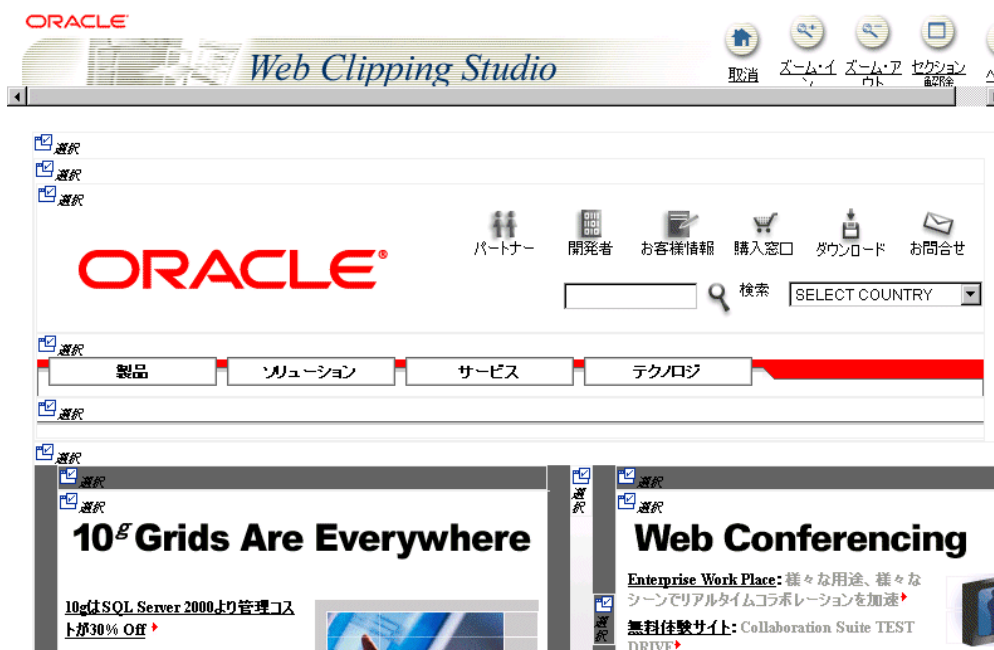
- クリップするコンテンツを含むページを表示した後は、Web クリップング・スタジオのバナーで、[図 4-7](#) に示すように、「セクション」をクリックします。

図 4-7 ターゲットの Web ページのセクション化



[図 4-8](#) に示すように、セクション化によって、ターゲットの Web ページがクリップ可能な複数のセクションに分割されます。「セクション」をクリックした後は、表示されたページのリンクがブラウズできなくなります。ナビゲーションを続行する場合は、Web クリップング・スタジオのバナーで「セクション解除」をクリックします。

図 4-8 セクション化されたターゲットの Web ページ



- クリップする Web コンテンツのセクションの左上にある「**選択**」をクリックします。クリッピングとして1度に選択できるセクションは、1つのみです。

注意： 使用可能なセクションの数を増やすには、Web クリッピング・スタジオのバナーで「**ズーム・イン**」をクリックします。たとえば、ネストした表の1レベルをドリルダウンするには、「**ズーム・イン**」をクリックします。使用可能なセクションの数を減らすには、「**ズーム・アウト**」をクリックします。

- 選択したセクションのプレビューが表示されます。目的のセクションが表示された場合は、Web クリッピング・スタジオのバナーで「**選択**」をクリックして、Web クリッピング・ポートレットの「デフォルトの編集」ページに戻ります。

ポートレットにクリップされたセクションを使用しない場合は、「**選択解除**」をクリックして、そのセクションを含むページに戻ります。そのページで別のセクションを選択するか、「**セクション解除**」をクリックして、別のページにナビゲートできます。

8. 「Web クリップングの検索」ページで「OK」をクリックすると、使用するページの Web クリップング・ポートレットに、選択された Web クリップングが表示されます。(このページで、デフォルトのプロパティを編集できます。詳細は、4-10 ページの「Web クリップング・ポートレットのプロパティの設定」を参照してください。)

図 4-9 に、Web クリップング・ポートレットに追加されたコンテンツを示します。

図 4-9 ポータル・ページの Web クリップング・ポートレットに追加されたクリップされたコンテンツ



Web クリッピング・ポートレットのプロパティの設定

様々なポートレット設定を編集することによって、Web クリッピング・ポートレットの外観を変更し、エンド・ユーザーとポートレットとの対話方法を指定できます。

Web クリッピング・ポートレットのプロパティを設定する手順は、次のとおりです。

1. Web クリッピング・ポートレットの上にある「**デフォルトの編集**」アイコンをクリックします。
2. リンク・ターゲットをポートレット内に表示する場合は、**URL リライティング**・リストから「**インライン**」を選択し、リンク・ターゲットを新規ウィンドウに表示する場合は、「**なし**」を選択します。
3. 「**タイトル**」フィールドに、ポートレット・バナーに表示するタイトルを入力します。
4. 「**説明**」フィールドに、ポートレットの説明を入力します。
5. 「**タイムアウト (秒)**」フィールドに、Web クリッピング・プロバイダが、コンテンツをクリップした Web ページへの接続を試みる時間 (秒) を入力します。
6. 「**有効期限 (分)**」フィールドに、キャッシュされたコンテンツの有効期間 (分) を入力します。指定した時間内に発生したポートレット・コンテンツへのリクエストは、キャッシュによって処理されます。

キャッシュの期限が切れた後、ポートレット・コンテンツへのリクエストは、ポートレットの Web クリッピング・データソースからコンテンツを取得して処理されます。キャッシュも、このコンテンツでリフレッシュされます。

7. Web クリッピング・ポートレット用のコンテンツをクリップしている間に、フォームに情報を入力した場合は、「**入力のパラメータ化**」セクションを使用できます。「**クリックして、パラメータ化を開始します。**」チェック・ボックスを選択し、Web クリッピング・ポートレットのコンテンツに関連付けられたパラメータをカスタマイズします。その後、次の手順に従います。
 - a. 「**パラメータ**」リストから、カスタマイズするパラメータを選択します。
 - b. ポートレットをカスタマイズするときに、パラメータに独自の値を指定することをエンド・ユーザーに対して許可する場合は、「**カスタマイズ可能**」リストから、そのパラメータを選択します。許可しない場合は、「**なし**」を選択します。
 - c. 「**表示名**」フィールドに、パラメータに表示する名前を入力します。
 - d. 「**デフォルト値**」フィールドに、パラメータにデフォルトで使用する値を入力します。
8. 「**OK**」をクリックします。

ポートレットへの外部アプリケーションの統合

Oracle Application Server Single Sign-On を利用し、認証されている外部 Web サイトからコンテンツをクリップして、外部アプリケーションを Web クリップング・ポートレットに統合できます。たとえば、特定のページにアクセスするためにログインが必要な「finance.yahoo.com」（外部アプリケーション）にアカウントがある場合は、そのページのクリップを Web クリップング・ポートレットに組み入れることができます。

外部アプリケーションを統合するには、次の手順に従います。

1. 次のように認証情報を指定して、OracleAS Portal に外部アプリケーションを設定します。
 - a. 「外部アプリケーションの管理」ポートレットにナビゲートします。（「**管理**」タブを選択してから「**ポートレット**」タブを選択します。「**管理ポートレット**」を選択します。「**SSO/OID**」を選択します。次に、「**SSO Server 管理**」を選択します。「**外部アプリケーションの管理**」を選択します。）
 - b. 「外部アプリケーションの追加」をクリックします。
 - c. 「アプリケーション名」に、アプリケーションの名前を入力します。たとえば、ユーザー名を「**Yahoo**」にします。
 - d. 「ログイン URL」に、<http://login.yahoo.com> など、アプリケーションにログインするための URL を入力します。
 - e. 「ユーザー名 /ID フィールド名」に、アプリケーションで使用するユーザー名を入力します。
 - f. 「パスワード」に、アプリケーションで使用するパスワードを入力します。OracleAS Portal では、この情報がアプリケーションへの接続に使用されます。
 - g. 認証方式として「**POST**」を選択します。

OracleAS Server Single Sign-On および外部アプリケーションの詳細は、『Oracle Application Server Single Sign-On 管理者ガイド』を参照してください。

2. 次のように、Web クリップング・ポートレットに外部アプリケーションを登録します。
 - a. 「**管理**」タブを選択してから「**ポートレット**」タブを選択します。
 - b. 「**プロバイダの登録**」を選択します。
 - c. 「プロバイダの登録」ページで、「名前」に **webClippingYahoo** と入力し、「表示名」に **Web Clipping Provider Yahoo** と入力します。「**実装スタイル**」に対して「**Web**」を選択します。
 - d. 「次へ」をクリックします。
 - e. 「接続の定義」ページの「**一般プロパティ**」セクションで、「**URL**」に、次のように入力します。

<http://server:port/portalTools/webClipping/providers/webClipping>

- f. ユーザーの識別については、「ユーザーの識別情報を、Web プロバイダのアプリケーション内の異なる名前にマップする必要があります。また、Web プロバイダは、ブラウザ・セッションを確立するために、外部アプリケーション・ログインを必要とします。このオプションを選択する場合は、下の外部アプリケーション ID を指定してください。」を選択します。
- g. 「外部アプリケーション ID」では、「値リスト」アイコンをクリックし、Yahoo に登録した外部アプリケーションを選択します。

図 4-10 に、「接続の定義」ページを示します。

図 4-10 Web クリップング・プロバイダに対する外部アプリケーションの指定

接続の定義
?

< 前へ
次へ >
完了
取消

ステップ 2 / 3 >>>

一般プロパティ

WebプロバイダのURLを指定して、その通信設定を構成します。

URL

Portalと同じCookieドメインのWebプロバイダ

該当する場合には、サービスIDを入力します。これで、複数のプロバイダに同じURLを使用してアクセスできるようにします。指定されたURLでアクセス可能な特定のプロバイダは、サービスIDで識別されます。

サービスID

(例: "urn:ADAPTER_PROVIDER"または"urn:webProvider")

Webプロバイダとの通信時に、Portalがユーザーの識別情報を設定する方法を指定します。

- ユーザーは、Webプロバイダ・アプリケーション内とシングル・サインオン認証内とで、同じ識別情報を保持します。
- ユーザーの識別情報を、Webプロバイダのアプリケーション内の異なる名前にマップする必要があります。また、Webプロバイダは、ブラウザ・セッションを確立するために、外部アプリケーション・ログインを必要とします。このオプションを選択する場合は、下の外部アプリケーションIDを指定してください。

外部アプリケーションID

- h. プロキシ設定が正しいことを確認します。「完了」をクリックします。
- 3. 4-3 ページの「ページへの Web クリップング・ポータルへの追加」で説明したように、ポータルをページに追加します。
- 4. 4-5 ページの「Web クリップング・ポータルに表示する Web ページのセクションの選択」で説明したように、Web クリップング・ポータルに表示する「finance.yahoo.com」セクションの 1 セクションを選択します。

これで、認証が必要なページからのクリップの場合でも、ポータルで使用できます。

1つのプロバイダに関連付けることができる外部アプリケーションは1つのみです。外部アプリケーションごとに新規のプロバイダを登録する必要があります。

例：ページへの Portal Center 情報の追加

この例では、Web クリップング・ポートレットを使用して、Portal Center の情報をポータル・ページに追加します。

この例の内容は、次のとおりです。

- 演習：ページへの Web クリップング・ポートレットの追加
- 演習：Portal Center でのクリップングの選択
- 演習：Web クリップング・ポートレットのプロパティの設定

演習：ページへの Web クリップング・ポートレットの追加

管理者は、すべてのユーザーに対して個人用ページを設定できます。ここでは、管理者がその機能を使用できると想定しています。この演習では、Web クリップング・ポートレットを個人用ページに追加します。

1. 「ページ・グループ」ポートレットの「ページの編集」セクションで、「ページをブラウズ」アイコンをクリックします。

デフォルトでは、「ページ・グループ」ポートレットは、「Portal ビルダー」ページの「構築」タブにあります。

2. 「ページ・グループ・マップ」で「個人用ページ」ノードを拡張してから、ユーザー名の最初の文字についてノードを拡張します。図 4-11 に、「個人用ページ」ノードを示します。

図 4-11 「ページ・グループ・マップ」のノードの拡張



3. ユーザー名の横にある「オブジェクトを戻す」をクリックします。個人用ページが表示されます。

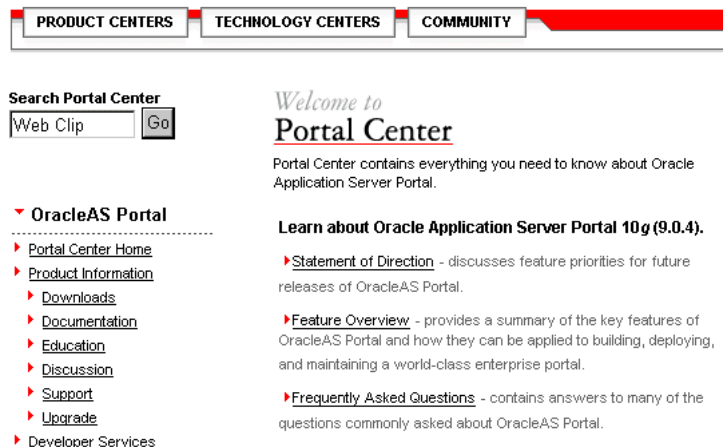
4. 「ページの編集」をクリックします。
5. 任意のポートレット領域で、「ポートレットの追加」アイコンをクリックします。
6. 「ポートレットの追加」 ページで、「Web クリップング・ポートレット」リンクをクリックします。
デフォルトでは、Web クリップング・ポートレットは、ポートレット・リポジトリの「Portal ツール」ページにあります。このページが見つからない場合は、**検索**フィールドを使用してポートレットを検索します。
7. Web クリップング・ポートレットが「選択したポートレット」リストに追加されます。「OK」をクリックします。

演習 : Portal Center でのクリッピングの選択

この演習では、Portal Center にナビゲートして特定の情報を検索し、その結果をポートレットに対するクリッピングとして選択します。

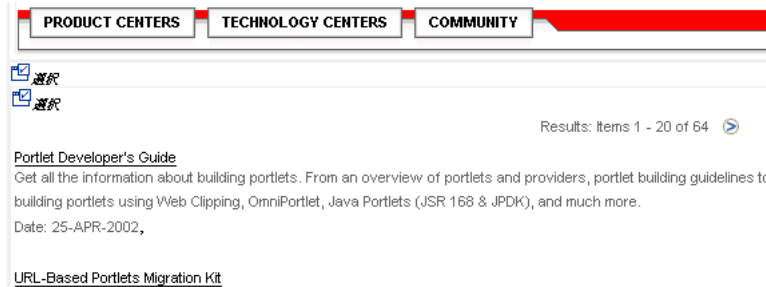
1. Web クリップング・ポートレットの上にある「デフォルトの編集」アイコンをクリックします。
2. Web クリップング・スタジオの「Web クリップングの検索」ページにある「URL ロケーション」フィールドに、`http://portalcenter.oracle.com` と入力します。「開始」をクリックします。
3. 図 4-12 に示すように、Portal Center のホーム・ページの左にある**検索**フィールドに検索文字列を入力します。この演習では、Web Clip と入力して「Go」をクリックします。

図 4-12 Portal Center のホーム・ページ上での情報の検索



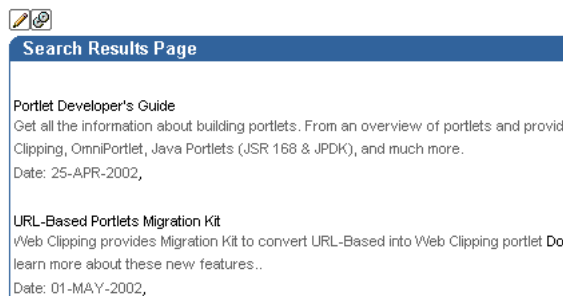
4. 検索結果が Web クリップング・スタジオに表示されます。「セクション」をクリックします。図 4-13 に示すように、Web クリップング・スタジオによって、ターゲットの Web ページがクリップ可能な複数のセクションに分割されます。

図 4-13 ターゲットの Web ページのセクション化



5. 検索結果の左上隅にある「選択」をクリックします。
検索結果のセクションのプレビューが表示されます。
6. 「選択」をクリックして、検索結果のセクションが、クリップするセクションであることを確認します。
7. 「OK」をクリックして、ページ上の Web クリップング・ポートレットに、選択した Web クリップングを表示します。図 4-14 に、ページ上に表示された Web クリップングを示します。

図 4-14 Web クリップング・ポートレットに表示された選択された Web クリップング

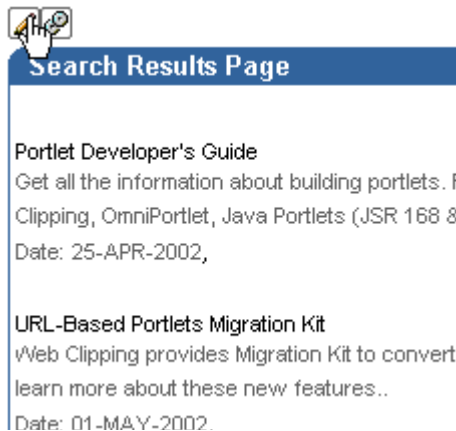


演習 : Web クリップング・ポートレットのプロパティの設定

この演習では、Web クリップング・ポートレットのプロパティを編集して、エンド・ユーザーがポートレットに様々な製品を表示できるようにします。

1. 図 4-15 に示すように、追加した Web クリップング・ポートレットの上にある「**デフォルトの編集**」アイコンをクリックします。

図 4-15 Web クリップング・ポートレットの「デフォルトの編集」のクリック



2. 表示されたページで、「**プロパティ**」セクションにある次の項目を変更します。
 - 「**タイトル**」フィールドに、**Portal Center Search** と入力します。このタイトルは、Web クリップング・ポートレットのバナー、およびユーザーが Web クリップングのパラメータをカスタマイズできるページに表示されます。

図 4-16 プロパティの設定

Webクリッピングの検索

Webクリッピングは、このポートレット・インスタンス用にすでに選択されています。「変更」をクリックすることによって、別の Webクリッピングを選択できます。

URL
ロケーション:

プロパティ

Webクリッピングのプロパティを設定できます。

このWebクリッピングは、ビューから変更できます。:

このWebクリッピングは、ビューから変更できます。:	<input type="button" value="true"/>
タイトル:	<input type="text" value="Portal Center Se"/>
説明:	<input type="text" value="New Desc"/>
タイムアウト(秒):	<input type="text" value="60"/> 範囲内の値を選択してください[1, 60]
有効期限(分):	<input type="text" value="30"/>

入力のパラメータ化

Webクリッピングは、パラメータ化できるように設定できます。チェックボックスをクリックして、パラメータ化を可能にする、Webクリッピング・スタジオで訪問した特定URLの特定パラメータの選択を開始します。これによって、Webクリッピングの表示は、ページ・ビューでカスタマイズできるようになります。また、これらのパラメータにはデフォルト値を入力できます。

クリックして、パラメータ化を開始します。:

3. ポートレットに表示されるコンテンツは、Portal Center の検索フィールドに情報を入力することによって生成されるため、検索で使用するパラメータをカスタマイズして、エンド・ユーザーによる独自の検索文字列の指定を許可できます。

「入力のパラメータ化」セクションで、「クリックして、パラメータ化を開始します。」チェック・ボックスを選択します。

4. パラメータ表の最後の行で、次の変更を行います。
 - 「パラメータ」列で、リストから「p_mainsearch」を選択します。
 - 「カスタマイズ可能」列で、リストから「PARAM1」を選択します。
 - 「表示名」列に、Portal Center Search と入力します。
 - 「デフォルト値」に Web Clip が表示されていることを確認し、適切なパラメータを選択していることを確認します。

図 4-17 に、「入力のパラメータ化」セクションを示します。

図 4-17 ユーザー入力に対するパラメータの指定

入力のパラメータ化

Webクリッピングは、パラメータ化できるように設定できます。チェックボックスをクリックして、パラメータ化を可能にする。Webクリッピング・スタジオで訪問した特定URLの特定パラメータの選択を開始します。これによって、Webクリッピングの表示は、ページ・ビューアでカスタマイズできるようになります。また、これらのパラメータにはデフォルト値を入力できます。

クリックして、パラメータ化を開始します。:

インデックス	URL	パラメータ	カスタマイズ可能	表示名	デフォルト値
0	http://portalcenter.	none	N/A		
1	http://portalcenter.	p_mainsearch	Param1	Portal	Web Clip

- 「OK」をクリックして、使用するページの Web クリッピング・ポートレットにデフォルトの検索結果を表示します。
- 図 4-18 に示すように、Web クリッピング・ポートレットのバナーで、「カスタマイズ」をクリックします。

図 4-18 Web クリッピング・ポートレットのバナーにある「カスタマイズ」のクリック



- 表示されたページで、「入力」セクションにスクロールします。検索文字列のパラメータ・フィールドに、このパラメータの表示名に指定した「Portal Center Search」が表示されていることに注意してください。「Portal Center Search」フィールドには、様々な検索文字列を入力します。たとえば、図 4-19 に示すように、「OmniPortlet 2003」と入力します。

図 4-19 パラメータに対する入力の指定

Webクリッピングの検索

Webクリッピングは、このポートレット・インスタンス用にすでに選択されています。「変更」をクリックすることによって、別のWebクリッピングを選択できます。

プロパティ

次のプロパティは、このWebクリッピングの作成者によって設定されました。

タイトル: Portal Center Search
 説明: New Desc
 タイムアウト(秒): 60
 有効期限(分): 30

入力

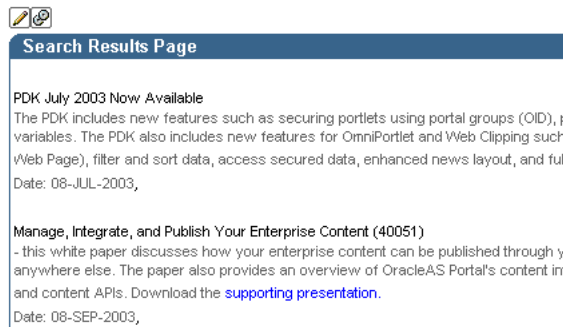
Webクリッピングはカスタマイズ可能になりました。これらのパラメータに初期値を入力できます。


Portal Center Search:

8. 「OK」をクリックします。

これで、Web クリッピング・ポートレットには、[図 4-20](#) に示すように、OmniPortlet 2003 の情報について Portal Center で実行した検索の結果が表示されます。

図 4-20 ユーザーが入力したパラメータに基づく新規 Web クリッピングの結果



 **Search Results Page**

PKD July 2003 Now Available
 The PDK includes new features such as securing portlets using portal groups (OID), c variables. The PDK also includes new features for OmniPortlet and Web Clipping such vWeb Page), filter and sort data, access secured data, enhanced news layout, and full
 Date: 08-JUL-2003,

Manage, Integrate, and Publish Your Enterprise Content (40051)
 - this white paper discusses how your enterprise content can be published through y anywhere else. The paper also provides an overview of OracleAS Portal's content int and content APIs. Download the [supporting presentation](#).
 Date: 08-SEP-2003,

Web クリップングに関する現在の制限

この項では、Web クリップングのリリース 9.0.4.1 に関する制限について説明します。リリースの最新の機能と制限の詳細は、リリース・ノートで確認してください。

- OracleAS Portal のエクスポート / インポート・ユーティリティを使用すると、コンテンツをポータル内の複数インストール間で移動できますが、これには、プロバイダ・カスタマイズのエクスポート / インポートは含まれません。具体的に言うと、Web クリップング・ポートレットの各インスタンスに定義されているクリッピングは、ポートレット・インスタンスとともにエクスポートまたはインポートされることはありません。その結果、ポートレット・インスタンスがインポートされたターゲットの OracleAS Portal のインスタンスには、定義されたクリッピングがありません。
- 接続しているサイトで、Cookie を操作するために多数の JavaScript が使用されている場合、または、書き出される HTML 文書を変更するために JavaScript メソッドの `document.write` が使用されている場合は、そのサイトからコンテンツをクリップできない場合があります。
- プロキシ認証。Web クリップングは、プロキシ認証のすべてをサポートしているわけではありません。サポートされているのはグローバル認証のみで、ユーザーごとの認証はサポートされていません。ファイアウォール外側の Web サイトへの接続に使用するプロキシ・サーバーがプロキシ認証用に構成されていて、ページがそのファイアウォールの外側にある場合は、手動でプロキシ認証を構成しないかぎり、Web クリップング・スタジオを使用してページをクリップすることはできません。詳細は、4-23 ページの「[プロキシ認証](#)」を参照してください。

Web クリップングのトラブルシューティング

この項では、Web クリップング・プロバイダまたは Web クリップング・スタジオの使用中に発生する可能性のある問題のトラブルシューティングに役立つ情報を提供します。

テスト・ページを使用したプロバイダの状態の確認

Web クリップング・プロバイダのテスト・ページを使用すると、プロバイダが適切に機能しているかどうかを判断できます。このテスト・ページにアクセスするには、次の場所にある Portal ツール・アプリケーションの初期画面で「Web クリップング・プロバイダ」をクリックします。

`http://Hostname:Port/portalTools`

「プロバイダ・テスト・ページ:Web クリップング」が表示されます。ここでは、次の情報が提供されます。

- ポートレット情報。Web クリップング・ポートレットに関する情報。(Web クリップング・プロバイダは、ポートレットを1つのみ含みます。)
- プロバイダ初期化パラメータおよび値。
- 構成を編集するページへのリンクを含めた、プロバイダの状態。

テスト・ページの使用方法的詳細は、『Oracle Application Server Portal 構成ガイド』の付録「Web クリップングの管理」を参照してください。

接続に関する問題の解決

クリップするサイトやクリップしたサイトへの接続の作成または保持で問題が発生した場合は、次の事項に注意します。

- プロキシ・サーバーで、ファイアウォール外側の HTTP サーバーへの接続が必要な場合は、そのプロキシ・サーバーが正しく構成されていることを確認します。詳細は、4-23 ページの「[プロキシ・サーバーの構成](#)」を参照してください。
- プロキシ・サーバーがプロキシ認証用に構成されている場合は、プロキシ認証を手動で構成しないかぎり、ファイアウォールの外側にあるページをクリップすると、HTTP エラー・コード 407 が表示されます。手動による構成の詳細は、4-23 ページの「[プロキシ認証](#)」を参照してください。
- リバース・プロキシを使用する場合は、リバース・プロキシ・サーバーが正しく構成されていることを確認します。リバース・プロキシの構成方法の詳細は、『Oracle Application Server Portal 構成ガイド』の「[拡張構成の実行](#)」の章を参照してください。

- Web クリップング・ポートレットにクリップを追加しているときに、接続の作成または保持に問題が発生した場合は、含まれているロード・バランサが正しく構成に設定されていないことが原因である可能性があります。
 - 複数の OC4J インスタンスが 1 つのロード・バランサの背後に設定されている場合は、Web クリップング・リポジトリと HTTP プロキシを、すべての OC4J インスタンスに対して同一になるように設定してから、OC4J インスタンスをロード・バランサに結合する必要があります。

Web クリップングには定義があり、この定義は、Oracle Database サーバーがホスティングする Web クリップング・リポジトリに永続的に格納される必要があります。複数の中間層の環境では、同一のリポジトリに定義を格納するように、すべての OC4J のインスタンスを構成する必要があります。

さらに、すべての OC4J のインスタンスの構成を、HTTP プロキシに対して同一にする必要があります。

- ロード・バランサは、セッション対応である必要があります。セッション対応でない場合、最初のリクエストは接続しますが、後続のリクエストは異なるインスタンスにルーティングされる可能性があります、失敗します。

ロード・バランサを使用した構成方法の詳細は、『Oracle Application Server Portal 構成ガイド』の「拡張構成の実行」の章を参照してください。

- URL が正しいことを確認するには、クリップの前に、クリップする URL をブラウザでテストします。また、URL をプロバイダの中間層からテストして、そこからのアクセスが可能であることを確認します。
- ページをクリップできない場合は、そのページが IFrame で過密になっていないことを確認します。ページ・ソースに注意しながら、ブラウザにページを表示します。ページに IFrame が含まれている場合は、その IFrame の src 属性が示す URL で開始します。
- クリップング内のイメージが取得されず、それ以外が取得される場合は、ブラウザのプロキシ設定を確認します。イメージはリンク (IMG タグの「src」属性を使用) として扱われるため、クリップされたサイトにあるイメージは、元のサイトから直接供給されます。クリップングの作成時や表示モード時にプロキシ設定を有効にすることが、イメージが必要な場合は、ブラウザのプロキシ設定を無効にすることで、クリップング内でのイメージの表示が禁止されます。ブラウザのプロキシ設定を有効にします。
- 接続できない場合は、エラー・ログを確認します (エラー・ログの詳細は、4-24 ページの「ロギング・レベルの設定」を参照)。拒否されているデータベースへのログオンに関するメッセージがログに含まれているかどうかを確認します。含まれている場合は、インフラストラクチャ・データベース内の PORTAL スキーマ・パスワードが、手動で変更され、Oracle Internet Directory に格納されているパスワードと一致していない可能性があります。パスワードの設定方法の詳細は、『Oracle Internet Directory 管理者ガイド』を参照してください。

プロキシ・サーバーの構成

プロキシ・サーバーで、ファイアウォール外側の HTTP サーバーへの接続が必要な場合は、そのプロキシ・サーバーが正しく構成されていることを確認します。

プロキシ・サーバーを構成するには、4-21 ページの「[テスト・ページを使用したプロバイダの状態の確認](#)」で説明したように、Web クリップング・プロバイダのテスト・ページに移動します。Web クリップング・プロバイダのテスト・ページで、「[HTTP プロキシ](#)」行の「[操作](#)」列にある「[編集](#)」をクリックします。「[プロバイダの編集](#)」ページで、「[HTTP プロキシのホスト名](#)」および HTTP プロキシの「[HTTP プロキシ・ポート](#)」を指定します。

ファイアウォール内側のサーバーにアクセスするには、「[この文字で始まるドメインにプロキシを使用しない](#)」を選択して URL を入力し、ファイアウォールを通過する必要がないドメイン名のリストを指定できます。新規の設定を有効にするために、OC4J を再起動する必要はありません。

プロキシ・サーバーの構成方法の詳細は、『[Oracle Application Server Portal 構成ガイド](#)』の付録「[Web クリップングの管理](#)」を参照してください。

プロキシ認証

Web クリップングは、プロキシ認証のすべてをサポートしているわけではありません。サポートされているのはグローバルなプロキシ認証で、ユーザーごとの認証はサポートされていません。プロキシ認証は、次の場所にある `provider.xml` ファイルを編集して、手動で構成する必要があります。

`OracleHome/j2ee/OC4J_Portal/applications/portalTools/webClipping/WEB-INF/providers/webClipping`

`provider.xml` ファイルの「`proxyInfo`」セクションで、次のように指定します。

- `proxyUser` に、ユーザー名を指定します。
- `proxyPassword` に、ユーザーのパスワードを指定します。
- `proxyType` に、**Basic** または **Digest** のいずれかの認証のタイプを指定します。
- `proxyRealm` に、レルム名を指定します。

次に、プロキシ認証の設定例を示します。

```
<proxyUser>root</proxyUser>
<proxyPassword>HfnkWjuXsWbB</proxyPassword>
<proxyType>Basic</proxyType>
<proxyRealm>webcl</proxyRealm>
```

ロギング・レベルの設定

デフォルトでは、Web クリップングのロギング・レベルはレベル 3 に設定されています。このレベルでは、構成、重大なエラーおよび警告に関する情報が提供されます。このレベルは、日常の操作にあわせたレベルです。デバッグに役立つ情報を表示するには、ロギング・レベルを 7 に設定します。

ロギング・レベルを設定するには、`web.xml` ファイルを編集して、`oracle.portal.log.LogLevel` パラメータにレベルを指定します。この `web.xml` ファイルは、次の場所にあります。

```
OracleHome/j2ee/OC4J_Portal/applications/portalTools/webClipping/WEB-INF
```

たとえば、デバッグ情報を表示するレベルに設定するには、次の例に示すように、パラメータ `oracle.portal.log.LogLevel` の値を 7 に設定します。

```
<context-param>
  <param-name>oracle.portal.log.LogLevel</param-name>
  <param-value>7</param-value>
</context-param>
```

テスト・ページへのアクセス時や Web クリップング・ポートレットの実行中に発生したエラーは、次のファイルのいずれかに書き込まれます。

```
OC4J_HOME/j2ee/home/application-deployments/portalTools/application.log
IAS_HOME/j2ee/OC4J_instance/application-deployments/portalTools/OC4J_instance_
default_island_1/application.log
```

Web クリップング・プロバイダの構成中に発生する可能性のあるエラーの詳細は、次の場所を参照してください。

```
http://host:port/portalTools/webClipping/htdocs/help/configuring.webclipping.html
```

Java ポートレットの構築

この章では、Java ポートレット仕様に基づいて Java ポートレットを構築する方法、Oracle Application Server Portal Developer Kit-Java (PDK-Java) を使用して Java ポートレットを作成する方法、および Struts アプリケーションからポートレットを作成する方法を説明します。

- Java ポートレット作成のガイドライン
- Java ポートレット仕様および WSRP の概要
 - WSRP と JPS の関連
- Oracle JDeveloper を使用した JPS 準拠のポートレットの構築
 - Oracle JDeveloper ポータル・アドインのインストール
 - JPS 準拠のポートレットの構築
- Oracle JDeveloper を使用した PDK-Java ポートレットの構築
 - Oracle JDeveloper ポータル・アドインのインストール
 - PDK-Java ポートレットの構築
 - レンダリング・モードの追加
 - ポートレットのカスタマイズ
 - パラメータの引渡しとイベントの送信
 - セッション情報へのアクセス
 - ポートレットのセキュリティの実装
 - キャッシュを使用したポートレットのパフォーマンスの向上
 - 多言語ポートレットの作成
- Oracle JDeveloper を使用した Struts ポートレットの作成
 - OracleAS Portal および Apache Struts のフレームワーク
 - Struts ポートレットの作成

Java ポートレット作成のガイドライン

Java ポートレット仕様 (JPS) または PDK-Java のいずれかに基づいて Java でポートレットを作成する場合は、この項のガイドラインに従ってください。このガイドラインでは、特に、ポートレットの表示モードを説明します。表示モードは、ポートレットの機能の一部です。

注意： JPS および OracleAS Portal は、それぞれ独自のモードを提供しています。ポートレットを JPS にコーディングする場合は、OracleAS Portal が提供する追加モードにマップするカスタム・ポートレット・モードを宣言できます。

OracleAS Portal ポートレットの表示モードは次のとおりで、表示方法と動作がそれぞれ異なります。JPS ポートレットの場合は、`portlet.xml` にカスタム・ポートレット・モードを定義できます。カスタム・モードの定義は、特に、ポートレットを他のベンダーのポータル実装と相互運用する場合に役立ちます。

- 共有画面モード (JPS の場合は表示モード)
- 編集モード (JPS および OracleAS Portal)
- デフォルト編集モード (JPS および OracleAS Portal)
- プレビュー・モード (JPS および OracleAS Portal)
- 全画面モード (OracleAS Portal)
- ヘルプ・モード (JPS および OracleAS Portal)
- 情報モード (JPS および OracleAS Portal)
- リンク・モード (OracleAS Portal)

共有画面モード（JPS の場合は表示モード）

ポートレットは共有画面モード（JPS では表示モードと呼ばれます）を使用して、他のポートレットと同じページに表示されます。ポートレットといえば、ほとんどの人がこのモードを思い浮かべます。共有画面モードでは、ポートレットは HTML の表のセル内にレンダリングされます。つまり、ポートレットには、HTML、プラグイン、Java アプレットなどの他のテクノロジーも含めて、表のセル内にレンダリング可能なコンテンツを表示できます。表のセルの実際のサイズは、ユーザー設定、ブラウザの幅、およびポートレットに表示されるコンテンツの量とスタイルによって異なります。ポートレットを開発する場合は、そのポートレットが他のポートレットとページを共有すること、およびポートレットのサイズや配置を完全には制御できないことに注意してください。

ポートレットのレンダリングに関する HTML ガイドライン

プレーン HTML は最も基本的なポートレットのレンダリング方法であり、ポートレットの開発者に大きな柔軟性を提供します。リンク、フォーム、イメージ、表など、HTML の表のセル内に表示できるほとんどの標準的な HTML パラダイムを使用できます。HTML が適切に作成されていない場合は、異なる複数のブラウザで表示したときに表示の一貫性がなくなり、ページの一部が表示されないこともあります。次のルールに従ってください。

- **標準 HTML を使用する。** HTML の正式な仕様は、W3C にあります（詳細は <http://www.w3.org/MarkUp/> を参照）。
- **閉じていないタグや不要なタグがないように注意する。** ページはブラウザの動作の影響を受けるため、タグが正しく閉じていないとページの動作は予測不可能になります。weblint (<http://www.weblint.org/>) や HTML Tidy (<http://www.w3.org/People/Raggett/tidy/>) などのツールは、閉じていないタグや不要なタグの検出に役立ちます。
- **HTML の表のセルに正しくレンダリングできる要素のみを使用する。** 一部の構造体は、表のセルに正しく表示されない理由で使用できません。たとえば、フレームは表に挿入されると表示されません。
- **ポートレットのコンテンツは簡潔にする。** 全画面のコンテンツを取得して、小さなポートレットから公開しないでください。ポートレットのコンテンツが小さくなりすぎて、小さなモニターではポートレットが使用不能になります。
- **ポートレット内に固定幅の HTML の表を作成しない。** ユーザーのページに表示されるポートレットの列幅を事前に知る方法はありません。与えられた空間よりも広い空間がポートレットに必要な場合、特定のブラウザでは、他のポートレットとオーバーラップする可能性があります。
- **折り曲がらない長いテキスト行を避ける。** 固定幅の表を使用した場合と同様の結果になります。特定のブラウザでは、他のポートレットとオーバーラップする可能性があります。
- **ページ・サイズ変更時の動作をチェックする。** ブラウザ・ウィンドウをサイズ変更した際のポートレットの動作をテストし、ポートレットがブラウザ・ウィンドウの様々なサイズで動作することを確認してください。

- **ブラウザのデフォルトのフォントが変更された場合の動作をチェックする。** ブラウザのフォント・サイズはいつでも変更できることが必要です。ポートレットは、その変更に対応する必要があります。

また、使用する HTML は、サイトのパフォーマンスに直接影響を与えます。ユーザーは、リクエストしたページの表示にかかる時間に基づいてパフォーマンスを判断します。ブラウザには、HTML を解釈して表示するための時間が必要です。このため、次のガイドラインに従ってください。

- **深くネストした表を使用しない。** 深くネストした表は、古いバージョンのブラウザのパフォーマンスを大きく低下させます。OracleAS Portal は、複数のレベルの表を描画してポートレットをレンダリングします。ポートレットが表の中で表を使用している場合、そのページのレンダリングには、かなり長い待機時間が必要な場合があります。
- **長く複雑な HTML は避ける。** ポートレットはページを他のポートレットと共有します。したがって、ポートレットの生成時間はページのパフォーマンス全体に大きく影響します。ポートレットが複雑な HTML をレンダリングしたり、サード・パーティのアプリケーションなど外部リソースを待機する場合は、ページのレンダリングが大幅に遅くなることがあります。

ポートレットのレンダリングに関する Cascading Style Sheet ガイドライン

ページ上のそれぞれのポートレットのフォントと色は、ユーザーが選択したスタイル設定に一致する必要があります。このために、これらのスタイル選択は、各 OracleAS Portal ページの Cascading Style Sheet (CSS) を使用して自動的に埋め込まれます。その後、ポートレットは直接または API を使用して、フォントと色の設定にアクセスします。

ブラウザによって CSS 仕様の実装レベルが異なりますが、OracleAS Portal は、この仕様の非常に基本的なサブセットを使用して、整合性のあるフォントと色を実現します。CSS の実装レベルによってブラウザ間でのページの一貫性が影響されることはありません。CSS 使用に関するガイドラインは、次のとおりです。

- **ハードコーディングではなく CSS を使用する。** フォントと色のハードコーディングは非常に危険です。フォントと色をハードコーディングすると、ユーザーがページ・スタイルの設定を変更した場合に、ポートレットが不釣り合いに見えることがあります。開発者はユーザーが設定するフォントや色を予測できないため、ユーザーが選択した背景色と同じフォント色でハードコーディングする可能性があります。この場合、ユーザーはポートレットを識別できなくなります。
- **テキストの書式設定には CSS API を使用する。** スタイルシートの定義は OracleAS Portal ページの最上部にあります。直接コールしないでください。かわりに、テキストの適切な書式設定のために用意されている API を使用してください。これによって、後でスタイルシートが変更された場合でもポートレットは正しく動作します。
- **CSS を使用した絶対位置の設定は避ける。** ユーザーは各自のポータル・ページをカスタマイズできるため、ポートレットが特定の位置に表示される保証はありません。

編集モード (JPS および OracleAS Portal)

ポートレットは編集モードを備えており、これによってユーザーはポートレットの動作をカスタマイズできます。編集モードには設定のリストがあり、ユーザーはその設定を変更できます。カスタマイズ可能な設定には、タイトル、コンテンツの種類、書式、情報量、フォーム要素のデフォルト、およびポートレットの外観や内容に影響するあらゆる要素が含まれます。

通常、ポータル・ユーザーは、ポートレット・バナーの「**カスタマイズ**」をクリックしてポートレットの編集モードにアクセスします。「**カスタマイズ**」をクリックすると、同じブラウザ・ウィンドウに新しいページが表示されます。このポートレットは、通常、ポートレットの設定を選択するダイアログ・ボックスを表す Web ページを作成します。設定を適用すると、自動的に元のページに戻ります。

編集モード・オプションに関するガイドライン

次のガイドラインに従って、編集モードでユーザーに公開する内容を管理してください。

- **ユーザーがポートレットのタイトルをカスタマイズできるようにする。** 同じポートレットが同じポータル・ページに複数回追加されることがあります。ユーザーがタイトルをカスタマイズできると混乱を避けることができます。
- **キャッシュを使用する場合はコンテンツを失効化する。** カスタマイズによってポートレットの表示やコンテンツに変更が生じた場合は、ポートレットのコンテンツが再生成され、キャッシュから戻されないことを確認する必要があります。確認しないと、誤ったコンテンツが表示されることがあります。
- **編集モードを管理ツールとして使用しない。** 編集モードは、ユーザーがポートレットの動作を変更するために使用します。プロバイダ設定の変更などの管理作業が必要な場合は、その作業専用の安全なポートレットを作成してください。

編集モードのボタンに関するガイドライン

整合性を維持し、ユーザーにとっても便利なように、次のボタンを次の順序で編集モードに実装してください。

- **OK:** ユーザーのカスタマイズを保存し、コール元のポータル・ページにブラウザをリダイレクトします。
- **適用:** ユーザーのカスタマイズを保存し、現在のページを再ロードします。
- **取消:** 変更内容を保存せずに、コール元のポータル・ページにブラウザをリダイレクトします。

カスタマイズ値のレンダリングに関するガイドライン

カスタマイズ設定を変更するフォームを表示する場合は、デフォルトの値を表示し、ユーザーが設定を毎回再入力しないで済むようにしてください。動作の整合性を維持するために、カスタマイズ値のレンダリングは特定の順序で行います。カスタマイズ値のレンダリング順序は、次のとおりです。

1. **ユーザー設定項目**：このユーザーのカスタマイズが使用可能な場合は、問い合せて表示します。
2. **インスタンス・デフォルト**：ユーザーのカスタマイズがない場合は、ポートレット・インスタンスのシステム・デフォルトを問い合せて表示します。これらのデフォルトは編集モードで設定され、このポートレット・インスタンスに対してのみ適用されます。
3. **ポートレット・デフォルト**：システム・デフォルトのカスタマイズがない場合は、一般的なポートレットのデフォルトを表示しますが、空白の場合があります。一般的なポートレットのデフォルトはポートレットにハードコーディングされている場合がありますが、前述の2つの条件のいずれかが該当する場合は上書きしてください。

このロジックによって、ポータル内の他のポートレットとの整合性がとれた予測可能な方法でカスタマイズを表すことができます。PDK-Java では、このようなロジックを簡単に実装できます。

デフォルト編集モード（JPS および OracleAS Portal）

ポートレットはデフォルト編集モードを備えており、ページ設計者はこのモードを使用して、特定のポートレット・インスタンスのデフォルト動作をカスタマイズできます。デフォルト編集モードには設定のリストがあり、ページ設計者はその設定を変更できます。カスタマイズ可能な設定には、タイトル、コンテンツの種類、書式、情報量、フォーム要素のデフォルト、およびポートレットの外観や内容に影響するあらゆる要素が含まれます。

これらのデフォルトのカスタマイズ設定によって、個々のポートレットの外観とコンテンツをすべてのユーザーに対して変更できます。デフォルト編集モードは、ポートレットの表示内容と表示形式に対してシステム・レベルのデフォルトを定義するためのモードであるため、このモードを管理ツールとして使用したり、他のポートレットの管理に使用することは避けてください。

ページ設計者は、「ページのカスタマイズ」から**他のユーザー用にカスタマイズ**を選択してデフォルト編集モードにアクセスします。このリンクには、そのポートレットのワイヤー・フレーム・ダイアグラムのバナーに「**デフォルトの編集**」というラベルが付いています。

「**デフォルトの編集**」をクリックすると、同じブラウザ・ウィンドウに新しいページが表示されます。このポートレットは、通常、ポートレット・インスタンスの設定をカスタマイズするダイアログ・ボックスを表す **Web** ページを作成します。設定が適用されると、自動的に元のページに戻ります。

デフォルト編集モード・オプションに関するガイドライン

次のガイドラインに従って、デフォルト編集モードでページ設計者に公開する内容を管理してください。

- **キャッシュを使用する場合はキャッシュを失効化する。**カスタマイズによってポートレットの表示やコンテンツに変更が生じた場合は、ポートレットのコンテンツが再生成され、キャッシュから戻されないことを確認する必要があります。確認しないと、誤ったコンテンツが表示されることがあります。
- **デフォルト編集モードを管理ツールとして使用しない。**デフォルト編集モードは、ユーザーがポートレットの動作を変更するために使用します。プロバイダ設定の変更などの管理作業が必要な場合は、その作業専用の安全なポートレットを作成してください。

デフォルト編集モードのボタンに関するガイドライン

整合性を維持し、ユーザーにとっても便利のように、次のボタンを次の順序で編集モードに実装してください。

- **OK:** ユーザーのカスタマイズを保存し、コール元のポータル・ページにブラウザをリダイレクトします。
- **適用:** ユーザーのカスタマイズを保存し、現在のページを再ロードします。
- **取消:** 変更内容を保存せずに、コール元のポータル・ページにブラウザをリダイレクトします。

カスタマイズ値のレンダリングに関するガイドライン

カスタマイズ設定を変更するフォームを表示する場合は、デフォルトの値を表示し、ページ設計者が設定を毎回再入力しないで済むようにしてください。動作の整合性を維持するために、カスタマイズ値のレンダリングは特定の順序で行います。カスタマイズ値のレンダリング順序は、次のとおりです。

1. **インスタンス設定:** ポートレット・インスタンスのシステム・デフォルトを問い合わせ表示します。
2. **ポートレット・デフォルト:** システム・デフォルトのカスタマイズがない場合は、一般的なポートレットのデフォルトを表示しますが、空白の場合があります。一般的なポートレットのデフォルトはポートレットにハードコーディングされている場合がありますが、システム・デフォルトで上書きしてください。

このロジックによって、ポータル内の他のポートレットとの整合性がとれた予測可能な方法でカスタマイズを表すことができます。

プレビュー・モード (JPS および OracleAS Portal)

ポートレットはプレビュー・モードを備えており、ポートレットをページに追加する前に、それがどのように表示されるかをユーザーに示します。プレビュー・モードは、ポートレットの機能を視覚的に表現したものです。

通常、ポータル・ユーザーは、「ポートレットの追加」ページで「プレビュー」アイコンをクリックして、ポートレットのプレビュー・モードにアクセスします。選択したポートレットのプレビューがウィンドウに表示されます。表示後に、ユーザーは、そのポートレットをページに追加できます。ポータル管理者は、ポートレット・リポジトリからプレビュー・モードにアクセスできます。

このモードでポートレットがレンダリングされる場合、ポータルはポータル・バナーを描画しないことに注意してください。

プレビュー・モードに関するガイドライン

次のガイドラインに従って、プレビュー・モードでユーザーに公開する内容を管理してください。

- **ポートレットの機能に関する情報を提供する。**プレビュー・モードでは、ユーザーがポートレットの実際のコンテンツや機能に関する情報を得られるように、十分なコンテンツを生成する必要があります。
- **ポートレットのプレビューは大きくしない。**このモードで生成されるデータの量は、数行以内の HTML または 1 つの画面ショットに止めてください。プレビュー・モードは小さい領域に表示されるため、ウィンドウのサイズより大きくなると見栄えが悪くなり、ユーザーによるスクロールが必要になります。
- **ライブ・ハイパーリンクを使用しない。**「ポートレットの追加」ページまたはポートレット・リポジトリにレンダリングしたリンクは、予想どおりの動作をしないことがあります。ハイパーリンクは、下線付きのフォントでシミュレートできます。
- **アクティブ・フォーム・ボタンを使用しない。**「ポートレットの追加」ページまたはポートレット・リポジトリにレンダリングしたフォームは、予想どおりの動作をしないことがあります。フォーム要素をレンダリングする場合、そのフォーム要素はどこにもリンクしないでください。

全画面モード (OracleAS Portal)

ポートレットは、全画面モードを使用して、他のポートレットとページを共有している場合よりも詳細な表示を行います。全画面モードによって、ポートレットは画面全体を使用できます。

たとえば、ポートレットで経費情報を表示する場合、共有画面モードでは経費の多い上位 10 人のサマリーを表示し、全画面モードでは各人の経費合計を表示できます。また、ポートレットは、Web アプリケーションへのショートカットも提供できます。ポートレットの共有画面モードに経費領収書発行へのインタフェースがある場合は、全画面モードから経費アプリケーション全体にリンクすることも可能です。

ポータル・ユーザーは、ポートレットのタイトルをクリックして、ポートレットの全画面モードにアクセスします。

厳密には、JPS ポートレットに全画面モードはありません。ただし、表示モード（共有画面モード）とウィンドウの最大化を使用して、JPS ポートレット用に全画面モードと同等の機能を実装できます。

全画面モードに関するガイドライン

次のガイドラインに従って、全画面モードでユーザーに公開する内容を管理してください。

- **HTML の表のセルに正しくレンダリングできる要素のみを使用する。** 全画面モードはウィンドウ全体を使用できますが、ポートレットは HTML の表のセル内で書式設定されます。したがって、表のセルに正しく表示されない構造体は使用しないでください。たとえば、フレームは表に挿入されると表示されません。
- **前ページにナビゲートする手段を提供する。** ユーザーが元のポータル・ページに戻るためのリンクまたはボタンを提供する必要があります。

ヘルプ・モード (JPS および OracleAS Portal)

ポートレットは、ヘルプ・モードを使用して、そのポートレットの機能と使用方法に関する情報を表示します。ユーザーはこのモードで、ポートレットおよびそのコンテンツと機能に関する有用な情報を検索できます。

Portal のユーザーは、ポートレット・バナーの「ヘルプ」をクリックしてポートレットのヘルプ・モードにアクセスします。

ヘルプ・モードに関するガイドライン

次のガイドラインに従って、ヘルプ・モードでユーザーに公開する内容を管理してください。

- **ポートレットの使用方法を説明する。** ポートレットのインタフェースのみでは、ポートレットのすべての機能がユーザーに理解されない可能性があります。ポートレットの機能とその活用方法を説明してください。

情報モード (JPS および OracleAS Portal)

ユーザーは、実行中のポートレットのバージョン、発行と著作権の情報、および作者への連絡方法を参照できる必要があります。登録が必要なポートレットは、このモードから Web ベースのアプリケーションまたは連絡先情報にリンクできます。

ポータル・ユーザーは、ポートレット・バナーの「**情報**」をクリックしてポートレットの情報モードにアクセスします。同じブラウザ・ウィンドウに新しいページが表示されます。この新しいページにコンテンツを生成するか、またはユーザーを既存のページやアプリケーションに移動させることができます。

情報モードに関するガイドライン

次のガイドラインに従って、情報モードでユーザーに公開する内容を管理してください。

- **関連の著作権、バージョンおよび作成者の情報を表示する。**ユーザーは、使用しているポートレットの内容と詳細情報の入手先を参照できる必要があります。この情報ページは、ポートレットのサポート時に重要になることがあります。

リンク・モード (OracleAS Portal)

ポートレットはリンク・モードを使用して、そのポートレットへのリンクをモバイル・ページにレンダリングします。ユーザーがこのリンクをクリックすると、ポートレットが表示モードでコールされます（ただし、コンテンツ・タイプは異なります）。

Oracle Mobile XML コンテンツ・タイプのサポートを宣言する JPS ポートレットの場合、OracleAS Portal は次のいずれかの方法でリンクをレンダリングします。

- ポートレットがサポートを宣言している場合は、ウィンドウを最小化した状態でポートレットの表示モードをコールします。
- それ以外の場合は、ポートレットのタイトルを使用してリンクをレンダリングします。

リンク・モードに関するガイドライン

次のガイドラインに従って、リンク・モードでユーザーに公開する内容を管理してください。

- **コンテンツを制限する。**ユーザーには共有画面モードと同じデータを表示する必要がありますが、モバイル・デバイスの小さい画面での表示が困難または不可能なグラフィック・イメージや不要なコンテンツはすべて除きます。

Java ポートレット仕様および WSRP の概要

企業ポータル・プロジェクトに従事する組織では、アプリケーションの統合が主要な課題でした。これまでユーザーは、単一のポータル・プラットフォーム専用の API を使用してポートレットを開発してきましたが、多くの場合、特定のポータル・ベンダーによるポートレットでは不十分でした。このような状況は、次の標準を導入することで大きく変わります。

- Web Services for Remote Portlets (WSRP)
- JSR 168 に基づく Java ポートレット仕様 (JPS)¹

この 2 つの標準によって、異なる複数のポータル製品を相互運用できるポートレットを開発できるため、組織におけるポートレットの可用性が拡大します。この可用性の拡大によって、企業ポータルの構築時には、組織の生産性を大幅に向上させることができます。

WSRP は Web サービスの標準です。この標準によって、ポータルまたは他の中間 Web アプリケーションを使用する、ユーザー対象で視覚的な Web サービスのプラグ・アンド・プレイが可能になります。1 つの標準である WSRP によって、標準対応のコンテナと WSRP ポータル間の相互運用が実現します。WSRP では、次のものを定義します。

- WSRP サービスを起動するための Web Services Definition Language (WSDL) インタフェース。
- WSRP サービスで発生するマークアップのためのマークアップ・フラグメント・ルール。
- WSRP サービスとメタデータを公開、検索およびバインドするためのメソッド。

JPS は、ポートレットとポータル間の相互運用を可能にし、集計、パーソナライズ、表示形式およびセキュリティの各領域に対処する一連の API を定義する仕様です。JPS では、次の機能を提供するコンテナ・サービスを定義します。

- ポートレット機能をコーディングするためのポートレット API
- ポートレット・コンテナ内でユーザー相互作用を作成するための URL リライティング・メカニズム
- ポートレットのセキュリティとパーソナライズ

Oracle は、WSRP 委員会に積極的に参加しており、JPS の専門家グループのメンバーでもあります。Oracle では、これらの標準に対するサポートをコミットし、WSRP 対応のポータルを本稼働リリースする予定です。現在、Oracle Technology Network (OTN) のメンバーは、次のことができます。

- ホスティングされているプレリリース版の WSRP ポータルを表示できます。ホスティングされているこのポータルで、ユーザーは一連のサンプル WSRP ポータルを表示し、プロバイダ (プロデューサーとも呼ばれます) を登録してポートレットをページに追加することもできます。

¹ Java ポートレット仕様 1.0 は、Java Specification Request 168 および JSR168 専門家グループによって作成されました。

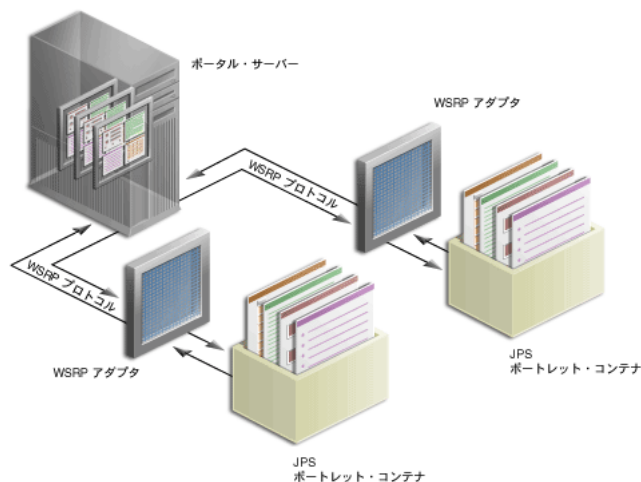
- 開発者用のプレビュー版の WSRP ポータルを表示できます。
- Oracle が提供する JPS 準拠のポートレット・コンテナのプレビュー・リリースを表示できます。このリリースでは、Java ポートレットが WSRP サービスとして公開されます。
- WSRP 対応のポートレットの相互運用性を検証できます。
- 前述の標準に基づいて相互運用可能なポートレットを構築するためのツールをダウンロードできます。

WSRP と JPS の関連

WSRP はポータル・サーバーとポートレット・コンテナ間の通信プロトコルです。一方、JPS にはポートレットを構築するための Java ポートレット API が記述されています。これらの標準を組み合わせることで、開発者は、内部または外部のソースからのアプリケーションをポートレットとして WSRP ポータルに統合できます。これによって、OracleAS Portal リポジトリからポートレットを選択するだけで簡単にポータル・ページを構築できるようになります。図 5-1 に、WSRP 仕様のアーキテクチャを示します。

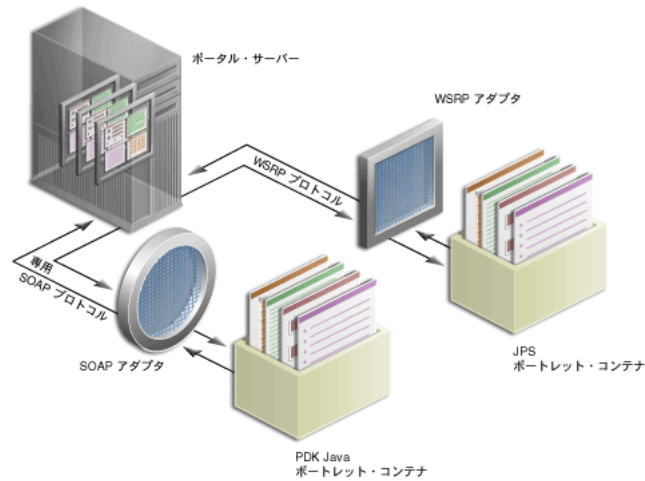
注意： 図 5-1 は、JPS ポートレットを WSRP とともに使用方法を示していますが、WSRP は JPS 以外のポートレットでも使用できることに注意してください。

図 5-1 WSRP 仕様のアーキテクチャ



OracleAS Portal の既存のアーキテクチャは WSRP 委員会で指定されたアーキテクチャと似ているため、OracleAS Portal では、Oracle のポータルと既存の API (PDK-Java) 間の通信、および新しい Java ポートレット API との間の通信もサポートできます。図 5-2 に、WSRP ポータルのアーキテクチャを示します。JPS 準拠のポートレット・コンテナは WSRP プロトコルを使用して通信を行い、PDK-Java ポートレット・コンテナは Oracle の専用プロトコルを使用して通信を行うことに注意してください。

図 5-2 OracleAS Portal の WSRP アーキテクチャ



Oracle JDeveloper を使用した JPS 準拠のポートレットの構築

ポートレットを構築する際は、OTN から拡張機能をダウンロードして Oracle JDeveloper に追加できます。この拡張機能を使用するには、次のステップを実行します。

- [Oracle JDeveloper ポータル・アドインのインストール](#)
- [JPS 準拠のポートレットの構築](#)

この項では、次のことを前提にしています。

- ポートレットの表示モードなど、ポートレットに関する用語の知識があること。[第 1 章「ポートレットの概要」](#) および 5-2 ページの「[Java ポートレット作成のガイドライン](#)」を参照してください。
- Java ポートレット・コンテナをダウンロードおよびインストール済で、ポートレットをデプロイする Oracle Application Server Containers for J2EE 9.0.4 のコンテナが存在すること。
- Oracle JDeveloper に関する知識があり、それを使用して Java コンポーネントを構築およびデプロイする方法を習得していること。Oracle JDeveloper は OTN-J からダウンロードできます。

Oracle JDeveloper ポータル・アドインのインストール

OracleAS Portal Developer Kit (PDK) には、Oracle JDeveloper 用のアドインをインストールするために必要なライブラリが用意されています。このライブラリを使用すると、ポートレットを開発する際の柔軟性や生産性が大幅に向上します。この拡張機能には、JPS 準拠のポートレットを構築するウィザードと、PDK-Java ポートレットを構築するウィザードが含まれています。いずれのウィザードも、ポートレット・スケルトンの作成手順をガイドするため、開発者に必要なのは、独自のビジネス・ロジックを実装するのみです。

アドインを取得する手順は、次のとおりです。

1. <http://portalstudio.oracle.com> にアクセスします。
2. このページの左側にある「**Integration/Utilities**」をクリックします。
3. 「**Portal Add-In for Oracle JDeveloper**」をクリックして `portal-addin.zip` をダウンロードします。
4. 「**Install Instructions**」をクリックして、その指示に従ってください。

注意： <http://portalstudio.oracle.com> にアクセスして `portal-addin.zip` を検索することもできます。

JPS 準拠のポートレットの構築

Java のポートレット・ウィザードを正常にインストールした後は、Oracle JDeveloper を使用して、次のように相互運用可能なポートレットを迅速かつ簡単に開発できます。

- **ポートレットの作成**: ウィザードを使用して、基本的なポートレット・コードとフレームワークに必要な構成ファイルを作成します。
- **ポートレット・ロジックの追加**: 独自のビジネス・ロジックを使用してサンプル・コードを拡張します。
- **アプリケーション・サーバーへのポートレットのデプロイ**: Oracle JDeveloper を使用して、作成したアプリケーションをアプリケーション・サーバーにデプロイします。
- **ポートレットの登録と表示**: ポートレットをローカルの OracleAS Portal インスタンスに登録して表示します。ローカル・インスタンスがない場合は、<http://portalstandards.oracle.com> にアクセスしてください。

ポートレットの作成

この項では、ポートレット・ウィザードの手順を説明します。最初に、実装するポートレットの表示モード、および実装メソッド (JSP、HTTP サーブレット、Java クラスまたは HTML) を選択できます。次に、ウィザードによって、選択したモードごとに簡単なサンプル実装が作成されます。

1. Oracle JDeveloper を開き、このプロジェクトを作成するワークスペースをクリックします。ワークスペースがない場合は、次の手順に従ってワークスペースを作成できます。
 - a. 「アプリケーション-ナビゲータ」で「**アプリケーション**」ノードを右クリックし、コンテキスト・メニューから「**新規**」を選択します。
 - b. 「新規ギャラリー」ダイアログ・ボックスの「項目」リストから「**ワークスペース**」を選択します。
 - c. 「OK」をクリックします。
 - d. 「**ワークスペース名**」および「**ディレクトリ名**」を入力し、ワークスペース作成ダイアログ・ボックスの「**新規の空のプロジェクトを追加**」の選択を解除します。
 - e. 「OK」をクリックします。
2. 「アプリケーション-ナビゲータ」でワークスペースの名前を右クリックし、コンテキスト・メニューから「**新規プロジェクト**」を選択します。
3. 「新規ギャラリー」ダイアログ・ボックスの「項目」リストから「**空のプロジェクト**」をクリックします。
4. 「OK」をクリックします。
5. ワークスペース作成ダイアログ・ボックスで「**プロジェクト名**」および「**ディレクトリ名**」を入力します。

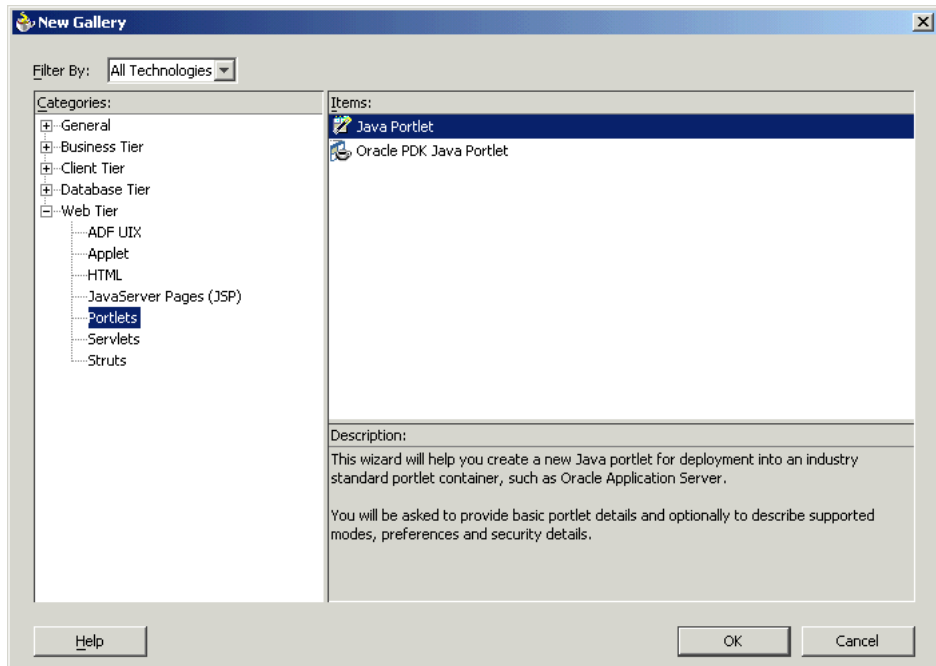
6. 「OK」をクリックします。
7. プロジェクトを右クリックし、コンテキスト・メニューから「新規」を選択します。
8. 「カテゴリ」リストで「Web Tier」カテゴリを開き、「Portlet」をクリックします。

注意：「Portlet」が表示されていない場合は、5-14 ページの「Oracle JDeveloper ポータル・アドインのインストール」を参照して、アドインが正しくインストールされていることを確認してください。

9. 「項目」リストで「Java Portlet」をクリックします。

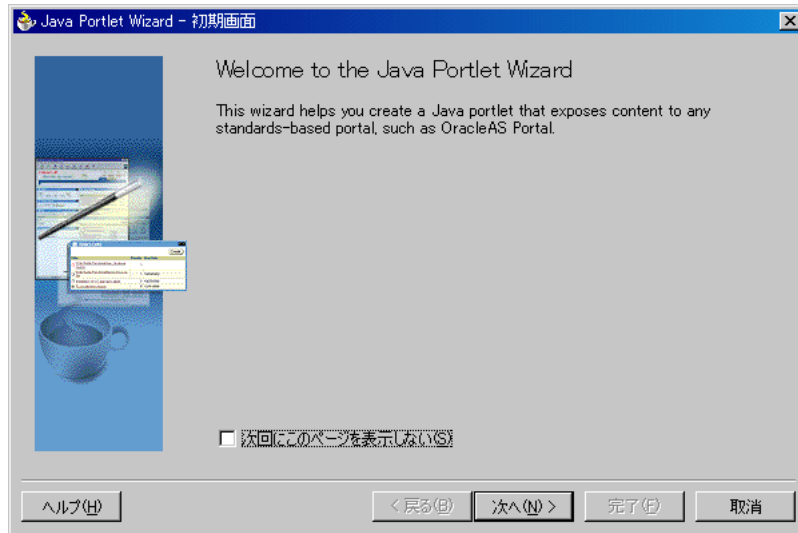
注意：「Java Portlet」をクリックすると、JPS 準拠のポートレットを作成するためのポートレット・ウィザードが開きます。「Oracle PDK Java Portlet」をクリックすると、PDK-Java ポートレットを作成するためのポートレット・ウィザードが開きます。

図 5-3 新規ダイアログ・ボックス



10. 「OK」をクリックします。ポートレット・ウィザードが表示されます。
11. ウィザードの「初期画面」ページが表示された後、「次へ」をクリックします。

図 5-4 「初期画面」ページ

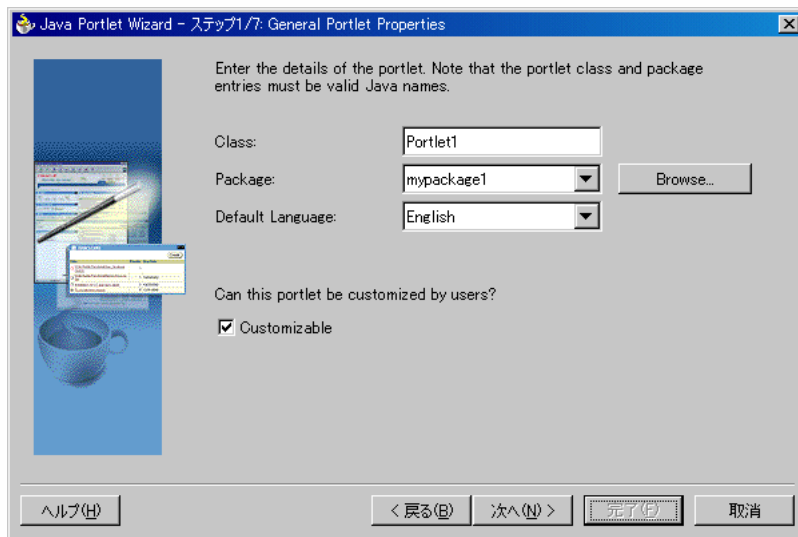


12. 「General Portlet Properties」ページで、表 5-1 および図 5-5 に示す値を入力します。

表 5-1 「General Portlet Properties」の値

プロパティ	値
Class	ウィザードで作成するクラスの名前を入力します。このフィールドにはデフォルトのクラス名が表示されます。そのまま使用するか、または変更できます。
Package	プロジェクトのパッケージを参照し、クラスが常駐するパッケージを選択します。パッケージを選択しない場合、ウィザードではプロジェクトのデフォルト・パッケージが使用されます。
Default Language	ポートレットがサポートするデフォルト言語を選択します。ウィザードでは、デフォルトで英語が使用されます。
Customizable	エンド・ユーザーによるポートレットのカスタマイズが可能かどうかを選択します。ウィザードでは、デフォルトでカスタマイズ可能です。

図 5-5 「General Portlet Properties」 ページ



13. 「次へ」をクリックします。

注意：「完了」ボタンは、ウィザードのステップ 2 が完了するまで使用できません。

14. 「Name and Attribution」 ページで、表 5-2 および図 5-6 に示す値を入力します。

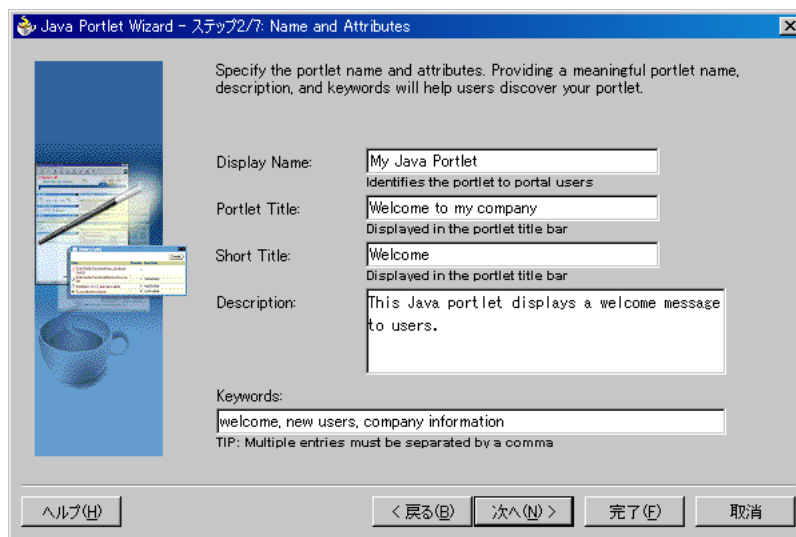
表 5-2 「Name and Attribution」 の値

プロパティ	値
Display Name (必須)	OracleAS Portal のカタログまたはリポジトリに表示する名前を入力します。この例では、My Java Portlet と入力します。
Portlet Title (必須)	ポータル・ページのポータルレット・ヘッダーに表示するタイトルを入力します。この例では、Welcome to my company と入力します。
Short Title	モバイル・デバイス用のポータルレット・ヘッダーに表示するタイトルを入力します。この例では、Welcome と入力します。

表 5-2 「Name and Attribution」の値 (続き)

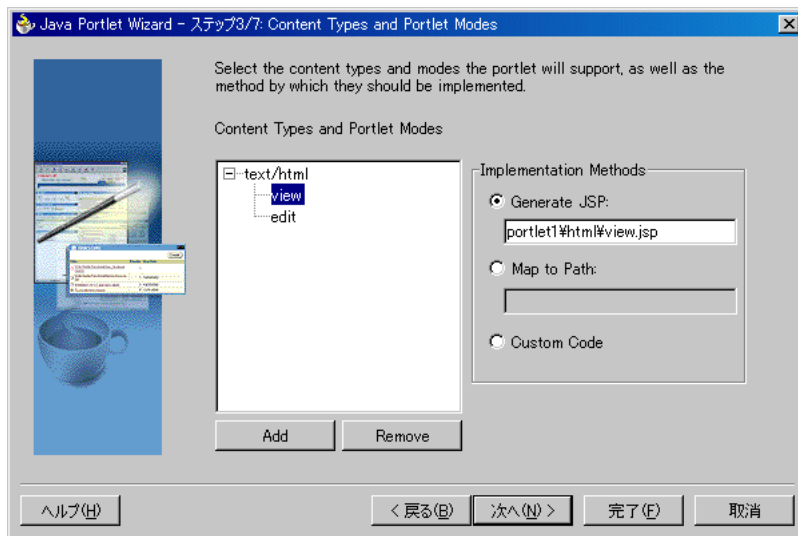
プロパティ	値
Description	ポートレットの説明を入力します。この例では、This Java portlet displays a welcome message to users. と入力します。
Keywords	ユーザーがポータル内でポートレットを検索するのに役立つキーワードを入力します。この例では、welcome, new users, company information と入力します。

図 5-6 「Name and Attribution」 ページ



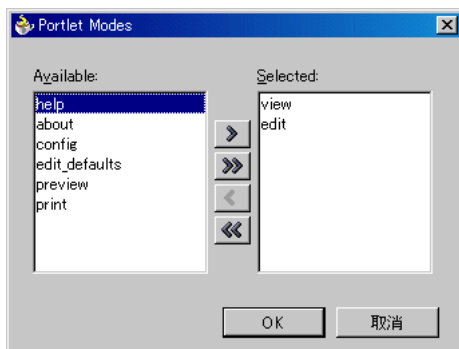
15. 「次へ」(ポートレットのプロパティ指定を続行する場合) または「完了」をクリックします。ここで「完了」をクリックすると、ウィザードでは、残りのすべての設定についてデフォルト値が選択されます。
16. 図 5-7 に示す「Content Types and Portlet Modes」ページで、ポートレットのコンテンツ・タイプを選択し、ポートレット・モードを実装にマップします。デフォルトでは、コンテンツ・タイプは text/html で、ポートレット・モードは「View」でポートレットが表示されます。「General Portlet Properties」ページで「Customizable」を選択した場合は、text/html のポートレット・モードとして「Edit」も表示されます。

図 5-7 「Content Types and Portlet Modes」 ページ



- a. text/html 以外のコンテンツ・タイプを追加する必要がある場合は、text/html をクリックし、「Add」をクリックします。この例では、他のコンテンツ・タイプを追加する必要はありません。
- b. 他のポータルレット・モードを追加する必要がある場合は、既存のポータルレット・モード（たとえば、「View」）をクリックし、「Add」をクリックします。図 5-8 に示すように、使用可能なポータルレット・モードのリストが表示されます。ポータルレット・モードを追加する場合は、任意のポータルレット・モードを使用可能モード・リストから選択済モード・リストに移動します。この例では、他のポータルレット・モードを選択する必要はありません。「Portlet Modes」ダイアログ・ボックスで作業を終了した後は、「OK」をクリックします。

図 5-8 「Portlet Modes」 ダイアログ・ボックス

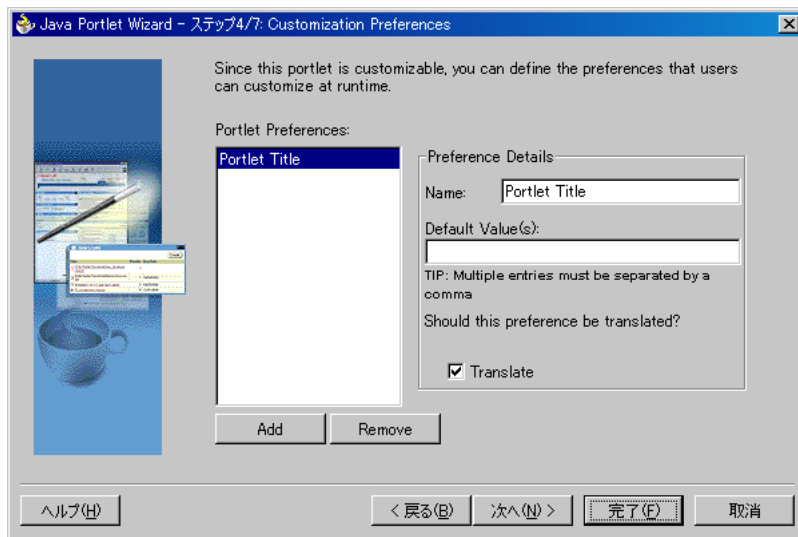


- c. 必要なポートレット・モードをすべて追加した後は、各モードについて実行する機能を選択する必要があります。ポートレット・モードごとに、ポートレット・モードをクリックし、右側のラジオ・グループから任意の機能を選択します。この例では、「Edit」と「View」の両方のポートレット・モードに対して「**Generate JSP**」を選択し、生成された JSP のデフォルト・パスとファイル名をそのまま使用します。ポートレット・モードの詳細は、5-2 ページの「[Java ポートレット作成のガイドライン](#)」を参照してください。

注意：「Generate JSP」または「Custom Code」を選択すると、指定した場所にコードが生成されます。「Map to Path」を選択すると、個別に作成するリクエストは既存の Web リソースにルーティングされます。

17. 「次へ」をクリックします。
18. ウィザードの前のステップで、「General Portlet Properties」ページの「**Customizable**」を選択した場合は、図 5-9 に示すように「Customization Preferences」ページが表示され、ポートレットの設定を宣言できます。ウィザードの前のステップで「**Customizable**」を選択しなかった場合、このページは表示されません。「Customization Preferences」ページで、設定名、デフォルト値および設定値を翻訳するかどうかを指定します。

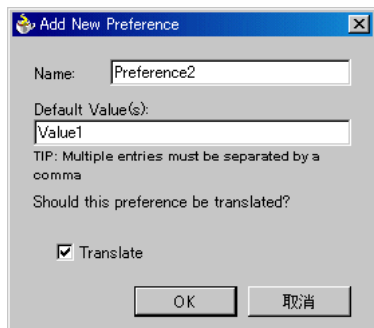
図 5-9 「Customization Preferences」 ページ



- a. 設定を追加するには、「Add」をクリックし、図 5-10 に示すように「Add New Preference」ダイアログ・ボックスで名前、デフォルト値、および翻訳するかどうかを指定します。

注意： 名前は常に翻訳されますが、デフォルト値は必ずしも翻訳する必要はありません。たとえば、値が整数の場合、翻訳は不要です。

図 5-10 「Add New Preference」ダイアログ・ボックス



- b. 設定を削除する場合は、削除する設定を「Portlet Preferences」リストから選択し、「Delete」をクリックします。

19. 「次へ」をクリックします。
20. 図 5-11 に示す「Security Roles」ページで、ポートレットにセキュリティ・ロールを追加できます。ウィザードには、事前定義されたセキュリティ・ロールはありません。ウィザードは、セキュリティ・ロールについて `web.xml` を解析し、ポートレットで参照できるようにします。ここでは新規プロジェクトにセキュリティ・ロールを定義しないため、このステップでの作業はありません。セキュリティ・ロールは、後で JPS に従って手動で作成できます。

図 5-11 「Security Roles」 ページ



21. 「次へ」をクリックします。
22. 「Caching」ページで、ポートレットのキャッシュをデフォルトで有効にするかどうかを指定します。ポートレット自体で、指定のレスポンスのコンテンツをキャッシュすることもできます。このページで指定する設定は、レスポンスに対するキャッシュ条件がポートレットで指定されていない場合に有効になります。この例では、図 5-12 に示すように、次のように指定してください。
 - a. 「Cache Portlet」をクリックします。
 - b. 「Cache Content Expires After」をクリックします。
 - c. キャッシュされたコピーのデフォルトの有効期限（60 秒）をそのまま使用します。

注意： ポートレットでデフォルトでキャッシュしない場合は、「**Do Not Cache By Default**」を選択します。この場合、ウィザードではキャッシュ有効期限が 0 秒に設定されます。前述したとおり、このキャッシュ設定が有効となるのは、レスポンスに対するキャッシュ条件がポートレットで指定されていない場合のみです。

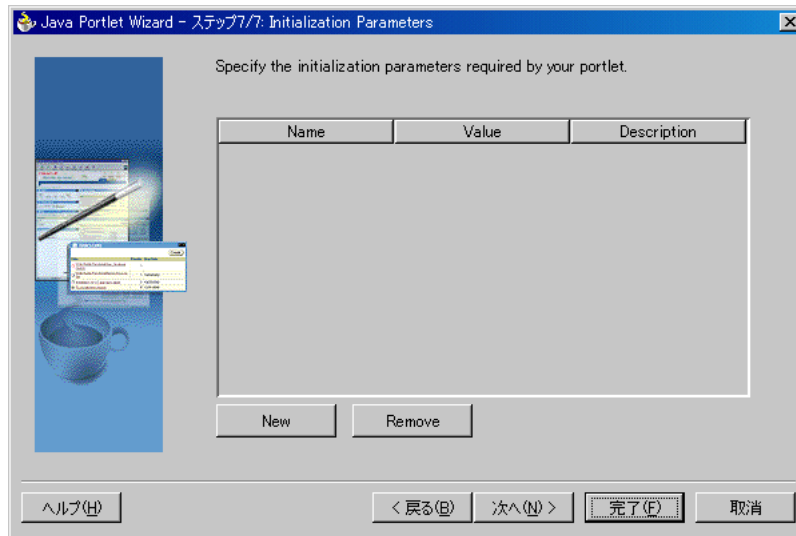
ここでキャッシュしないように設定し、後で、ポートレットでデフォルトでキャッシュする必要がある場合は、簡単に変更できます。変更するには、ウィザードで生成した `portlet.xml` ファイルのキャッシュ有効期限の値を、0 より大きい数字に指定します。

図 5-12 「Caching」 ページ



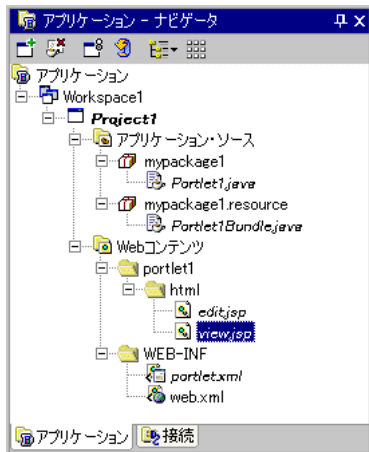
23. 「次へ」をクリックします。
24. 図 5-13 に示す「Initialization Parameters」ページでは、ポートレットに必要な初期化パラメータを追加できます。初期化パラメータによって、`.war` ファイルの内容を決定する Web アプリケーション開発者が指定されます。このファイルは、JNDI 変数のかわりに、Web アプリケーション（サーブレットやポートレット）のすべてのコンポーネントの動作を互換性のある方法で構成するために使用します。初期化パラメータの詳細は、Java ポートレット仕様を参照してください。この例では、初期化パラメータは必要ありません。

図 5-13 「Initialization Parameters」 ページ



25. 「次へ」をクリックします。
26. 「完了」をクリックすると、ポートレット用のファイルが生成されます。この例では、「アプリケーション-ナビゲータ」のプロジェクト・ノードに対して次のファイルが生成されます (図 5-14 を参照)。
 - 「Content Types and Portlet Modes」 ページで「**Generate JSP**」を選択した場合は、各表示モードのコードが生成されます。「**Custom Code**」を選択した場合、そのコードはポートレットの Java クラスに常駐することになります。
 - 2つの Java クラス
 - portlet.xml
 - web.xml

図 5-14 「アプリケーション - ナビゲータ」



ポートレット・ロジックの追加

デフォルト実装を作成した後は、独自のビジネス・ロジックを追加してサンプル・コードを拡張し、必要な機能を実装できます。機能の追加方法は、JavaDoc または JPS を参照してください。この例では、この手順を実行する必要はありません。次のデプロイ手順に進んでください。

アプリケーション・サーバーへのポートレットのデプロイ

ウィザードを終了してポートレットを生成した後は、そのポートレットをアプリケーション・サーバーにデプロイできます。JPS 準拠のポートレットを作成した場合は、そのポートレットを、ウィザードを使用してベンダーの JPS 準拠のコンテナにデプロイできます。次に、JPS 準拠のポートレットを、Oracle Application Server Containers for J2EE で実行されている Oracle の WSRP コンテナにデプロイする手順を説明します。

- [Oracle Application Server Containers for J2EE への接続の確立](#)
- [WAR ファイルのデプロイ](#)

Oracle Application Server Containers for J2EE への接続の確立 アプリケーション・サーバーへの接続を確立する手順は、次のとおりです。

注意： 次の各手順では、Oracle Application Server Containers for J2EE のスタンドアロン・インスタンスにデプロイする手順を説明します。完全な Oracle Application Server インスタンスへのデプロイ方法は、Oracle JDeveloper のオンライン・ヘルプ・システムを参照してください。

1. プロジェクトが開いていない場合は、Oracle JDeveloper を起動して前項で作成したプロジェクトを開きます。
2. ナビゲータで「**接続**」タブをクリックします。
3. 「**接続**」ノードを右クリックし、コンテキスト・メニューから「**アプリケーション・サーバー接続の作成**」を選択します。次の手順に従って表示されているウィザードを終了します。
 - a. 「初期画面」ページが表示されている場合は、「**次へ**」をクリックします。今後、「初期画面」ページを表示しない場合は、「**次回にこのページを表示しない**」を選択してください。
 - b. わかりやすい接続名（たとえば、PDKStandardsOC4J）を入力し、接続タイプとして「**スタンドアロンOC4J**」を選択します。
 - c. 「**次へ**」をクリックします。
 - d. 管理者のユーザー名とパスワードを入力します。このパスワードは、PDK 標準の Oracle Application Server Containers for J2EE のインストール時に設定されています。
 - e. 「**次へ**」をクリックします。
 - f. [表 5-3](#) に示す情報を入力します。

表 5-3 アプリケーション・サーバー接続の作成時の設定

設定	値
URL	この設定の完全な RMI URL を入力します。次に例を示します。 <code>ormi://my.machine.com:23791/</code> RMI ポート番号は、次のファイルを参照してください。 <code>OC4J_HOME/j2ee/home/config/rmi.xml</code>
宛先の Web サイト	デプロイした J2EE アプリケーション・ファイルが含まれる宛先の Web サイトの名前を入力します。この例では、デフォルト値の <code>http-web-site</code> をそのまま使用します。
OC4J の <code>admin.jar</code> がインストールされているローカル・ディレクトリ	前述の URL 設定で指定したリモート・サーバーとバージョンの互換性がある、ローカルの <code>admin.jar</code> へのパスを入力します。Oracle JDeveloper 9.0.3 を使用して構築した JPS 準拠のポートレットの場合は、Oracle Application Server Containers for J2EE リリース 9.0.4 のインスタンスを指すように、この設定のデフォルト・パスを変更する必要があります。次に例を示します。 <code>OC4J_HOME¥j2ee¥home</code>

- g. 「次へ」をクリックします。
- h. 「接続のテスト」をクリックして、接続詳細を確認します。接続が正常な場合は、成功メッセージが表示されます。テストに失敗した場合は、接続情報の修正が必要になる場合があります。
- i. 「完了」をクリックします。

WAR ファイルのデプロイ WAR ファイルを作成してデプロイする手順は、次のとおりです。

1. 「アプリケーション-ナビゲータ」に移動します。
2. 現在のプロジェクトで、web.xml を右クリックし、コンテキスト・メニューから「WAR デプロイメント・プロファイルの作成」を選択します。
3. 「デプロイメント・プロファイルの保存」ダイアログ・ボックスで、名前をわかりやすい名前（たとえば、jsrportlet1.deploy）に変更します。
4. 「OK」をクリックします。
5. プロファイル設定ダイアログ・ボックスで、次の手順を実行します。
 - a. 「J2EE Web コンテキスト・ルートを指定」をクリックし、隣接するフィールドに my-portlet と入力します。
 - b. 「OK」をクリックします。
6. デプロイメント・プロファイル（たとえば、jsrportlet1.deploy）を右クリックし、コンテキスト・メニューから「配布先」→アプリケーション・サーバー接続（たとえば、PDKStandardsOC4J）を選択します。
7. Oracle JDeveloper の下部にある**デプロイ・ログ**に「デプロイが完了」メッセージが表示されるのを待ち、エラーが発生していないことを確認します。
8. ログに表示された URL（たとえば、<http://myserver.com:8888/my-portlet>²）に /portlets?WSDL を追加して、JPS 準拠のポータルレットを OracleAS Portal に登録する際に使用する URL を作成します。次に例を示します。

<http://myserver.com:8888/my-portlet/portlets?WSDL>

注意： ログに、Oracle JDeveloper でリモート・サーバーの HTTP ポート番号を判別できないというメッセージが表示される場合があります。通常は、Oracle Application Server Containers for J2EE のホーム・ページにアクセスする URL（たとえば、<http://myserver.com:8888>）を調べるとポート番号を判別できます。

² この章の例では、通常、デフォルト・ポートのポート 8888 を使用します。使用するシステムに応じて、別のポート番号を使用することもできます。

ポートレットの登録と表示

プロバイダとそのポートレットを作成してデプロイした後は、そのプロバイダを OracleAS Portal に登録する必要があります。プロバイダを登録すると、そのプロバイダを検索して通信するために必要な情報が OracleAS Portal に提供されます。プロバイダを登録した後は、プロバイダとそのポートレットがポートレット・リポジトリで使用可能になります。また、OracleAS Portal のナビゲータにもプロバイダとそのポートレットのリストが表示されます。

注意： ポートレット・ビルダーなどの組み込みツールを使用してポートレットやプロバイダを作成すると、OracleAS Portal によってそのプロバイダが自動的に登録されます。さらに、作成したポートレットは、ポートレット・リポジトリに自動的に表示されます。OracleAS Portal には組み込みポートレットも用意されており、それらは事前構成済みのプロバイダに含まれています。たとえば、複雑な設定が不要なポートレットの OmniPortlet と Web Clipping は、すでに OracleAS Portal に登録されています。これらのポートレットは「ポートレットの追加」リストで参照できます。ただし、ポートレットとプロバイダをプログラムで作成した場合、そのプロバイダは、ポータル・ユーザーが使用できるように登録する必要があります。

この項では、プロバイダを登録し、ポートレットをページに追加する方法を説明します。

- ローカルの OracleAS Portal インスタンスへの登録
- portal.standards.com への登録
- ポートレットの追加

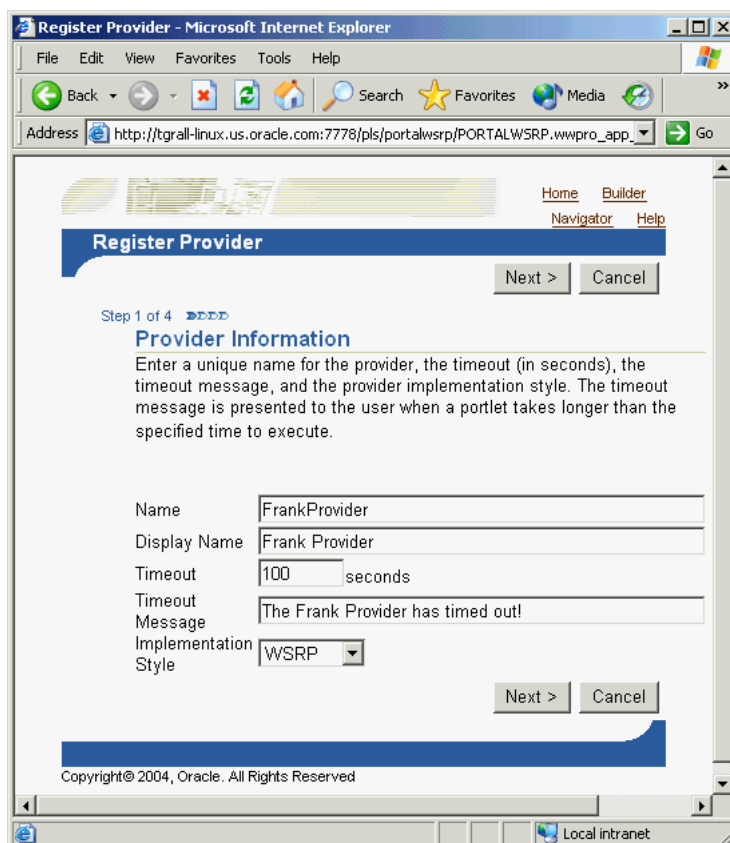
ローカルの OracleAS Portal インスタンスへの登録 WSRP 準拠の OracleAS Portal のローカル・インスタンスがある場合は、次の手順でポートレットを登録できます。

1. OracleAS Portal を開き、通常どおりログインします。プロバイダを登録するには、プロバイダに対する管理権限または編集権限が必要です。これらの権限がない場合は、管理者に要求する必要があります。
2. 「ビルダー」ページで作業していない場合は、右上隅にある「**ビルダー**」をクリックします。
3. 「**管理**」タブをクリックします。
4. 「**ポートレット**」サブタブをクリックします。
5. 「リモート・プロバイダ」ポートレットで、「**プロバイダの登録**」をクリックします。
6. [図 5-15](#) に示す「プロバイダの登録」ページで、[表 5-4](#) に示す値を入力します。

表 5-4 「プロバイダの登録」 ページの値

設定	値
名前	<code>your_nameProvider</code>
表示名	<code>your_name Provider</code>
タイムアウト	100
タイムアウト・メッセージ	<code>The your_name Provider has timed out!</code>
実装スタイル	WSRP

図 5-15 「プロバイダの登録」 ページ

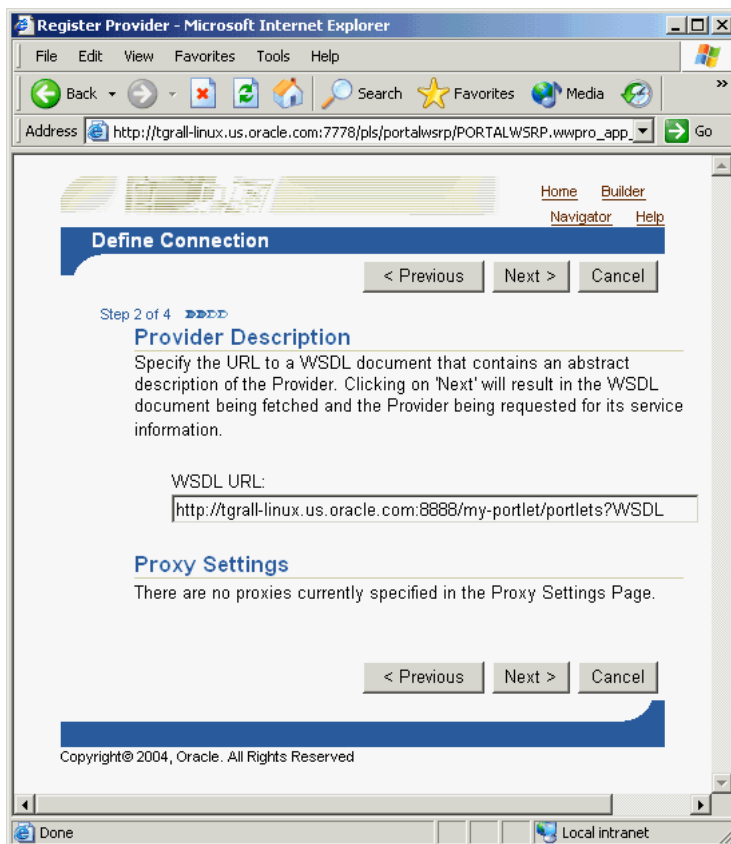


7. 「次へ」をクリックします。

8. 図 5-16 に示す「接続の定義」ページで、「WSDL URL」フィールドにプロバイダの WSDL URL を入力します。この URL は、5-28 ページの「WAR ファイルのデプロイ」の手順 8 で作成したものです。次に例を示します。

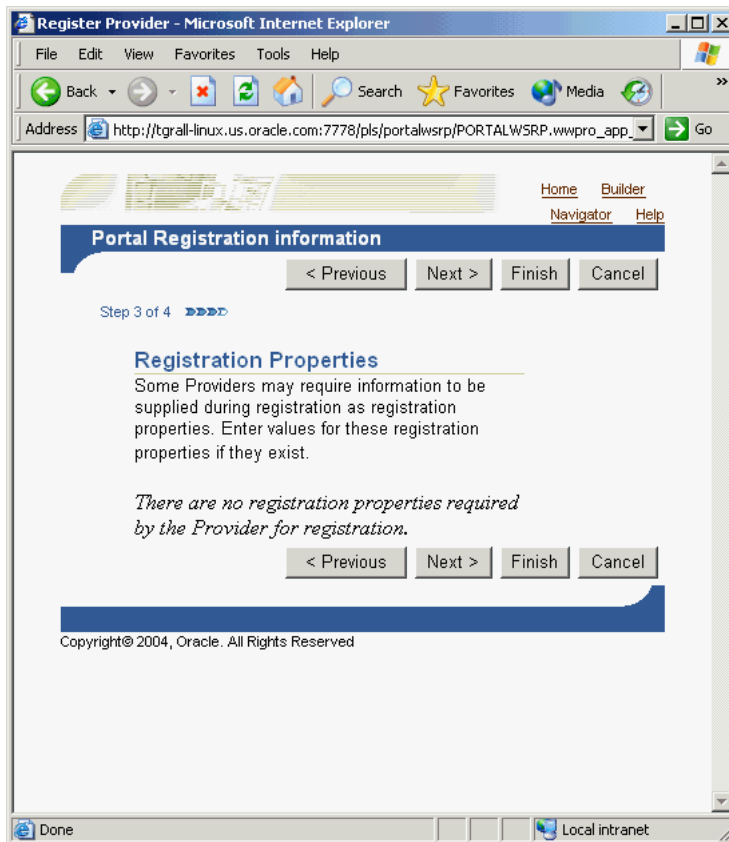
`http://myserver.com:8888/my-portlet/portlets?WSDL`

図 5-16 「接続の定義」ページ



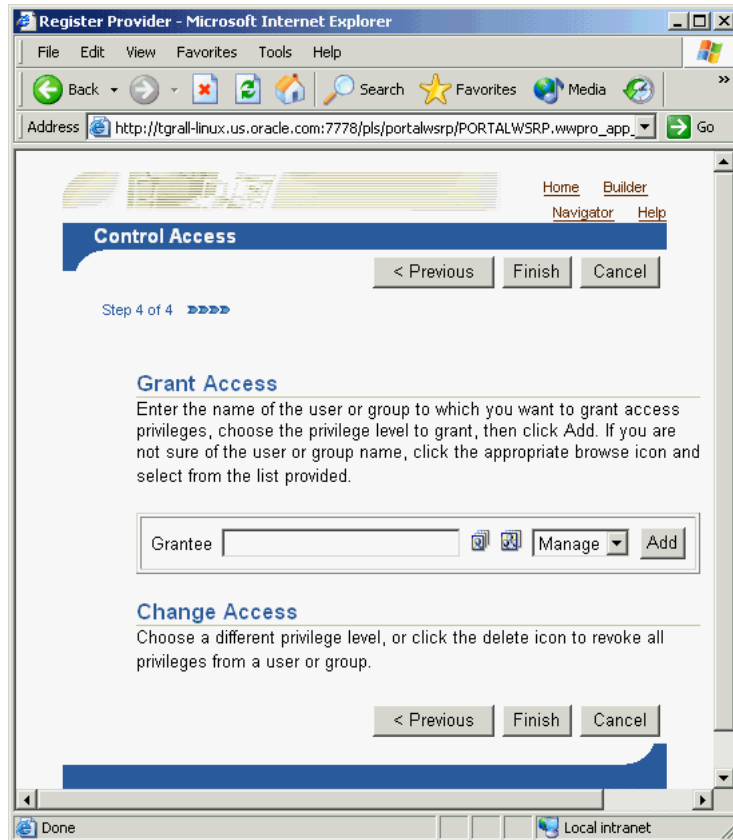
9. 「次へ」をクリックします。
10. 図 5-17 に示すポータル登録情報ページで、プロバイダに必要な登録プロパティを入力します。必要な登録プロパティがない場合は、次の手順に進みます。

図 5-17 ポータル登録情報ページ



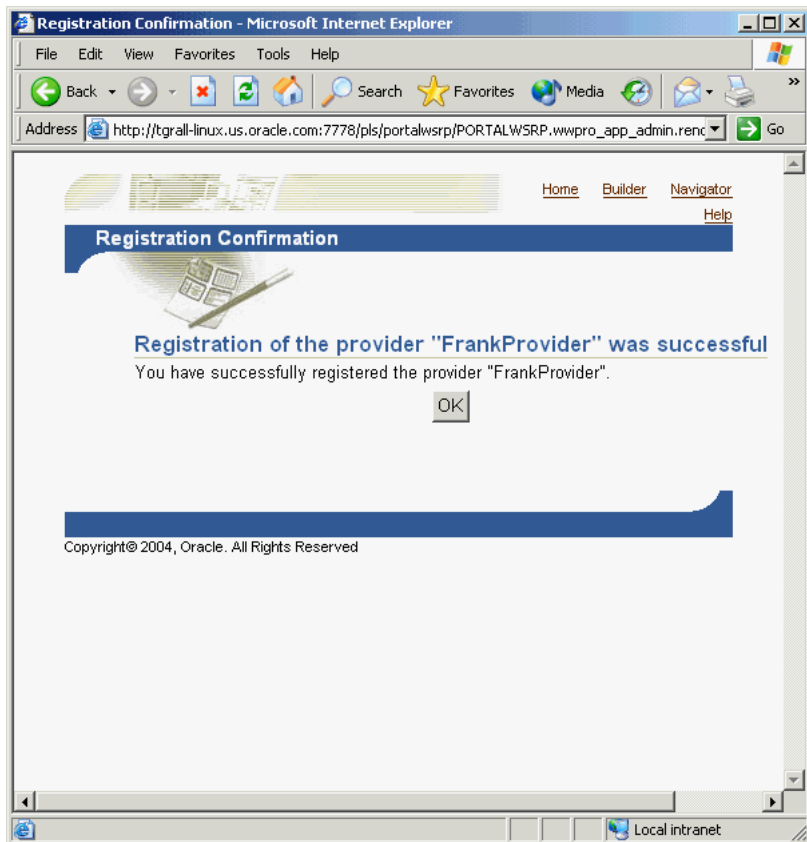
11. 「次へ」をクリックします。
12. 必要に応じて、図 5-18 に示す「アクセス制御」ページで、このプロバイダに対する権限を割り当てます。この例では、「ユーザー/ロール」にグループ（たとえば、PORTAL_ADMINISTRATORS）を入力し、「追加」をクリックします。

図 5-18 「アクセス制御」 ページ



13. 「完了」 をクリックします。図 5-19 に示すような「登録確認」 ページが表示されます。

図 5-19 「登録確認」 ページ



portalstandards.com への登録 WSRP 準拠の OracleAS Portal のローカル・インスタンスがない場合は、次の手順でポータルレットを登録できます。

1. 次の OTN サイトの OracleAS Portal 検証サービスにアクセスします。
<http://portalstandards.oracle.com/>
2. 次の OTN サイトにあるドキュメント『How to Test Interoperability』の指示に従ってください。

<http://portalstandards.oracle.com/pls/wsrc/docs/PAGE/WSRPPORTALPAGE/VERIFICATIONSERVER/TAB31866/HOW.TO.REGISTER.WSRP.PROVIDERS.HTML>

ポートレットの追加 これでは、ポートレット・リポジトリ内のポートレットと同様に、ポートレットをページに追加できます。ポートレットをページに追加するには、『OracleAS Portal ユーザーズ・ガイド』の「ポートレットの追加」の指示に従ってください。

Oracle JDeveloper を使用した PDK-Java ポートレットの構築

ポートレットを構築する際は、OTN から拡張機能をダウンロードして Oracle JDeveloper に追加できます。この拡張機能を使用するには、次の手順を実行します。

- [Oracle JDeveloper ポータル・アドインのインストール](#)
- [PDK-Java ポートレットの構築](#)

この項では、次のことを前提にしています。

- ポートレットの表示モードなど、ポートレットに関する用語の知識があること。[第 1 章「ポートレットの概要」](#) および 5-2 ページの「[Java ポートレット作成のガイドライン](#)」を参照してください。
- Java ポートレット・コンテナをダウンロードしてインストール済で、ポートレットをデプロイする Oracle Application Server Containers for J2EE 9.0.4 のコンテナが存在すること。
- Oracle JDeveloper に関する知識があり、それを使用して Java コンポーネントを構築およびデプロイする方法を習得していること。Oracle JDeveloper は OTN-J からダウンロードできます。

Oracle JDeveloper ポータル・アドインのインストール

Oracle JDeveloper ポータル・アドインのインストール方法は、5-14 ページの「[Oracle JDeveloper ポータル・アドインのインストール](#)」を参照してください。

PDK-Java ポートレットの構築

Oracle JDeveloper ポータル・アドインを正常にインストールした後は、Oracle JDeveloper を使用して、次のようにポートレットを迅速かつ簡単に開発できます。

- [ポートレットとプロバイダの作成](#): ポートレット・ウィザードを使用して、基本的なポートレット・コードとプロバイダ・フレームワークに必要な構成ファイルを作成します。
- [ポートレット・ロジックの追加](#): 独自のビジネス・ロジックを使用してサンプル・コードを拡張します。
- [ポートレットとプロバイダの検証](#): Oracle JDeveloper の組込み J2EE サーバーを使用して、プロバイダとそのポートレットの構成を検証できます。

- **アプリケーション・サーバーへのデプロイ** : Oracle JDeveloper を使用して、作成したアプリケーションをアプリケーション・サーバーにデプロイします。
- **ポートレットの登録と表示** : ポートレットをローカルの OracleAS Portal インスタンスに登録して表示します。

ポートレットとプロバイダの作成

この項では、ポートレット・ウィザードの手順を説明します。最初に、実装するポートレットの表示モード、および実装メソッド (JSP、HTTP サーブレット、Java クラスまたは HTML) を選択できます。次に、ウィザードによって、選択したモードごとに簡単なサンプル実装が作成されます。

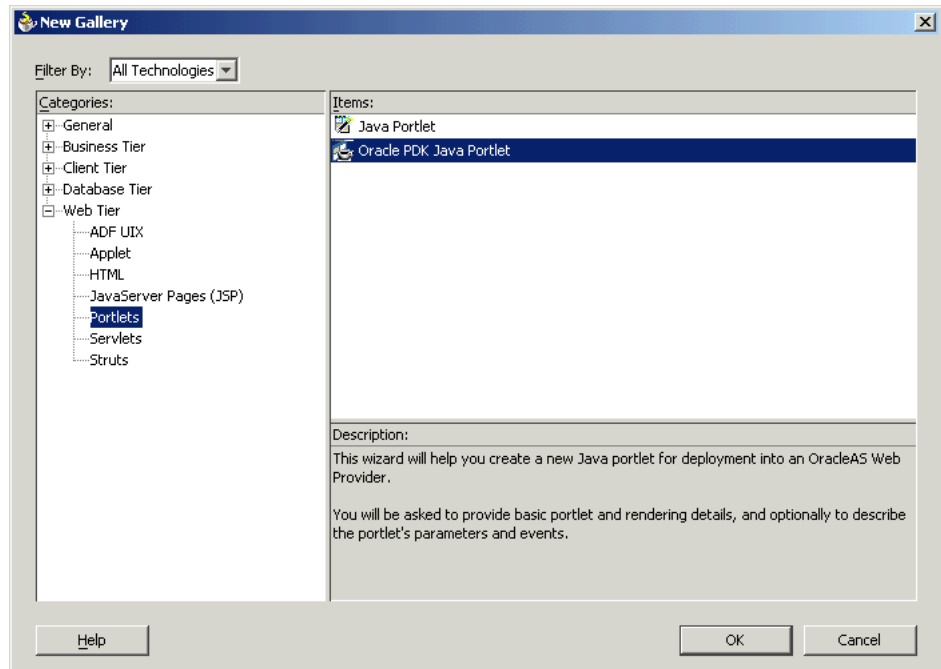
1. Oracle JDeveloper を開いて、このプロジェクトを作成するワークスペースをクリックします。ワークスペースがない場合は、次の手順に従ってワークスペースを作成できます。
 - a. 「アプリケーション」 → 「ナビゲータ」で「**アプリケーション**」ノードを右クリックし、ポップアップ・メニューから「**新規**」を選択します。
 - b. 「新規ギャラリー」ダイアログ・ボックスの「項目」リストから「**ワークスペース**」を選択します。
 - c. 「OK」をクリックします。
 - d. 「ワークスペース名」および「**ディレクトリ名**」を入力し、ワークスペース作成ダイアログ・ボックスの「**新規の空のプロジェクトを追加**」の選択を解除します。
 - e. 「OK」をクリックします。
2. 「アプリケーション」 → 「ナビゲータ」でワークスペースの名前を右クリックし、ポップアップ・メニューから「**新規プロジェクト**」を選択します。
3. 「新規ギャラリー」ダイアログ・ボックスの「項目」リストから「**空のプロジェクト**」をクリックします。
4. 「OK」をクリックします。
5. ワークスペース作成ダイアログ・ボックスで「**プロジェクト名**」および「**ディレクトリ名**」を入力します。
6. 「OK」をクリックします。
7. プロジェクトを右クリックし、ポップアップ・メニューから「**新規**」を選択します。
8. 「カテゴリ」リストで「Web Tier」カテゴリを開き、「**ポートレット**」をクリックします。

注意: 「ポートレット」が表示されていない場合は、5-14 ページの「[Oracle JDeveloper ポータル・アドインのインストール](#)」を参照して、アドインが正しくインストールされていることを確認してください。

9. 「項目」リストで「**Oracle PDK Java Portlet**」をクリックします。

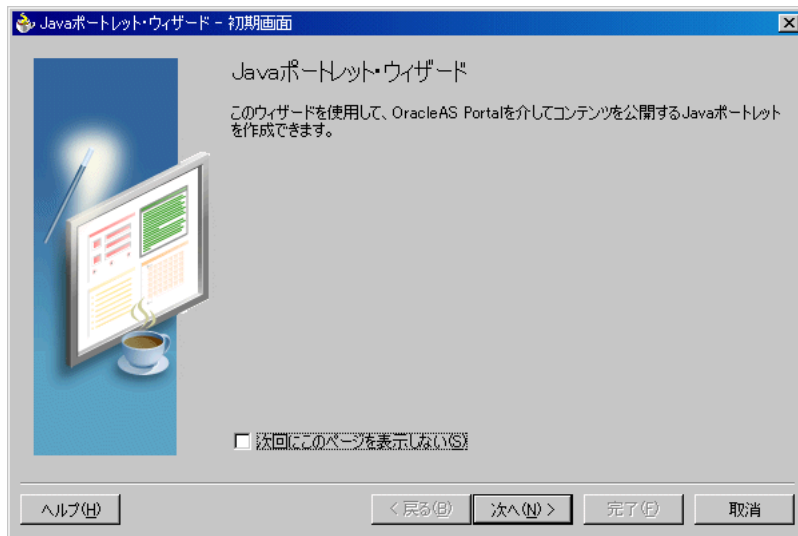
注意：「**Java Portlet**」をクリックすると、JPS 準拠のポートレットを作成するためのポートレット・ウィザードがオープンします。「**Oracle PDK Java Portlet**」をクリックすると、PDK-Java ポートレットを作成するためのポートレット・ウィザードがオープンします。

図 5-20 Oracle PDK Java ポートレットの「新規ギャラリー」ダイアログ・ボックス



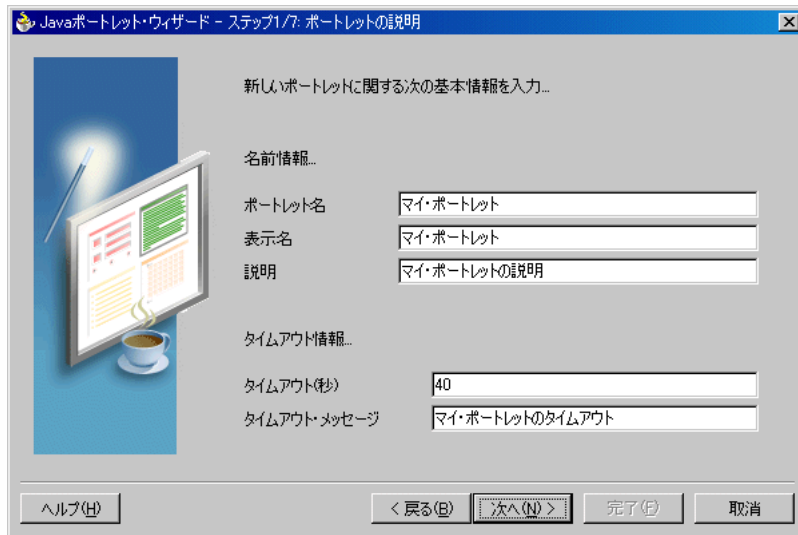
10. 「OK」をクリックします。ポートレット・ウィザードが表示されます。
11. ウィザードの「初期画面」ページが表示された後、「次へ」をクリックします。

図 5-21 「初期画面」 ページ



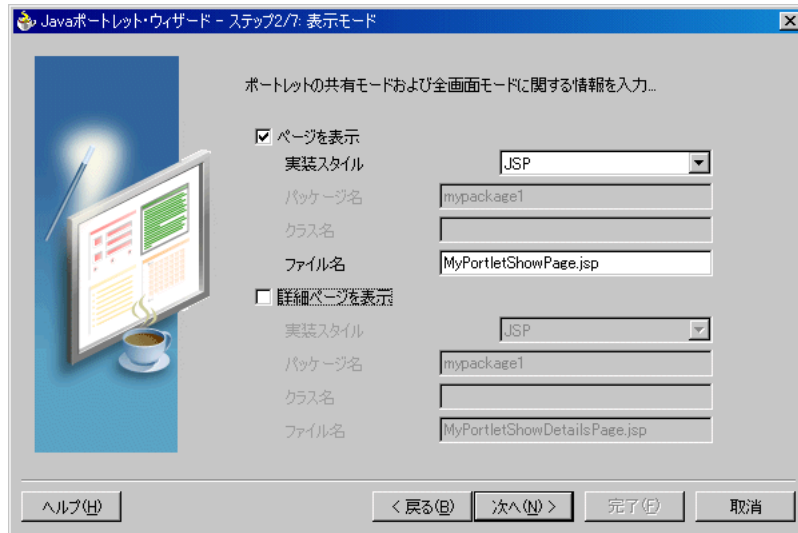
12. 図 5-22 に示す「ポートレットの説明」ページで、名前、説明およびタイムアウト設定を入力します。この例では、このページのデフォルト値をそのまま使用します。

図 5-22 「ポートレットの説明」 ページ



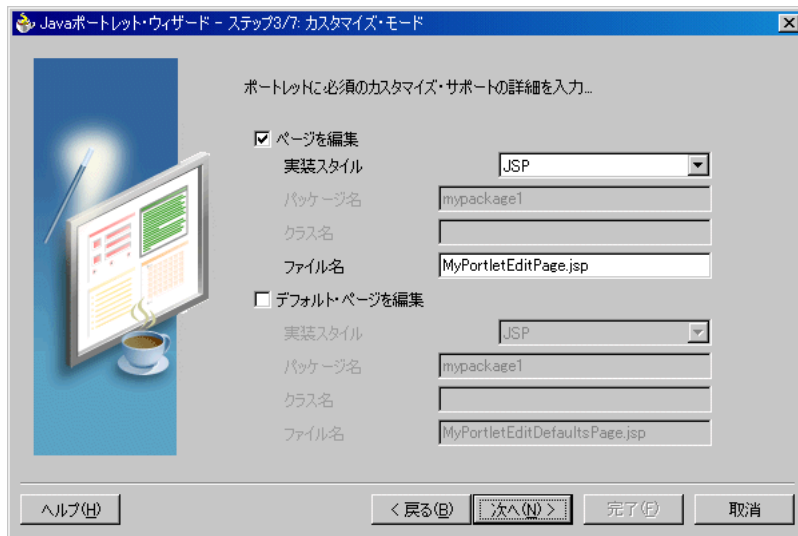
13. 「次へ」をクリックします。
14. ウィザードの次の各ページでは、このポートレットのポートレット・モードを定義できます。図 5-23 に示す「表示モード」ページでは、「ページを表示」に対する実装スタイル、および「詳細ページを表示」を実装するかどうかを選択できます。
 - a. デフォルトでは、「ページを表示」が選択されています。リストから実装スタイルを選択します。この例では、「JSP」を選択します。次に、ポートレット・ウィザードで生成する JSP のファイル名を入力します。この例では、デフォルトのファイル名をそのまま使用します。
 - b. ポートレットに詳細ページが必要な場合は、「詳細ページを表示」を選択し、適切な実装スタイルおよびファイル名を入力します。この例では、詳細ページを表示する必要はありません。

図 5-23 「表示モード」ページ



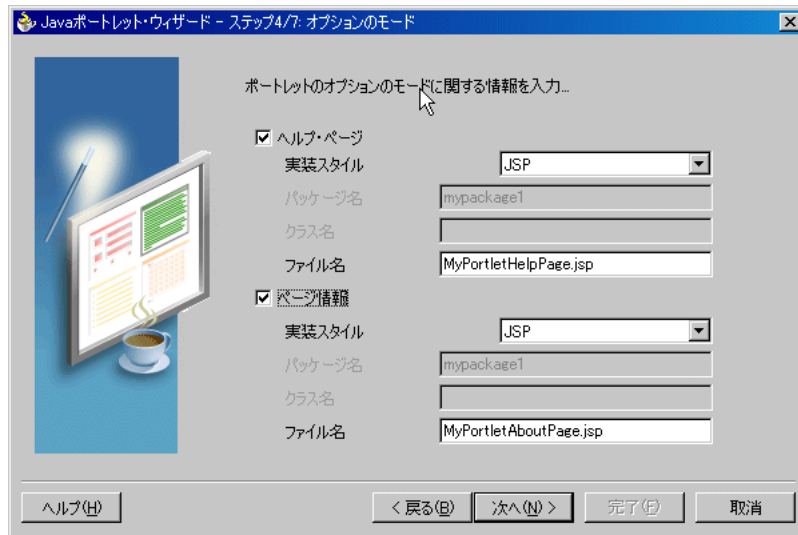
15. 「次へ」をクリックします。
16. 図 5-24 に示す「カスタマイズ・モード」ページでは、「ページを編集」がすでに選択されています。この例では、「実装スタイル」で「JSP」を選択し、デフォルトのファイル名をそのまま使用します。
17. 「デフォルト・ページを編集」を選択します。この例では、「実装スタイル」で「JSP」を選択し、デフォルトのファイル名をそのまま使用します。

図 5-24 「カスタマイズ・モード」 ページ



18. 「次へ」をクリックします。
19. 図 5-25 に示すオプションのモード・ページでは、デフォルトで選択されている項目はありません。この例では、「ヘルプ・ページ」を選択し、「実装スタイル」で「JSP」を選択して、デフォルトのファイル名をそのまま使用します。
20. 「ページ情報」を選択し、「実装スタイル」で「JSP」を選択して、デフォルトのファイル名をそのまま使用します。

図 5-25 オプションのモード・ページ

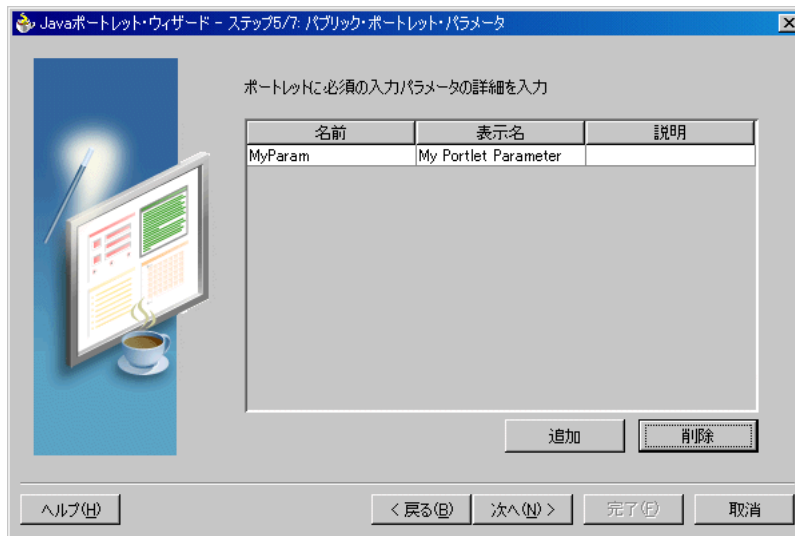


21. 「次へ」をクリックします。
22. 図 5-26 に示す「パブリック・ポートレット・パラメータ」ページで、「追加」をクリックします。
23. 表 5-5 および図 5-26 に示す値を入力します。

表 5-5 パブリック・ポートレット・パラメータ

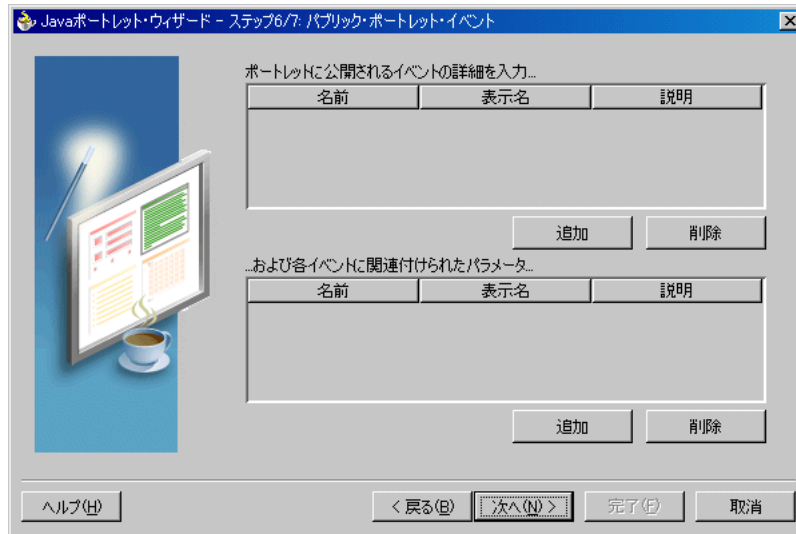
名前	表示名	説明
MyParam	My Portlet Parameter	このパラメータは値を表示します。

図 5-26 「パブリック・ポートレット・パラメータ」 ページ



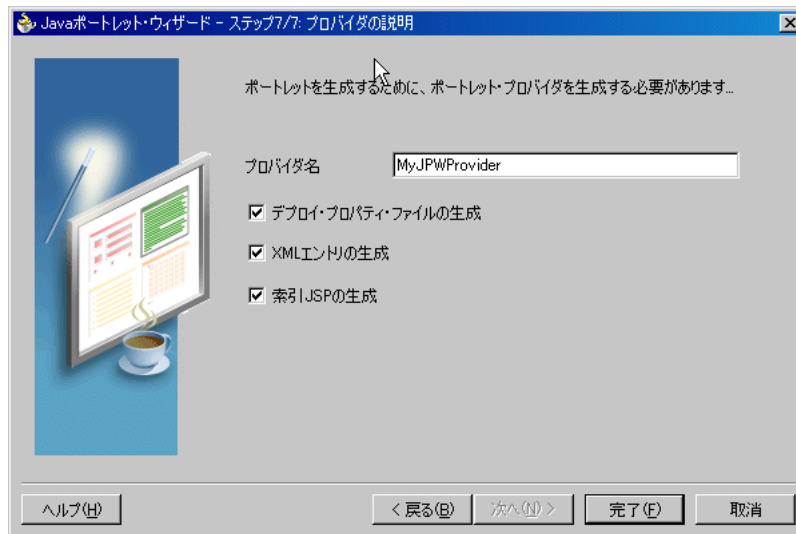
24. 「次へ」をクリックします。
25. 図 5-27 に示す「パブリック・ポートレット・イベント」ページで、パラメータをイベントにマップできます。この例では、このページを空のままにして「次へ」をクリックします。

図 5-27 「パブリック・ポートレット・イベント」 ページ



26. 図 5-28 に示す「プロバイダの説明」ページで、「プロバイダ名」に MyJPWProvider と入力します。すべてのチェック・ボックスが選択されていることを確認します。

図 5-28 「プロバイダの説明」 ページ

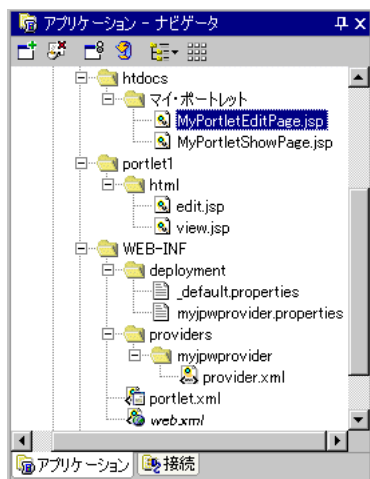


27. 「完了」をクリックすると、ポートレット用のファイルが生成されます。この例では、「アプリケーション - ナビゲータ」のプロジェクトに対して次のファイルが生成されます (図 5-29 を参照)。

- 選択した各ポートレット・モード用のファイル
- provider.xml
- web.xml
- index.jsp
- _default.properties
- myjpwprovider.properties

これらすべてのファイルは、ポートレットを正常にデプロイして実行するために必要です。ただし、index.jsp は Oracle JDeveloper でテスト用に使用されます。

図 5-29 「アプリケーション - ナビゲータ」



ポートレット・ロジックの追加

デフォルト実装を作成した後は、ビジネス・ロジックを追加してサンプル・コードを拡張し、必要な機能を実装できます。この例では、この手順を実行する必要はありません。次のテストと登録の手順に進んでください。

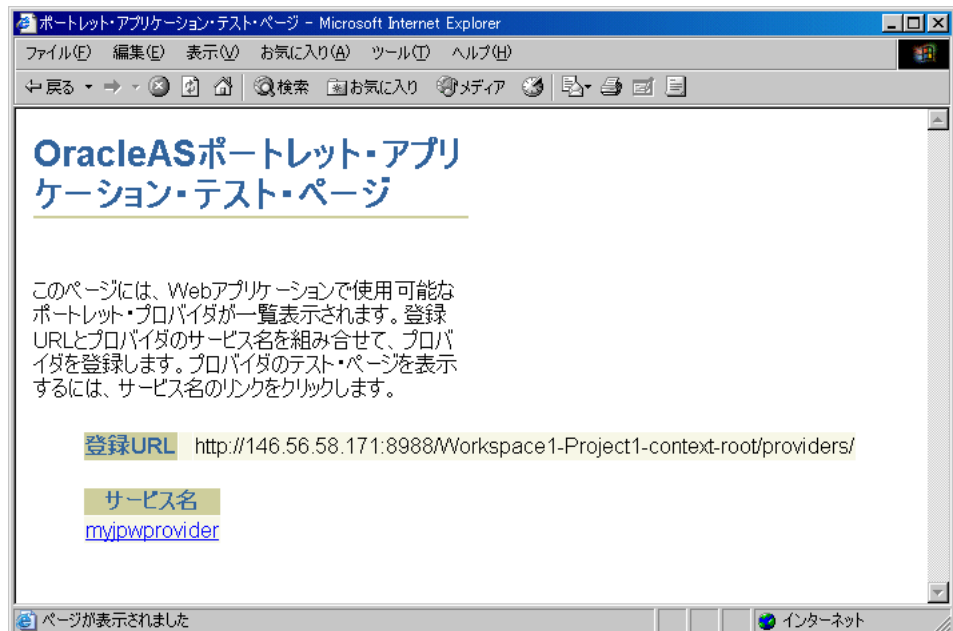
ポートレットとプロバイダの検証

ポートレットを作成した後は、構成をチェックして、ポートレットとそのプロバイダが正しく動作することを確認する必要があります。

注意： この手順は、テスト目的でのみ実行します。この手順を実行した後は、5-50 ページの「ポートレットの登録と表示」で説明するように、プロバイダを登録する必要があります。また、開発と本稼働では、5-46 ページの「アプリケーション・サーバーへのデプロイ」で説明するように、必ずポートレットをアプリケーション・サーバーにデプロイする必要があります。

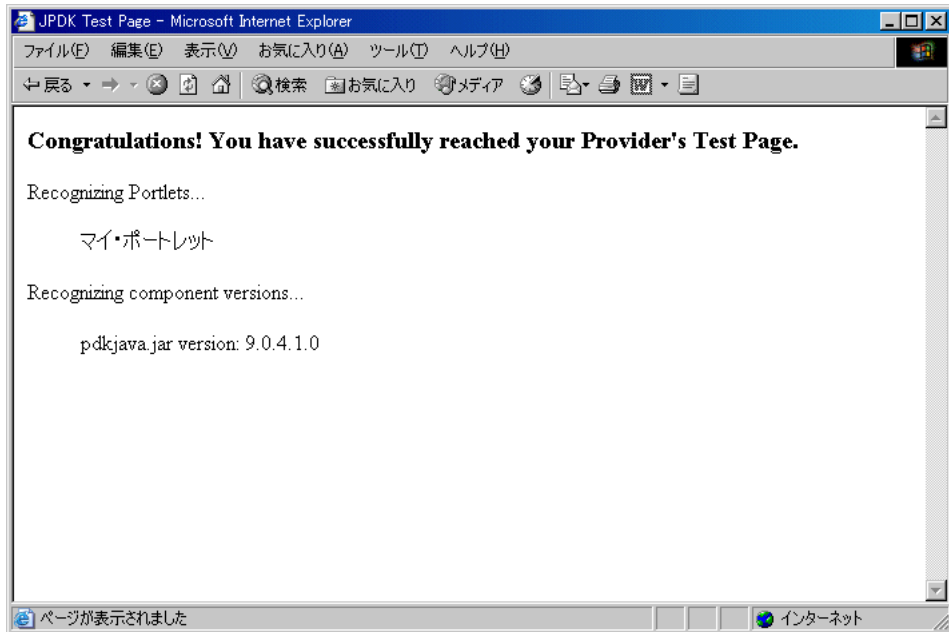
1. Oracle JDeveloper が開いていない場合は開き、前項で作成したプロジェクトを開きます。
2. ナビゲータで、ポートレットの `index.jsp` ファイルを検索して右クリックします。
3. コンテキスト・メニューから「実行」を選択します。ブラウザが開き、[図 5-30](#) に示すようなページが表示されます。

図 5-30 「ポートレット・アプリケーション・テスト・ページ」



4. 「サービス名」の下にあるリンクをクリックします。ブラウザが開き、[図 5-31](#) に示すようなページが表示されます。次の手順でプロバイダを登録する際は、このページに表示される URL が必要になることに注意してください。

図 5-31 プロバイダ・テスト・ページ



アプリケーション・サーバーへのデプロイ

ウィザードを終了してポートレットを生成した後は、そのポートレットを、アプリケーション・サーバーである Oracle Application Server Containers for J2EE にデプロイできます。次に、ポートレットを Oracle Application Server Containers for J2EE にデプロイする方法を説明します。

- [Oracle Application Server Containers for J2EE への接続の確立](#)
- [WAR ファイルのデプロイ](#)

Oracle Application Server Containers for J2EE への接続の確立 アプリケーション・サーバーへの接続を確立する手順は、次のとおりです。

注意： 次の各手順では、Oracle Application Server Containers for J2EE のスタンドアロン・インスタンスにデプロイする手順を説明します。完全な Oracle Application Server インスタンスへのデプロイ方法は、Oracle JDeveloper のオンライン・ヘルプ・システムを参照してください。

1. Oracle JDeveloper が開いていない場合は起動し、5-36 ページの「[ポートレットとプロバイダの作成](#)」で作成したプロジェクトを開きます。
2. ナビゲータで、「[接続](#)」を右クリックし、コンテキスト・メニューから「[アプリケーション・サーバー接続の作成](#)」を選択します。次の手順に従って表示されているウィザードを終了します。
 - a. 「初期画面」ページが表示されている場合は、「[次へ](#)」をクリックします。今後、「初期画面」ページを表示しない場合は、「[次回にこのページを表示しない](#)」を選択してください。
 - b. わかりやすい接続名（たとえば、PDKJavaOC4J）を入力し、接続タイプとして「[スタンドアロン OC4J](#)」を選択します。
 - c. 「[次へ](#)」をクリックします。
 - d. 管理者のユーザー名とパスワードを入力します。このパスワードは、Oracle Application Server Containers for J2EE のインストール時に設定されています。
 - e. 「[次へ](#)」をクリックします。
 - f. [表 5-6](#) に示す情報を入力します。

表 5-6 アプリケーション・サーバー接続の作成時の設定

設定	値
URL	この設定の完全な RMI URL を入力します。次に例を示します。 <code>ormi://my.machine.com:23791/</code> RMI ポート番号は、次のファイルを参照してください。 <code>OC4J_HOME/j2ee/home/config/rmi.xml</code>
宛先の Web サイト	デプロイした J2EE アプリケーション・ファイルが含まれる宛先の Web サイトの名前を入力します。この例では、デフォルト値の <code>http-web-site</code> をそのまま使用します。

表 5-6 アプリケーション・サーバー接続の作成時の設定（続き）

設定	値
OC4J の admin.jar がインストールされているローカル・ディレクトリ	<p>前述の URL 設定で指定したリモート・サーバーとバージョンの互換性がある、ローカルの admin.jar へのパスを入力します。</p> <p>Oracle Application Server Containers for J2EE 9.0.3 にデプロイするポートレットの場合は、Oracle JDeveloper 9.0.3 に含まれるデフォルトの admin.jar を使用できます。Oracle Application Server Containers for J2EE 9.0.4 にデプロイするポートレットの場合は、Oracle Application Server Containers for J2EE リリース 9.0.4 のインスタンスを指すように、この設定のデフォルト・パスを変更する必要があります。次に例を示します。</p> <p>OC4J_HOME¥j2ee¥home</p>

- g. 「次へ」をクリックします。
- h. 「接続のテスト」をクリックして、接続詳細を確認します。接続が正常な場合は、成功メッセージが表示されます。テストに失敗した場合は、接続情報の修正が必要になる場合があります。
- i. 「完了」をクリックします。

WAR ファイルのデプロイ WAR ファイルを作成してデプロイする手順は、次のとおりです。

1. ポートレット・プロジェクトを右クリックし、コンテキスト・メニューから「新規」を選択します。
2. 「新規」ダイアログ・ボックスの「カテゴリ」で「一般」→「デプロイメント・プロファイル」を選択し、「項目」で「WAR ファイル - J2EE Web モジュール」を選択します。
3. 「デプロイメント・プロファイルの保存」ダイアログ・ボックスで、名前をわかりやすい名前（たとえば、myj2eeportlet1.deploy）に変更します。
4. 「OK」をクリックします。
5. プロファイル設定ダイアログ・ボックスで、次の手順を実行します。
 - a. 「J2EE Web コンテキスト・ルートを指定」をクリックし、myj2eeportlet1 と入力します。
 - b. 左側のペインで、「ファイル・グループ」→「WEB-INF/lib」→作成者を選択します。
 - c. ポートレット開発をチェックします。
 - d. 「OK」をクリックします。
 - e. 「ファイル」→「すべて保存」を選択します。

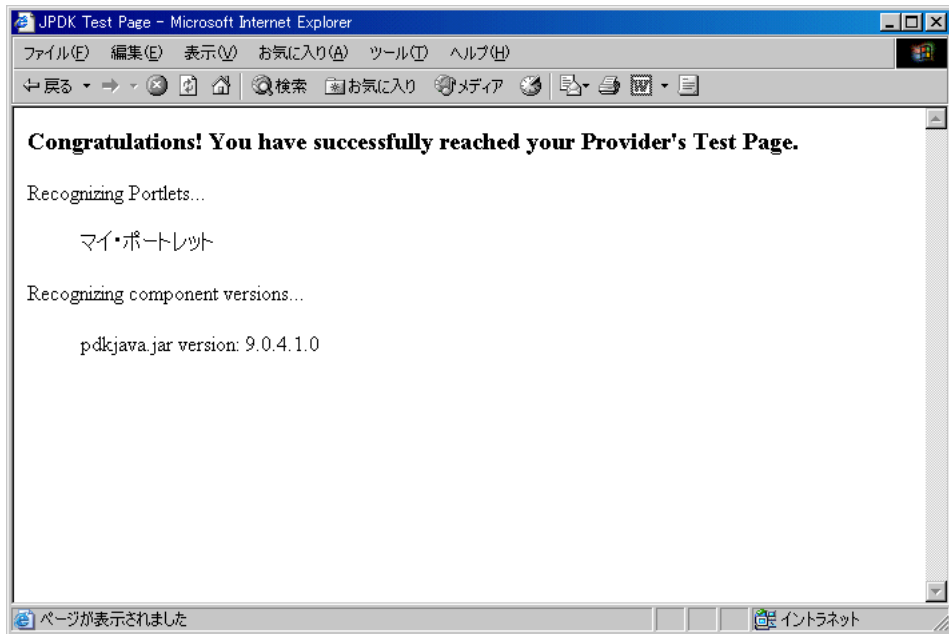
6. デプロイメント・プロファイル（たとえば、myj2eeportlet1.deploy）を右クリックし、コンテキスト・メニューから「**配布先**」→アプリケーション・サーバー接続（たとえば、PDKJavaOC4J）を選択します。
7. Oracle JDeveloper の下部にあるデプロイ・ログに「デプロイが完了」メッセージが表示されるのを待ち、エラーが発生していないことを確認します。
8. **デプロイ・ログ**に表示された URL（たとえば、`http://myserver.com:8888/myj2eeportlet1`）に `/providers` を追加して、J2EE ポートレットをテストして OracleAS Portal に登録する際に使用する URL を作成します。次に例を示します。

```
http://myserver.com:8888/myj2eeportlet1/providers
```

注意： ログに、Oracle JDeveloper でリモート・サーバーの HTTP ポート番号を判別できないというメッセージが表示される場合があります。通常は、Oracle Application Server Containers for J2EE のホーム・ページにアクセスする URL（たとえば、`http://myserver.com:8888`）を調べるとポート番号を判別できます。

9. 前の手順で作成した URL をブラウザに入力します。図 5-32 に示すようなページが表示されます。

図 5-32 PDK-Java ポートレットのテスト・ページ



ポートレットの登録と表示

プロバイダとそのポートレットを作成してデプロイした後は、そのプロバイダを OracleAS Portal に登録する必要があります。プロバイダを登録すると、そのプロバイダを検索して通信するために必要な情報が OracleAS Portal に提供されます。プロバイダを登録した後は、プロバイダとそのポートレットがポートレット・リポジトリで使用可能になります。また、OracleAS Portal のナビゲータにもプロバイダとそのポートレットのリストが表示されます。

注意： ポートレット・ビルダーなどの組込みツールを使用してポートレットやプロバイダを作成すると、OracleAS Portal によってそのプロバイダが自動的に登録されます。さらに、作成したポートレットは、ポートレット・リポジトリに自動的に表示されます。OracleAS Portal には組込みポートレットも用意されており、それらは事前構成済みのプロバイダに含まれています。たとえば、複雑な設定が不要なポートレットの OmniPortlet と Web Clipping は、すでに OracleAS Portal に登録されています。これらのポートレットは「ポートレットの追加」リストで参照できます。ただし、ポートレットとプロバイダをプログラムで作成した場合、そのプロバイダは、ポータル・ユーザーが使用できるように登録する必要があります。

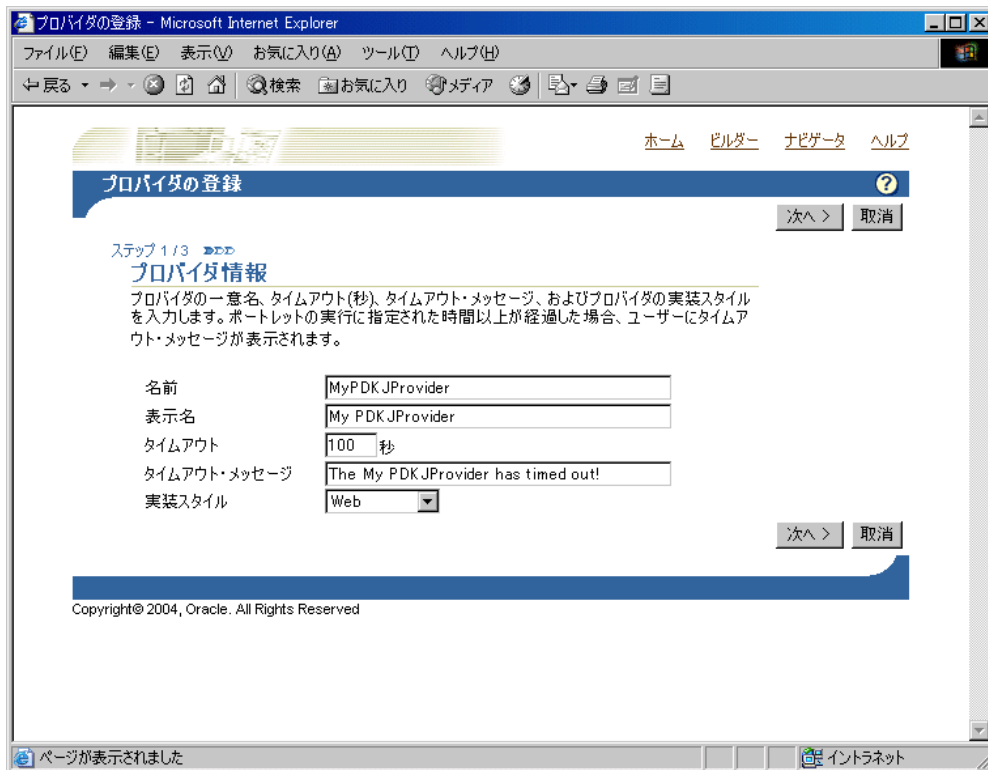
この項では、プロバイダを登録し、ポートレットをページに追加する方法を説明します。

1. OracleAS Portal を開き、通常どおりログインします。プロバイダを登録するには、プロバイダに対する管理権限または編集権限が必要です。これらの権限がない場合は、管理者に要求する必要があります。
2. 「ビルダー」 ページで作業していない場合は、右上隅にある「**ビルダー**」をクリックします。
3. 「**管理**」 タブをクリックします。
4. 「**ポートレット**」 サブタブをクリックします。
5. 「リモート・プロバイダ」 ポートレットで、「**プロバイダの登録**」をクリックします。
6. [図 5-33](#) に示す「プロバイダ情報」 ページで、[表 5-7](#) に示す値を入力します。

表 5-7 「プロバイダ情報」 ページの値

設定	値
名前	<code>your_namePDKJProvider</code>
表示名	<code>your_name PDKJProvider</code>
タイムアウト	100
タイムアウト・メッセージ	<code>The your_name PDKJProvider has timed out!</code>
実装スタイル	Web

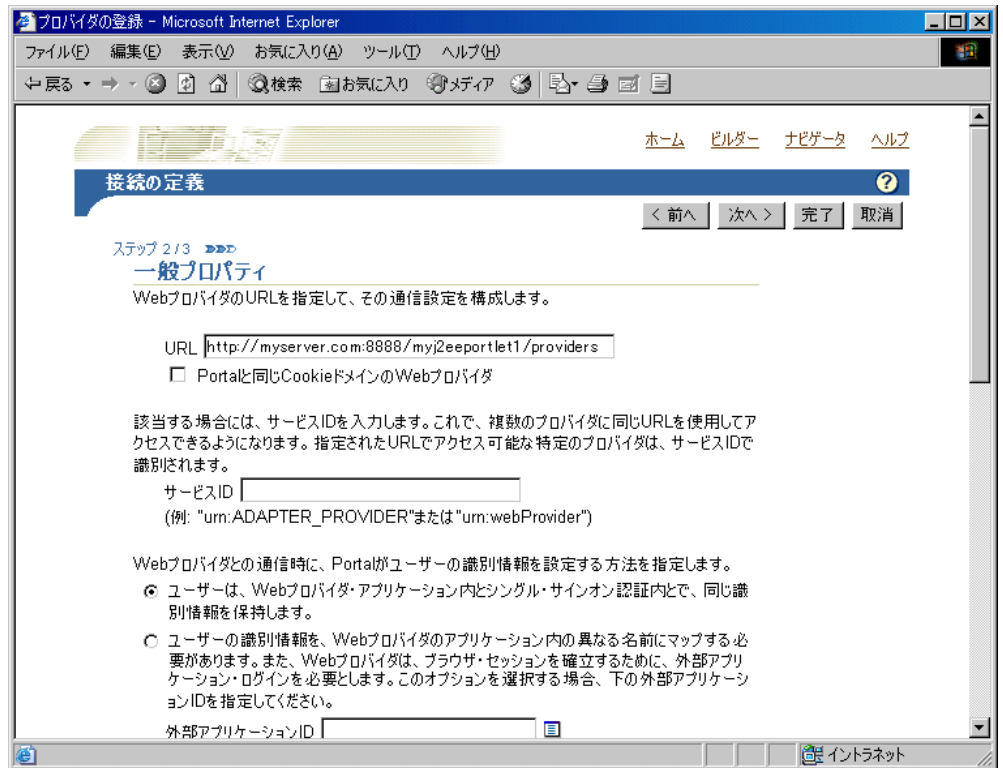
図 5-33 「プロバイダの登録」ページ



- 「次へ」をクリックします。
- 図 5-34 に示す「接続の定義」ページで、「URL」フィールドにプロバイダの URL を入力します。この URL は、5-48 ページの「WAR ファイルのデプロイ」の手順 8 で作成したものです。次に例を示します。

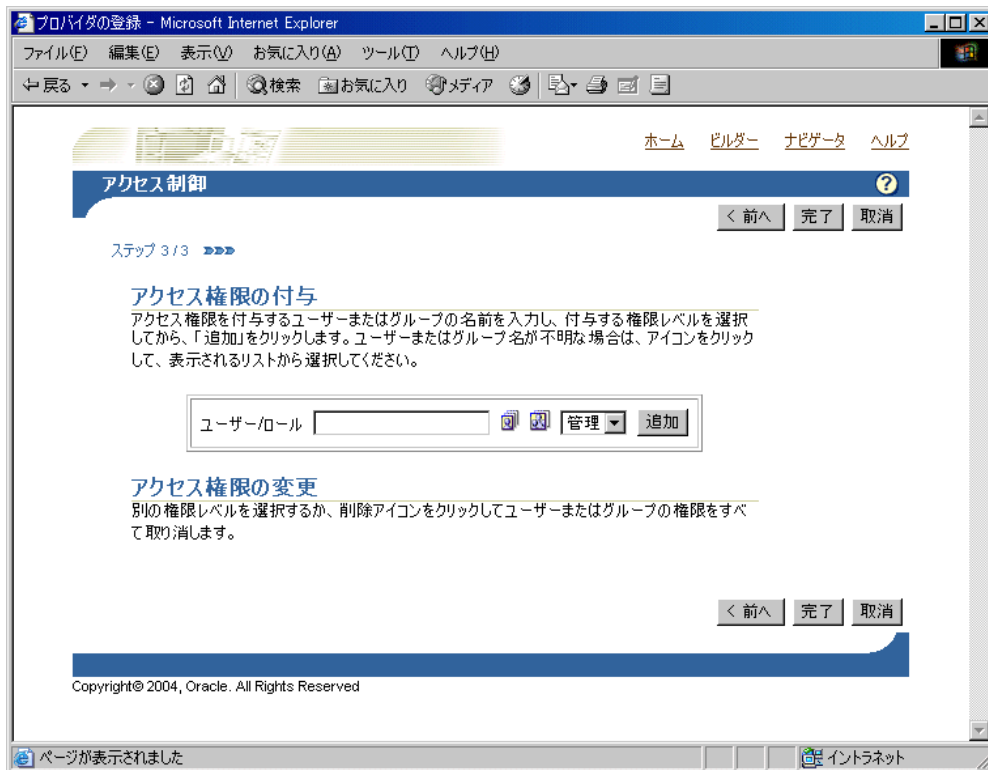
`http://myserver.com:8888/myj2eeportlet1/providers`

図 5-34 「接続の定義」 ページ



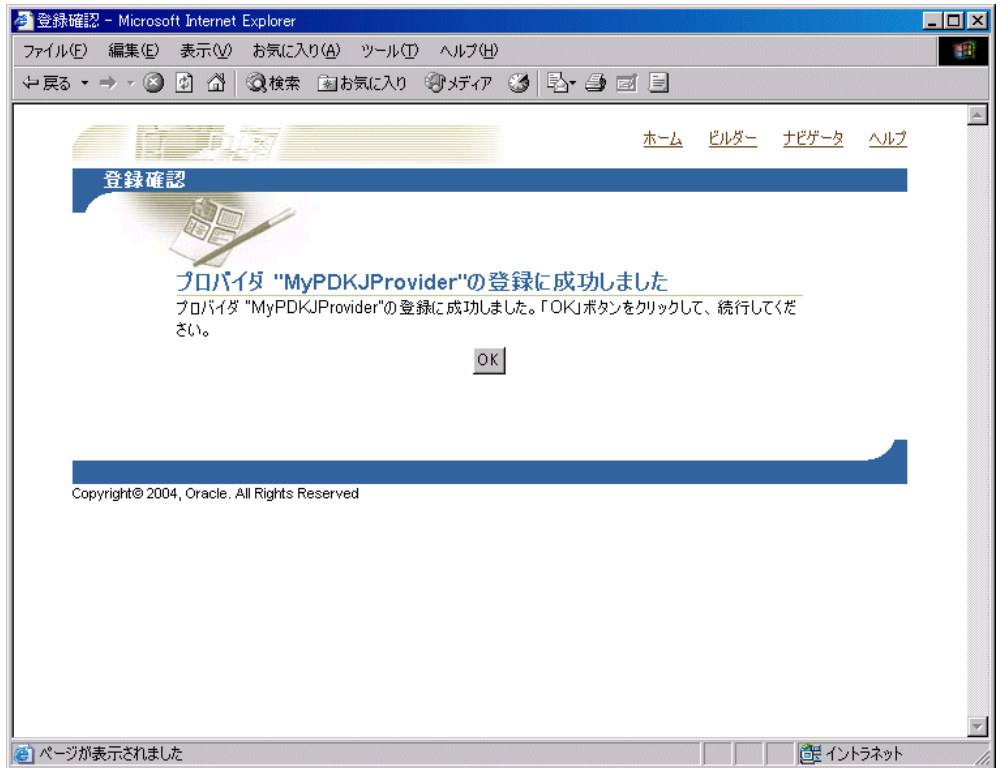
9. 「次へ」をクリックします。
10. 必要に応じて、図 5-35 に示す「アクセス制御」ページで、このプロバイダに対する権限を割り当てます。この例では、「ユーザー / ロール」にグループ（たとえば、PORTAL_ADMINISTRATORS）を入力し、「追加」をクリックします。

図 5-35 「アクセス制御」 ページ



11. 「完了」をクリックします。図 5-36 に示すような「登録確認」ページが表示されます。

図 5-36 「登録確認」 ページ



- これで、ポートレット・リポジトリ内のポートレットと同様に、ポートレットをページに追加できます。ポートレットをページに追加するには、『Oracle Application Server Portal ユーザーズ・ガイド』の「ポートレットの追加」の指示に従ってください。

レンダリング・モードの追加

表示モードを追加するには、ポートレット・ウィザードの各ページで該当するボックスをチェックします。ウィザードの使用方法は、5-35 ページの「PDK-Java ポートレットの構築」を参照してください。ウィザードで選択した表示モードごとに、基本的な HelloWorld スケルトンが作成されています。ポートレットの作成後に表示モードを追加する必要がある場合やウィザードでは入手できないモード（プレビューまたはリンク）を追加する場合は、`provider.xml` および Oracle JDeveloper の HTML または JSP を更新して手動で追加できます。レンダリング・モードを追加する手順は、次のとおりです。

- 追加の表示モードの実装
- XML プロバイダ定義の更新
- ポートレットの表示

この項を終了すると、RenderManager を使用してあらゆるレンダリング・モードを実装できます。これは、すべてのモードの原理が同じであるためです。たとえば、この項ではヘルプ・モードの実装方法は詳しく説明しませんが、プロセスはここで説明するプレビュー・モードと同じであるため、方法は理解できます。

特別な**編集**と**デフォルト編集**の各レンダリング・モードをレンダリングするための追加要件は、OracleAS Portal の Oracle Technology Center にある記事「Adding Customization to Java Portlets」を参照してください。この記事で使用されている PDK ランタイム・クラスの詳細は、**JavaDoc** を参照してください。

前提

- ウィザードで作成したポートレットがページに正常に追加されていること。

追加の表示モードの実装

表示モードを手動で作成する際は、最初に、各モードの HTML ファイルまたは JSP を作成します。たとえば、プレビュー・モードを実装する場合は、HTML ファイルを作成してプレビュー・コンテンツを提供する必要があります。

HTML ファイルを作成してコンテンツをプレビューする手順は、次のとおりです。

1. Oracle JDeveloper で、ポートレットが含まれているプロジェクトを開き、「アプリケーション・ナビゲータ」でポートレットを選択して、HTML ページが適切な場所に作成されていることを確認します。Oracle JDeveloper のデザイン・ビューを使用すると、簡単に HTML ページを作成できます。たとえば、次の HTML はプレビュー・ページとして使用できます。

```
<p>This is the <i>preview</i> mode of your portlet!</p>
```

コンテンツをプレビューするための HTML ファイルを作成すると、XML プロバイダ定義を更新する準備が完了します。

XML プロバイダ定義の更新

追加のレンダリング・モードを公開する場合は、次のように XML プロバイダ定義を更新する必要があります。

- ブール・フラグを設定し、そのモードへのリンクまたはアイコンをレンダリングする必要があることを PDK フレームワークに示します。
- そのモードに対して作成した HTML ファイルまたは JSP を指定します。

たとえば、プレビュー・モードをレンダリングする手順は、次のとおりです。

1. プロバイダ定義ファイル `provider.xml` を編集し、タグを追加してプレビュー・モードを有効にします。

```
<showPreview>true</showPreview>
```

2. 5-56 ページの「追加の表示モードの実装」で作成した HTML ページをプレビュー・ページとして指定します。

```
<previewPage>/htdocs/myportlet/MyPortletPreviewPage.html</previewPage>
```

3. 更新内容を `provider.xml` に保存します。
4. ポートレットを再デプロイします。5-48 ページの「WAR ファイルのデプロイ」の手順 6 を参照してください。

ポートレットの表示

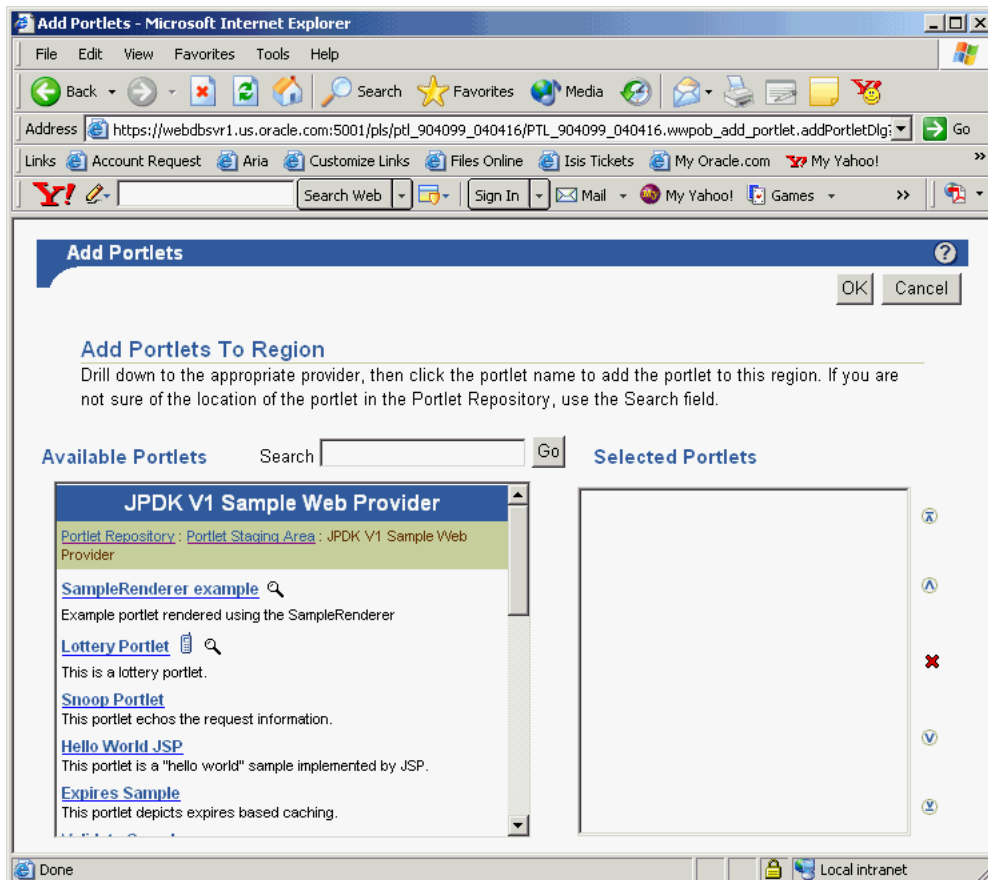
新規のレンダリング・モードを表示するには、更新した XML プロバイダ定義が再解析されるようにする必要があります。そのための手順は、次のとおりです。

1. 5-56 ページの「追加の表示モードの実装」で作成した HTML ファイルおよび `provider.xml` を、ポートレットをデプロイする Oracle Application Server Containers for J2EE インスタンスにコピーします。
2. プロバイダをリフレッシュします。
3. ポートレットが含まれているポータル・ページをリフレッシュします。

プレビュー・モードを表示する手順は、次のとおりです。

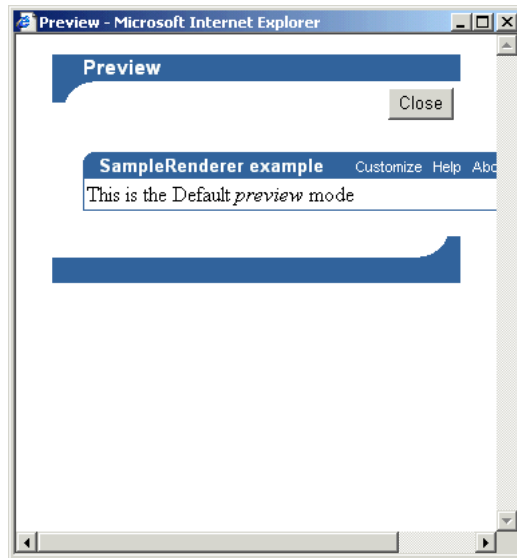
1. ページを編集するか、新規ページを作成して、「ポートレットの追加」を選択します。
2. プレビュー・モードを使用して、ポートレットの場所にナビゲートします（たとえば、Portlet ステージング領域）。図 5-37 に示すように、ポートレットの横にズーム・アイコンが表示されます。

図 5-37 「ポートレットの追加」 ページ



3. ポートレットの横にあるズーム・アイコンをクリックすると、図 5-38 に示すような「プレビュー」ウィンドウが表示されます。

図 5-38 「プレビュー」 ウィンドウ



ポートレットのカスタマイズ

5-56 ページの「レンダリング・モードの追加」では、PDK プロバイダ・フレームワークを使用して、ウィザードでポートレットを作成したときに無効だった表示モード、またはウィザードでは選択できない表示モード（リンク・モードとプレビュー・モード）を有効にしてレンダリングする方法を説明しました。この項では、2つのカスタマイズ・モード（編集およびデフォルト編集）を詳細に説明します。Java ポートレット・ウィザードで「ページを編集」と「デフォルト・ページを編集」をチェックすると、この2つのカスタマイズ・モードについてスケルトン・コードが生成されます。このスケルトン・コードを使用すると、数行のコードでパーソナライズ・フレームワークにアクセスでき、カスタマイズ・フレームワークおよび値を保持するためのデータ・ストアをすべて手動でコーディングする必要がありません。

ポートレットをカスタマイズするには、次の作業を実行する必要があります。

- ポートレットの「編集」ページを更新し、カスタマイズの変更を設定して取得します。
- ポートレットの「デフォルトの編集」ページを更新し、カスタマイズの変更を設定して取得します。
- ポートレットの「表示」ページを更新して、ユーザーが設定したカスタマイズを使用します。

ポートレット・ユーザーは、編集モードおよびデフォルト編集モードを使用して、ポートレットでサポートされている一連のカスタマイズ可能パラメータを変更できます。通常は、これらのパラメータによって、ポートレットを他のモードでレンダリングします。OracleAS Portal ページにあるポートレットの特定のインスタンスの場合、編集モードおよびデフォルト編集モードで行われたカスタマイズは、そのポートレットのそのインスタンスにのみ適用されます。

- **編集モード**でのカスタマイズは、カスタマイズを行う個別のユーザーに適用されます。このモードを有効にするには、表示モードで、ポートレット・ヘッダーにある「**カスタマイズ**」リンクをクリックします。
- **デフォルト編集モード**でのカスタマイズは、そのポートレット・インスタンスに対して特定のカスタマイズを行っていない、同じロケール内のすべてのユーザーに適用されます。通常、このモードを使用できるのはページ設計者のみで、ページ上の「**編集**」アイコンに従うことで有効にできます。

編集モードおよびデフォルト編集モードをレンダリングすると、PortletRenderer は、カスタマイズ・プロセスをサポートするために、次のいずれかのタスクを実行できます。

- **編集フォームのレンダリング**: ポートレットのカスタマイズ可能パラメータごとに、PortletRenderer は PortletPersonalizationManager を使用して現在の値を取得し、コントロールを HTML フォームでレンダリングします。これによって、現在の値を編集できます。
- **編集フォーム操作の処理**: 標準の編集フォーム・ヘッダーで「OK」または「適用」ボタンをクリックすると、PortletRenderer は PortletPersonalizationManager を使用して、編集フォームから送信されたカスタマイズ済パラメータを格納し、ブラウザを適切なポータル・ページにリダイレクトします。

したがって、PortletPersonalizationManager コントローラを使用することによって、PortletRenderer は、特定のポートレット・インスタンスやユーザーに適用されるカスタマイズ可能パラメータの現在の値を格納および取得できます。PDK フレームワークでは、一連のカスタマイズ済パラメータのコンテナとして PersonalizationObject の抽象化を使用し、一連のカスタマイズを格納するキーとして PortletReference を使用します。つまり、PortletPersonalizationManager は、特定の PortletReference の下にある永続的な PersonalizationObjects の格納と取得を可能にするメカニズムです。

プリファレンス・ストアは、OracleAS Portal の使用時に、ユーザー設定項目データ、ポートレットまたはプロバイダの設定、さらにポートレット・データなどの情報を格納するメカニズムです。ログアウトした後で再度ログインしても、以前保存したプリファレンスにアクセスできるという意味で、プリファレンス・ストアに格納された情報は永続的です。プリファレンス・ストアではユーザー設定項目情報が保持され、ユーザーがログインするたびにそのユーザーの設定項目が起動されます。PDK-Java には、PreferenceStore 実装を使用してカスタマイズ済データを永続化する PrefStorePersonalizationManager が用意されています。現在、PDK-Java には、DBPreferenceStore および FilePreferenceStore の 2 つの PreferenceStore 実装があります。DBPreferenceStore は、JDBC と互換性のあるリレーショナル・データベースを使用してデータを永続化し、FilePreferenceStore は、ファイル・システムを使用してデータを永続化します。

これらの実装の詳細は、**JavaDoc** を参照してください。

カスタマイズ機能をポートレットに追加するには、`PrefStorePersonalizationManager` を、デフォルトの `PersonalizationObject` 実装である `NameValuePersonalizationObject` とともに使用します。ユーザーがポートレット・タイトルをカスタマイズできるように、ウィザードではデフォルトで、編集モードおよびデフォルト編集モードに対して簡単な編集フォームが生成されます。この項では、ポータル・ユーザーがポートレット・メッセージをカスタマイズできるように、既存のコードを更新する方法を説明します。

前提

1. 前述した次の項を読み終えて理解していること。
 - [PDK-Java ポートレットの構築](#)
 - [レンダリング・モードの追加](#)
2. 「[ページを編集](#)」および「[デフォルト・ページを編集](#)」がチェックされた状態のウィザードで作成したポートレットをページに追加していること。

「編集」ページおよび「デフォルトの編集」ページでのカスタマイズの実装

ポートレットの「編集」ページは、ユーザーがポートレットをカスタマイズするときにコールされます。ウィザードで生成された JSP には、ポートレット・タイトルのカスタマイズに必要なコードがすべてデフォルトで含まれています。さらにカスタマイズする場合は、数行のコードを「編集」ページに挿入する必要があります。

生成されたコードの確認 ウィザードでは、デフォルトで次のコードが作成されます。

```
<%@page contentType="text/html; charset=windows-1252"
    import="oracle.portal.provider.v2.render.PortletRenderRequest"
    import="oracle.portal.provider.v2.http.HttpCommonConstants"
    import="oracle.portal.provider.v2.personalize.NameValuePersonalizationObject"
    import="oracle.portal.provider.v2.render.PortletRenderUtil"
%>

<%
    PortletRenderRequest pReq = (PortletRenderRequest)
        request.getAttribute(HttpCommonConstants.PORTLET_RENDER_REQUEST);
%>

<P>Hello <%=pReq.getUser().getName() %>.</P>
<P>This is the <b><i>Edit</i></b> render mode!</P>
<!-- This page both displays the customization
    form and processes it,. Display the form if
    there is no action parameter, process it
```

```

        otherwise --%>

<%
String actionParam = PortletRendererUtil.getEditFormParameter(pReq);
String action = request.getParameter(actionParam);
String title = request.getParameter("my2portlet_title");
NameValuePersonalizationObject data = (NameValuePersonalizationObject)
    PortletRendererUtil.getEditData(pReq);
// Cancel automatically redirects to the page, so
// will only receive OK or APPLY
if (action !=null)
{
    data.setPortletTitle(title);
    PortletRendererUtil.submitEditData(pReq, data);
    return;
}

// Otherwise just render the form.
title = data.getPortletTitle();
%>
<table border="0">
  <td width="20%">
    <p align="right">Title:</p>
  </td>
  <td width="80%">
    <input type="TEXT" name="my2portlet_title" value="<%= title %%">
  </td>
</table>

```

生成されたコードの変更 JSP には、ポートレット・タイトル用の入力フィールドが含まれています。このフィールドはポートレットの「カスタマイズ」ページを表し、ユーザーはこのページでポートレット・タイトルを更新できます。

1. 生成されたコード内のテーブルに続いて、ユーザーがポートレットの新しいメッセージを入力できるテキスト・フィールドとプロンプトを含む 2 番目のテーブルを追加します。

```

<table border="0">
  <tr>
    <td width="20%">
      <p align="right">Greeting:</p>
    </td>
    <td width="80%">
      <input type="TEXT" name="myportlet_greeting" value="<%= greeting %%">
    </td>
  </tr>
</table>

```

2. この HTML は、「編集」ページで新しいメッセージを入力するためのフィールドを指定します。この新しいメッセージは、ポートレットの共有画面モードで表示されます。次に、String title の下に、メッセージの値を取得する文字列を追加します。

```
String title = request.getParameter("my2portlet_title");  
String greeting = request.getParameter("myportlet_greeting");
```

3. ウィザードで「編集」ページが生成されると、パーソナライズ・フレームワークへのアクセスがページ・コードに自動的に含まれます。「編集」ページの最上部で、NameValuePersonalizationObject が宣言されています。OracleAS Portal にあるパーソナライズ用のこのフォームを使用すると、名前と値のペアを簡単に格納できます。

「編集」ページでは、ページの表示、またはページへの変更の適用を処理します。これまでに行った変更は、ページを表示するためのコードに影響を与えません。「編集」ページへの変更の適用は、if (action !=null) で始まるコード・ブロックで処理されます。

新しいポートレット・メッセージは、このコード・ブロックで格納する必要があります。また、ユーザーが何も変更しなかったため、既存のメッセージを取得する場合にも対処する必要があります。

```
if (action !=null)  
{  
    data.setPortletTitle(title);  
    //Put the new greeting.  
    data.putString("myportlet_greeting", greeting);  
    PortletRendererUtil.submitEditData(pReq, data);  
    return;  
}  
//Otherwise just render the form.  
title = data.getPortletTitle();  
//Get the old greeting.  
greeting = data.getString("myportlet_greeting");
```

これで、「編集」ページの更新が完了しました。

これらの変更内容は、「デフォルトの編集」ページにそのままコピーできます。「デフォルトの編集」ページは、ページ設計者またはポータル管理者がページで「**編集**」をクリックしてから「**デフォルトの編集**」をクリックするとコールされます。このページでは、ポートレットのこのインスタンスに対して、デフォルトのカスタマイズを設定します。JSP のコードが同じ場合でも、PDK フレームワークと OracleAS Portal は、表示モード（編集またはデフォルト編集）に応じて個別のカスタマイズを自動的に処理します。

「表示」ページでのカスタマイズの実装

ポートレットの共有画面モードでパーソナライズ・データにアクセスするには、「表示」ページに数行のコードを追加する必要があります。次の行が含まれます。

- import 文の追加
 - NameValuePersonalizationObject の宣言
 - カスタマイズ・データの取得
1. 「表示」ページを編集し、NameValuePersonalizationObject および PortletRendererUtil をインポートします。これらは、必要に応じて「編集」ページからコピーすることもできます。

```
<%@page contentType="text/html; charset=windows-1252"
import="oracle.portal.provider.v2.render.PortletRenderRequest"
import="oracle.portal.provider.v2.http.HttpCommonConstants"
import="oracle.portal.provider.v2.personalize.
NameValuePersonalizationObject"
import="oracle.portal.provider.v2.render.PortletRendererUtil"
%>
```

2. NameValuePersonalizationObject を宣言し、ポートレット・レンダリング・リクエストから編集データを取得します。この編集データは、ポートレットの「編集」ページからコピーすることもできます。

```
<%
PortletRenderRequest pReq = (PortletRenderRequest)
request.getAttribute(HttpCommonConstants.PORTLET_RENDER_REQUEST);
NameValuePersonalizationObject data = (NameValuePersonalizationObject)
PortletRendererUtil.getEditData(pReq);
%>
```

3. カスタマイズ・フレームワークから文字列情報を取得します。

```
String greeting = data.getString("myportlet_greeting");
```

4. ポートレットの共有画面モードでメッセージを表示する「表示」ページに、テキストを追加します。

```
<P>Hello <%= pReq.getUser().getName() %>.</P>
<P>This is the <b><i>show</i></b> render mode!</P>
<P>Greeting: <%= greeting %></P>
```

これで、ポートレットの「表示」ページの更新が完了しました。

XML プロバイダ定義内の設定情報

ポートレット・ウィザードでは、XML プロバイダ・ファイル (provider.xml) の PreferenceStore にアクセスするために必要なすべてのタグが生成されます。デフォルトでは、ウィザードはプロバイダ・レベルで FilePreferenceStore クラスを使用して設定を格納します。

```
<provider class="oracle.portal.provider.v2.DefaultProviderDefinition">
<session>>false</session>
<passAllUrlParams>>false</passAllUrlParams>
<preferenceStore class="oracle.portal.provider.v2.preference.FilePreferenceStore">
  <name>prefStore1</name>
  <useHashing>>true</useHashing>
</preferenceStore>
```

ポートレット・レベルでは、PrefStorePersonalizationManager を personalizationManager クラスとして使用し、NameValuePersonalizationObject をデータ・クラスとして使用するためのタグが追加されます。

```
<personalizationManager class="oracle.portal.provider.v2.personalize.
  PrefStorePersonalizationManager">
  <dataClass>oracle.portal.provider.v2.NewValuePersonalizationObject</dataClass>
</personalizationManager
```

FilePreferenceStore クラスを継続して使用する場合、XML プロバイダ定義に対する変更や更新は不要です。ただし、OracleAS Portal のグローバル環境 (たとえば、Oracle Application Server Containers for J2EE インスタンスのロード・バランスされた複数ノード・クラスターで実行中の場合) がある場合、または設定をデータベースに格納する場合は、クラスを FilePreferenceStore から DBPreferenceStore に変更できます。

注意： DBPreferenceStore の使用方法は、『Oracle Application Server Portal 構成ガイド』の Portal ツールおよび Web プロバイダの構成 (オプション) に関する項を参照してください。

ポートレットの表示

前の項で行ったカスタマイズの変更を表示するには、ポートレットをアプリケーション・サーバーまたは Oracle Application Server Containers for J2EE にデプロイし、ポートレットが含まれるページをリフレッシュする必要があります。ポートレットに NULL のメッセージが含まれていることを確認してください。ポートレット・タイトル・バーで「**カスタマイズ**」をクリックし、メッセージを更新します。ページに戻ると、変更内容が反映されています。

また、ページで「**編集**」をクリックしてから「**デフォルトの編集**」をクリックすると、「デフォルトの編集」をテストできます。すでにポートレットは変更されているため、パブリック・ユーザーまたは別のユーザーとしてページを表示するまで、変更内容は共有画面モードで表示されません。

パラメータの引渡しとイベントの送信

OracleAS Portal および PDK は、ポートレット開発者が再利用可能で複雑なポートレットを簡単に作成できるように、ページ・パラメータ、パブリック・ポートレット・パラメータ、プライベート・ポートレット・パラメータおよびイベントを提供しています。Oracle JDeveloper のポートレット・ウィザードは、パラメータとイベントを使用するように設定されたポートレットを作成します。この機能によって、開発者はビジネス・ロジックをポートレットに追加するだけで済み、`provider.xml` を変更する必要はありません。

パラメータおよびイベントの概要は、次の項を参照してください。

- 2-26 ページ「[パブリック・ポートレット・パラメータのサポート](#)」
- 2-27 ページ「[プライベート・ポートレット・パラメータのサポート](#)」
- 2-28 ページ「[イベントのサポート](#)」

前提

1. 5-35 ページの「[PDK-Java ポートレットの構築](#)」を読み終えて理解していること。
2. ウィザードで作成したポートレットがページに正常に追加されていること。

ポートレットへのパラメータの追加

5-35 ページの「[PDK-Java ポートレットの構築](#)」では、ウィザードを使用して、基本的なポートレットを作成し、`MyParam` というパラメータを指定しました。パラメータを作成していない場合は、ここで新しいポートレットを作成できます。ポートレットを作成するには、Oracle JDeveloper の「アプリケーション-ナビゲータ」で `provider.xml` を右クリックし、「[ポートレットの追加](#)」を選択して、「[PDK-Java ポートレットの構築](#)」で説明している手順を実行します。

ウィザードでは、コードまたはファイルを更新する必要がなくページ・パラメータを簡単にマップできるポートレットがデフォルトで作成されます。この項では、ウィザードで作成されたデフォルト・パラメータを使用します。

デフォルト・パラメータを使用するには、プロバイダを登録して、ポートレットをページに追加する必要があります。追加した後に、次のタスクを実行します。

- ページ・パラメータを作成します。
 - ページ・パラメータを Java ポートレットにマップします。
 - URL、またはこのページ・パラメータを渡す別のポートレットにパラメータ値を入力します。
1. ページ・プロパティの「[パラメータ](#)」タブに移動します。パラメータは、デフォルトで有効になっています。無効な場合は、続行する前に有効にしてください。
 2. `MyParameter` というページ・パラメータを作成し、デフォルト値は `My Default Value` と入力します。

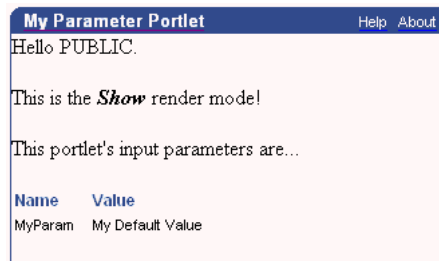
3. Java ポートレットを開き、作成したページ・パラメータをポートレット・パラメータにマップします。

```
My Portlet Parameter = Page Parameter MyParameter
```

4. ページに戻ります。ポートレットに My Default Value という値が表示されていることに注意してください。
5. ページを表示し、URL の最後にパラメータと値を入力します。

```
&MyParameter=This%20portlet%20works
```

図 5-39 パラメータ・ポートレット



OmniPortlet のシンプル・パラメータ・フォームなど、パラメータを渡すことができるポートレットがある場合は、「イベント」タブを使用して、そのポートレットから Java ポートレットにパラメータを簡単にマップできます。

ここで、ウィザードで生成されたコードおよびタグを確認すると、Java ポートレットでパラメータを有効にするには、数行のコードが必要であることがわかります。

provider.xml を確認します。ウィザードによって、inputParameter というタグ・グループが追加されています。これには、ポートレットがリスニングするパラメータの名前が含まれます。

```
<inputParameter class="oracle.portal.provider.v2.DefaultParameterDefinition">
  <name>MyParam</name>
  <displayName>My Portlet Parameter</displayName>
</inputParameter>
```

また、ウィザードによって、このパラメータを受信してパラメータ名とその値を表示する「表示」ページを表すコードが JSP で生成されています。

```
<%
ParameterDefinition params[] =
    pReq.GetPortletDefinition().getInputParameters();
%>

<p>This portlets input parameters are ...</p>
<table align="left" width="50%"><tr><td><span class="PortletHeading1">Value
</span></td></tr>
<%
String name = null;
String value = null;
String[] values = null;
for (int i = 0; i < params.length; i++)
{
    name = params[i].getName();
    values = pReq.getParameterValues(name);
    if (values != null)
    {
        StringBuffer temp = new StringBuffer();
        for (int j = 0; j < values.length; j++)
        {
            temp.append(values[j]);
            if (j + 1 != values.length)
            {
                temp.append(", ");
            }
        }
        value = temp.toString();
    }
    else
    {
        value = "No values submitted yet.";
    }
}
%>
<tr>
    <td><span class="PortletText2" <%= name %></span></td>
    <td><span class="PortletText2" <%= value %></span></td>
</tr>
<%
}
%>
</table>
```


イベントの送信

前の項では、パラメータを受信するポートレットを作成しました。次に、パラメータおよびイベントを、同じページまたは別のページの他のポートレットに渡すポートレットを作成します。OmniPortlet のシンプル・パラメータ・フォームのように、パラメータを他のポートレットに渡すシンプル・フォームを作成するための簡単で宣言的なインタフェースを提供するポートレットもあります。ただし、渡されるイベントやポートレットの外観を完全に制御する必要がある場合は、イベントを Java ポートレットに追加できます。

ポートレット・ウィザードでは、パラメータを他のポートレットに渡すために必要なすべてのコードが作成されるわけではありません。ウィザードでは `provider.xml` のタグが更新され、開発者は必要なビジネス・ロジックを JSP コードに追加する必要があります。イベントを使用するポートレットを作成するには、次のタスクを実行します。

- ポートレット・ウィザードを使用して、新しいポートレットを作成します。
- コードを JSP ページに追加します。
- このポートレットのパラメータを、5-66 ページの「[ポートレットへのパラメータの追加](#)」で作成したポートレットにマップします。

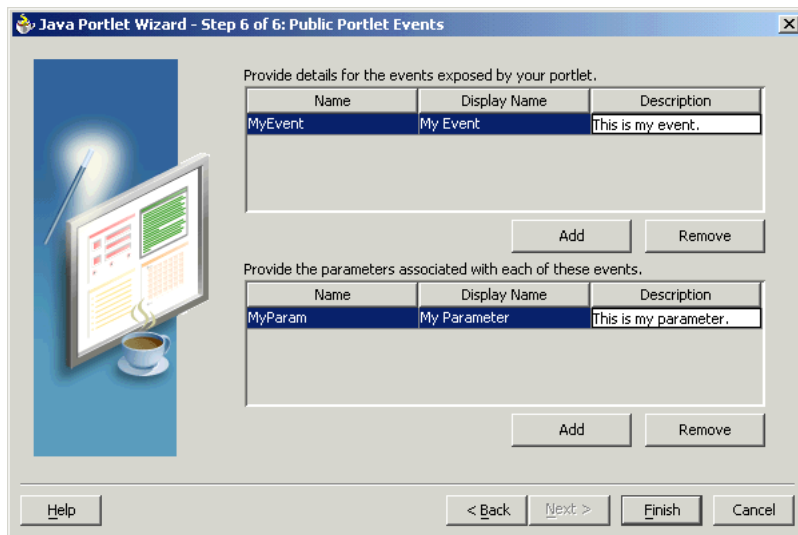
イベント・ポートレットの作成 イベント・ポートレットを作成する手順は、次のとおりです。

1. ポートレット・ウィザードを起動して、同じプロバイダに `MyEventsPortlet` という新しいポートレットを作成します。通常どおりにウィザードを進めます。ウィザードのステップ 5 で、パラメータを作成します。ウィザードのステップ 6 で、[表 5-8](#) に示す情報を入力します。

表 5-8 イベント

イベント領域	名前	表示名	説明
公開されるイベント	MyEvent	My Event	これが my event です。
関連付けられたパラメータ	MyParam	My Parameter	これが my parameter です。

図 5-40 ポートレット・ウィザードの「パブリック・ポートレット・イベント」ページ



ウィザードによって、次のコードが `provider.xml` に生成されます。

注意： 次の例では、入力パラメータとイベント・パラメータの名前がいずれも `MyParam` になっていることに注意してください。これらは、名前が同じでも異なるパラメータです。

```
<showDetails>false</showDetails>
<inputParameter class="oracle.portal.provider.v2.
  DefaultParameterDefinition">
  <name>MyParam</name>
  <displayName>My Parameter</displayName>
</inputParameter>
<event class="oracle.portal.provider.v2.DefaultEventDefinition">
  <name>MyEvent</name>
  <displayName>My Event</displayName>
  <parameter class="oracle.portal.provider.v2.DefaultParameterDefinition">
    <name>MyParam</name>
    <displayName>My Parameter</displayName>
  </parameter>
</event>
<renderer class="oracle.portal.provider.v2.render.RenderManager">
```

2. 必要なクラスをインポートします。
 - `oracle.portal.provider.v2.event.EventUtils`
 - `oracle.portal.utils.NameValue`
 - `oracle.portal.provider.v2.url.UrlUtils`
3. パラメータ値を別のポートレットに渡すリンクを追加します。次のサンプル・コードに示すように、前のポートレットと同じページ・パラメータを受信しますが、それに加えてイベントを渡すリンクも作成します。

```
<%@page contentType="text/html; charset=windows-1252"
import="oracle.portal.provider.v2.render.PortletRenderRequest"
import="oracle.portal.provider.v2.http.HttpCommonConstants"
import="oracle.portal.provider.v2.ParameterDefinition"
import="oracle.portal.provider.v2.event.EventUtils"
import="oracle.portal.utils.NameValue"
import="oracle.portal.provider.v2.url.UrlUtils"
%>
<%
PortletRenderRequest pReq = (PortletRenderRequest)
request.getAttribute(HttpCommonConstants.PORTLET_RENDER_REQUEST);
%>
<%
    NameValue[] parameters = new NameValue[2];
    parameters[0] = new NameValue( EventUtils.eventName("MyEvent"), "");
    parameters[1] = new
NameValue(EventUtils.eventParameter("MyParam"), pReq.getParameter
("MyParam"));
%>
<span class="portletText1"><br>
<a href="<%= UrlUtils.constructLink
(pReq, pReq.getRenderContext().getEventURL(), parameters , true, true)%>">
The value of the stock is <%= pReq.getParameter("MyParam") %>
</a>
<br><br></span>
```

注意： このサンプル・コードでは、NULL 値を処理していません。ポートレットを最初にページに追加する際にエラーが発生する場合がありますが、ポートレットをページ・パラメータにマップすると正常に機能します。

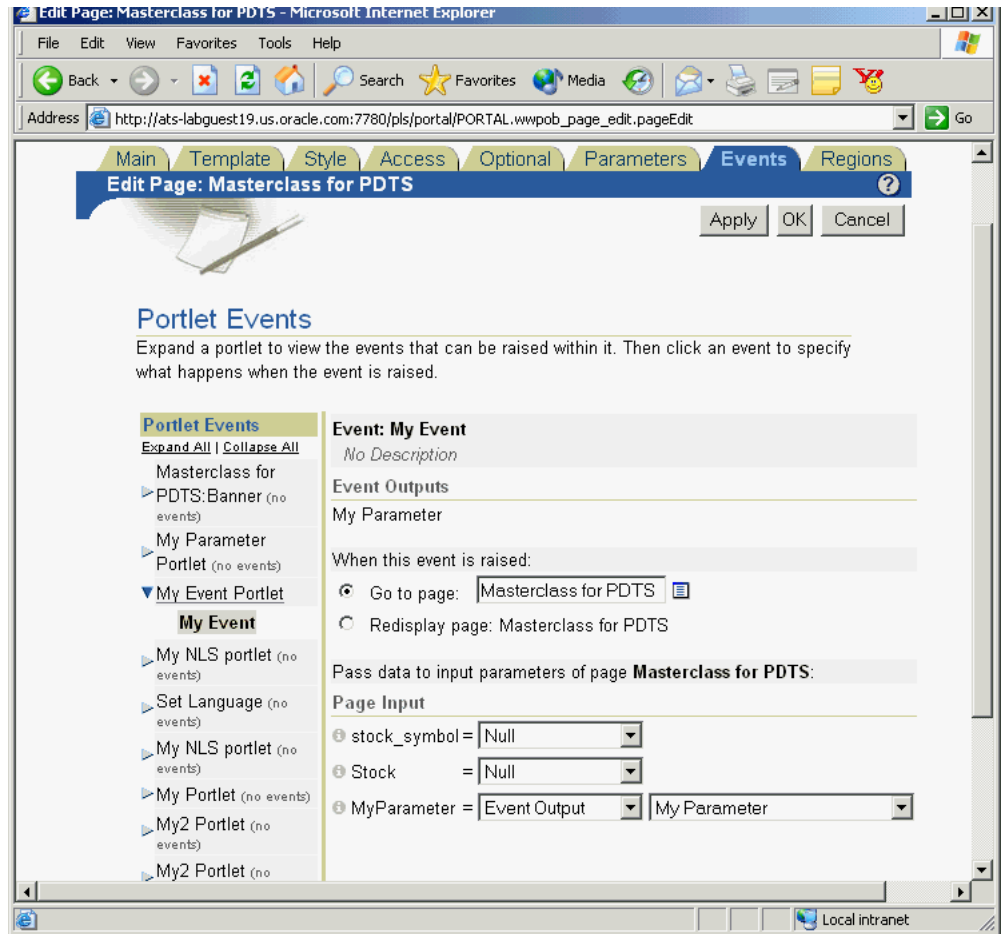
4. ポートレットを、前のポートレット（パラメータ・ポートレット）以外の別のページ（同じページ・グループ内）に追加します。ポートレットを開き、前のポートレットと同じパラメータを受信するようにマップします。

```
My Parameter = Page Parameter MyParameter
```

5. 「パラメータ」タブで変更内容を適用し、「イベント」タブに移動します。イベント・ポートレットを開き、イベントを選択します。「ページに移動」を選択し、イベントを渡すページを検索します。パラメータ・ポートレットが存在するページを選択します。イベントをページ・パラメータ MyParameter として渡すように、このポートレットを構成します。

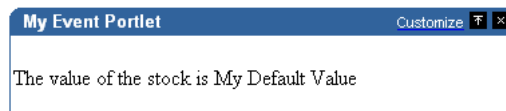
```
MyParameter = Event Output MyParameter
```

図 5-41 「ページの編集」のポートレット・イベント



6. 「OK」をクリックすると、ページが表示されます。イベント・ポートレットには、ページから受信した値を表示するリンクが表示されます。

図 5-42 パラメータ変更前の My Event ポートレット



7. パラメータ値を URL に追加すると、ポートレットのリンクにその値が表示されます。
`&MyParameter=20`
8. リンクをクリックすると、その値がページのパラメータ・ポートレットに渡されます。

図 5-43 パラメータ変更後の My Event ポートレット



セッション情報へのアクセス

ユーザーがポータル・ページにアクセスすると、OracleAS Portal は、パブリックな未認証のセッションを起動して Cookie を保持し、そのセッションに関する情報を複数リクエストにわたって追跡します。ユーザーが OracleAS Portal にログインすると、このセッションはログイン・ユーザーの認証済セッションとなります。このポータル・セッションは、次の場合に終了します。

- ブラウザ・セッションが終了したとき（つまり、ユーザーがすべてのブラウザ・ウィンドウを閉じたとき）
- ユーザーが明示的にログアウトしたとき
- ユーザーのアイドル時間が設定時間を超えたためにセッションがタイムアウトしたとき

ポータル・セッションは、ユーザーがポータルに初めてアクセスしてから、この 3 つの場合のいずれかによってセッションが終了するまで存続します。

セッション・ストアを使用すると、ポータル・セッションの間存続する情報を保存および取得できます。この情報が有効で使用可能なのは、そのセッションの存続期間中のみです。セッション・ストアに格納するのは、一時情報のみです。アプリケーション開発者はこのセッション・ストアを使用して、現行のユーザー・セッションに関する情報を格納できます。セッション・ストアのデータは複数のポートレット間で共有できます。

格納する情報を複数のセッションにわたって存続させる必要がある場合は、セッション・ストアのかわりにプリファレンス・ストアに格納できます。次に、一般的なセッション・ストアの適用例を示します。

- ロードや演算にコストがかかるデータ（たとえば、検索結果）をキャッシュする場合。
- ポートレットの現在の状態をキャッシュする場合。この状態の例には、ポートレットに表示された検索結果の現在の範囲（ページ）やユーザーが実行したイベントの順序などがあります。

セッション記憶域を実装する前には、パフォーマンス・コストを慎重に検討する必要があります。ポートレットとプロバイダはリモートであるため、セッション・ストアに情報を作成して保持する操作は、少ない情報量であっても比較的成本がかかる可能性があります。このため、多数のユーザーが頻繁にアクセスするパブリック・ページについては、セッション記憶域の実装を避けた方がよい場合があります。

また、Web プロバイダでセッション・ストアを使用する場合は、メモリー内の状態情報を追跡する必要があるステートフルなアプリケーションを作成します。同様に、プリファレンス・ストアのファイル・システム実装を使用する場合も、ステートフルなアプリケーションを作成します。

スケーラビリティに重要な関心がある場合、ステートフルなアプリケーションは問題発生の原因となる可能性があります。ステートフルなアプリケーションは、使用している OracleAS Portal 構成のロード・バランスとフェイルオーバー・メカニズムに影響を与えることがあります。これは、同じ OracleAS Portal インスタンスにアクセスする複数の中間層をデプロイしても、状態を追跡するためには面倒なルーティング（同じノードが後続のリクエストを同じセッションで処理する）を実装する必要があるためです。このルーティングによって、ノードがクラッシュした場合に不均衡なロード・バランスやセッション・データの消失が発生し、フェイルオーバーに影響を与えることがあります。多くの開発者がステートレスなアプリケーションの構築を選ぶ理由の1つはここにあります。ただし、スケーラビリティが問題でない場合、ステートフルなアプリケーションによる問題は発生しません。

これまでの各項では、PDK フレームワークを使用して様々なレンダリング・モードでポートレットのコンテンツをレンダリングする方法、カスタマイズなどの機能を実装する方法、およびパラメータとイベントについて説明してきました。この項では、ポートレット用にセッション記憶域を実装する方法を説明します。

この項では、共有画面モードでポートレットをレンダリングした回数をカウントするために、セッション記憶域を使用します。

前提

1. 5-35 ページの「PDK-Java ポートレットの構築」を読み終えて理解していること。
2. ウィザードで作成したポートレットがページに正常に追加されていること。

セッション記憶域の実装

PDK フレームワークでは、セッションを `ProviderSession` オブジェクトを使用して表します。このオブジェクトはプロバイダ・インスタンスの `initSession` メソッドへのコールの際に確立されます。このオブジェクトは、`ProviderUser` と関連付けられています。OracleAS Portal からのリクエスト間でデータを永続的に保持するには、`ProviderSession` オブジェクト上の `setAttribute` メソッドを使用してセッション・オブジェクトにデータを書き込む必要があります。このメソッドは `java.lang.Object` を `java.lang.String` にマップし、そのマッピングをセッション・オブジェクト内に格納します。`String` は、その後、セッションがまだ有効である場合、後続のリクエスト中に `Object` を取得するために使用できます。

次のような場合は、プロバイダ・セッションが無効になることがあります。

- セッションがタイムアウトした場合
- ProviderSession 上の invalidate メソッドがコールされた場合
- サブレット・コンテナを実行中の JVM プロセスが終了した場合

同じ ProviderInstance に含まれるすべてのポートレットは、特定の ProviderUser に対して同じセッションを共有します。したがって、特定のポートレット・インスタンスに対して一意のデータは、セッション内の一意の String にマップする必要があります。これは、PortletRendererUtil クラス内の portletParameter メソッドを使用して行います。このメソッドは、そのインスタンス用に生成された識別子を先頭に付けることで、提供された String パラメータまたは属性名を PortletInstance に対して一意にします。戻されたインスタンス固有の名前を使用すると、セッションにポートレット・インスタンス・データを書き込むことができます。PDK フレームワークのクラスの詳細は、JavaDoc を参照してください。

セッション記憶域を実装するには、次のタスクを実行する必要があります。

- ProviderSession、PortletRendererUtil および HttpPortletRendererUtil をインポートします。
- プロバイダ・セッションを取得します。
- Java ポートレット内からセッションにアクセスして、セッションの読取りと書き込みを行います。
- provider.xml で session を true に設定します。
- セッション記憶域のプロバイダを登録し、「ログイン周期」を設定します。

次に、現行セッションでポートレットがレンダリングされた回数を表示するセッション・カウントをポートレットに追加する手順を説明します。

1. ウィザードを使用してポートレットを作成した後は、Oracle JDeveloper の「表示」ページの JSP を編集できます。次のクラスをインポートする必要があります。

```
<%@page contentType="text/html; charset=windows-1252"
import="oracle.portal.provider.v2.render.PortletRenderRequest"
import="oracle.portal.provider.v2.http.HttpCommonConstants"
import="oracle.portal.provider.v2.ProviderSession"
import="oracle.portal.provider.v2.render.PortletRendererUtil"
import="oracle.portal.provider.v2.render.http.HttpPortletRendererUtil"
%>
```

2. 最初に有効なセッションをチェックし、次にカウントを増分して表示するコードを挿入します。セッションが有効で、以前に格納した値が存在する場合は、その値を表示し、カウントを増分して新しい値を格納します。セッションは有効だが以前に格納した値がない場合は、カウントを 1 に初期化し、その値を表示して格納します。また、このポートレットの一意の文字列キーを取得し、その文字列キーを配列で使用してセッションを

カウントできます。セッション情報を受信していない場合は、ユーザーに対して、再度ログインが必要であることを示す情報を提供できます。

```
<%
PortletRenderRequest pReq = (PortletRenderRequest)
request.getAttribute(HttpCommonConstants.PORTLET_RENDER_REQUEST);
ProviderSession pSession = pReq.getSession();
if (pSession != null)
{
String key = PortletRenderUtil.portletParameter(pReq, "count");
Integer i = (Integer)pSession.getAttribute(key);
if (i == null)
{
i = new Integer(0);
}
i = new Integer(i.intValue()+1);
pSession.setAttribute(key, i);
}%>

<p>Render count in this session: <%=i%> </p>

<%
}
else
{
}%>

<p>The session has become invalid</p>
<br>
Please log out and log in again.
<%
}
}%>
```

3. デフォルトでは、ウィザードで `provider.xml` の `session` を `true` に設定することはありません。プロバイダがポータルからセッション情報を受信するためには、このフラグを更新する必要があります。このタグを `true` に設定する必要があるのは、プロバイダまたはポートレットでセッション情報を使用する場合のみです。このフラグを `true` に設定すると、プロバイダ・コールに余分なロードが追加されます。

```
<provider class="oracle.portal.provider.v2.DefaultProviderDefinition">
<session>true</session>
```

4. セッションをサポートするためにプロバイダを登録し、その「ログイン周期」を「セッションごとに1回」に設定します。

ポートレットの表示

Java ポートレットをページに追加していない場合は、ここで追加します。次のタスクを実行します。

- プロバイダの「ログイン周期」の値を「ユーザー・セッションごとに1回」に設定します。
- プロバイダをリフレッシュして新しい変更内容を反映させます。
- セッションがすでに無効の場合は、再度ログインします。

ポートレットのセキュリティの実装

これまでの各項では、PDK フレームワークを使用して様々な表示モードでポートレットのコンテンツをレンダリングする方法、カスタマイズなどの機能を実装する方法、パラメータとイベント、およびセッション記憶域について説明してきました。この項では、Java ポートレットで使用できるセキュリティ・サービスについて説明します。この項で参照されている PDK クラスの詳細は、[JavaDoc](#) を参照してください。

前提

1. 5-35 ページの「[PDK-Java ポートレットの構築](#)」を読み終えて理解していること。
2. ウィザードで作成したポートレットがページに正常に追加されていること。

ポートレットのセキュリティ機能

この項では、ポートレット・プロバイダの保護に使用できる様々なセキュリティ機能を説明します。

認証 ユーザーが初めて OracleAS Portal インスタンスにログインするときは、ID を確認してアクセス権限を取得するためにパスワードを入力する必要があります。この認証は、OracleAS Single Sign-On サーバーで実行されます。詳細は、5-80 ページの「[シングル・サインオン](#)」を参照してください。

認可 認可によって、特定のユーザーがポートレットの表示または対話を行えるかどうか判断されます。OracleAS Portal には、次の 2 種類の認可チェックが用意されています。

- **Portal のアクセス制御リスト (ACL)** : OracleAS Single Sign-On によって認証されると、OracleAS Portal では ACL を使用して、フォルダやポートレットなどのポータル・オブジェクトに対して操作を実行するユーザー権限を決定します。ユーザーの操作の範囲は、単なるオブジェクトの表示から、そのオブジェクトに対する管理機能の実行にまで及びます。特定の権限が付与されているグループに属していない場合、OracleAS Portal では、その権限に関連付けられている操作を実行できません。詳細は、5-82 ページの「[OracleAS Portal アクセス制御リスト \(ACL\)](#)」を参照してください。

- **プログラムによるポートレットのセキュリティ**: プログラムによって独自のセキュリティ・マネージャを実装することも可能です。詳細は、5-83 ページの「[ポートレットのセキュリティ・マネージャ](#)」を参照してください。

通信セキュリティ これまでに説明したユーザー認証と認可では、プロバイダが受信したメッセージの信頼性までは確認されません。プロバイダを完全に保護するには、OracleAS Portal と Web プロバイダ間の通信を保護する必要があります（この方法は、OracleAS Portal データベースの内部で実行するデータベース・プロバイダには適用されません）。通信が保護されていないと、誰かが OracleAS Portal インスタンスを装い、Web プロバイダをだまして機密情報を返信させる恐れがあります。通信セキュリティには、次の3種類があります。

- **OracleAS Portal サーバー認証**では、プロバイダへのアクセスを少数の認証されているマシンに制限します。この方法では、受信した HTTP メッセージの IP アドレスまたはホスト名をトラステッド・ホストのリストと比較します。リストに IP アドレスまたはホスト名がある場合、メッセージはプロバイダに渡されます。リストにないと、そのメッセージはプロバイダに達する前に拒否されます。詳細は、5-85 ページの「[OracleAS Portal のサーバー・セキュリティ](#)」を参照してください。
- **メッセージ認証**では、共有キーに基づいてチェックサムをプロバイダのメッセージに追加します。プロバイダがメッセージを受信すると、予想されるチェックサムの値を計算し、その値と実際に受信した値とを比較することによって、メッセージの信頼性が確認されます。それらの値が同じ場合、メッセージは受け入れられます。値が異なっている場合、メッセージは拒否され、その後の処理も行われません。送信中にメッセージが不法に記録されて後で再送信される可能性を低くするために、チェックサムにはタイムスタンプが含まれています。詳細は、5-86 ページの「[メッセージ認証](#)」を参照してください。
- **メッセージの暗号化**は、OracleAS Portal からプロバイダへの通信に HTTPS プロトコルを使用することによって行われます。メッセージは高度に暗号化され、その中でデータが保護されます。暗号化によって高水準のセキュリティを確保できますが、メッセージごとに追加処理が必要になるため、パフォーマンスが低下します。詳細は、5-87 ページの「[HTTPS 通信](#)」を参照してください。

通信セキュリティの詳細は、『Oracle Application Server Portal 構成ガイド』を参照してください。

シングル・サインオン

ポートレットは、アプリケーションへのウィンドウの役目を果たします。ポートレットにはサマリー情報が表示され、アプリケーションのすべての機能にアクセスする方法が提供されます。ポートレットは、アプリケーションの機能をポータルで直接公開したり、タスクを実行するためにアプリケーション自体にアクセスできる深いリンクを提供します。

シングル・サインオンの詳細は、『Oracle Application Server Portal 構成ガイド』を参照してください。

アプリケーションでは、ポートレットを介してアプリケーションにアクセスするユーザーを認証する必要があります。次のアプリケーション認証方法があります。

- **パートナ・アプリケーション**。この認証方法では、アプリケーションのユーザーは、OracleAS Portal の認証ユーザーと同じです。
- **外部アプリケーション**。この認証方法では、OracleAS Portal ユーザーとアプリケーションのユーザーは異なりますが、アプリケーションのユーザー名とパスワードは OracleAS Portal ユーザーが管理します。
- **アプリケーションによる認証がない場合**。この場合、プロバイダと OracleAS Portal 間の通信はまったく保護されません。

パートナ・アプリケーション パートナ・アプリケーションは、認証のために OracleAS Portal と同じ OracleAS Single Sign-On を共有するアプリケーションです。したがって、ユーザーが OracleAS Portal にすでにログインしているときは、再度ログインしなくてもその ID がパートナ・アプリケーションにアサートされます。

パートナ・アプリケーションは、OracleAS Single Sign-On と密接に統合されています。ユーザーがパートナ・アプリケーションへのアクセスを試みると、パートナ・アプリケーションはそのユーザーの認証を OracleAS Single Sign-On に委任します。有効なユーザー名とパスワードを入力したユーザーが、あるパートナ・アプリケーションに対して一度認証されると、そのユーザーは、同じ OracleAS Single Sign-On インスタンスを共有している他のパートナ・アプリケーションにアクセスするときにユーザー名やパスワードを入力する必要がありません。OracleAS Single Sign-On では、ユーザーが正常に認証されたことを確認し、認証が成功したことを他のパートナ・アプリケーションに示します。

パートナ・アプリケーションの利点は次のとおりです。

- OracleAS Portal と OracleAS Single Sign-On Server を最も密接に統合できます。
- ユーザーはシングル・サインオンを最も有効に利用できます。
- OracleAS Portal とプロバイダ間でのユーザー名やパスワードの転送がないため、最も安全な形式で統合できます。

パートナ・アプリケーションの不利な点は次のとおりです。

- アプリケーションのユーザー・コミュニティが OracleAS Portal のユーザー・コミュニティのサブセットである可能性があっても、アプリケーションは OracleAS Portal と同じユーザー・リポジトリを共有する必要があります。この点は考慮する必要がありますが、そのアプリケーションを公開するポータル・ページへのアクセスをアプリケーションのユーザー・コミュニティに制限できるため、この問題は重大ではありません。
- アプリケーションを1つ以上の OracleAS Single Sign-On インスタンスと密接に統合できるのは、それらのインスタンスが同じユーザー・リポジトリを共有している場合のみです。
- アプリケーションは、認証を OracleAS Single Sign-On に委任するように作成する必要があります。
- アプリケーションのソース・コードへのアクセス権限が必要です。

外部アプリケーション 外部アプリケーションでは、OracleAS Portal とは異なる認証サーバーを使用します。このアプリケーションでは、OracleAS Portal が使用する Single Sign-On Server の異なるインスタンスまたは他の認証方法を使用できます。ただし、Single Sign-On Server では、そのユーザーについて、外部アプリケーションのユーザー名とパスワードが格納されます。これは、ユーザーが Oracle Portal にすでにログインしている場合、そのユーザーはユーザー名やパスワードを入力しなくても外部アプリケーションにログインできることを意味します。

ユーザーの認証を管理しているアプリケーションは、管理者がそのアプリケーションを外部アプリケーションとして登録している場合、OracleAS Single Sign-On と柔軟に統合できます。以前に OracleAS Single Sign-On によって認証されたユーザーが初めて外部アプリケーションにアクセスすると、OracleAS Single Sign-On では、その外部アプリケーションを使用してユーザーの認証を試みます。認証プロセスでは、そのアプリケーションの登録情報と、ユーザーのユーザー名とパスワードとを結合する HTTP リクエストが送信されます。ユーザーがまだその外部アプリケーションのユーザー名とパスワードを登録していない場合、OracleAS Single Sign-On では、認証要求を行う前に、必要な情報の入力をユーザーに要求します。ユーザーが外部アプリケーションのユーザー名とパスワードを入力すると、OracleAS Single Sign-On では新しいユーザー名とパスワードがユーザーの OracleAS Portal のユーザー名にマップされ、格納されます。格納された情報は、次回ユーザーが外部アプリケーションによる認証が必要なときに使用されます。

外部アプリケーションの**利点**は次のとおりです。

- 多数のポータルとの統合が可能です。ただし、優先するポータルがある場合は、アプリケーションをその優先ポータルのパートナ・アプリケーションとして統合し、他のポータルの外部アプリケーションとして統合できます。
- ユーザーはシングル・サインオンを利用できます。ただし、ユーザーは異なるユーザー名やパスワードを保持する必要があります。さらに、外部アプリケーションのユーザー名のマッピングも保持する必要があります。

- ユーザー・リポジトリおよび OracleAS Single Sign-On に依存せずに、複数のポータルと統合できます。
- アプリケーションのソース・コードへのアクセス権限が不要です。

外部アプリケーションの**不利な点**は次のとおりです。

- ポータルと同じユーザー・リポジトリを共有しません。このため、エンド・ユーザーによるユーザー情報の管理が必要です。
- ユーザー名とパスワードは、SSL を実装しないかぎり、プレーン・テキストでプロバイダに送信されます。

アプリケーションによる認証がない場合 プロバイダはリクエストを送信する OracleAS Portal を完全に信頼します。プロバイダは、ユーザーがログインしているかどうかの確認やポータル・ユーザー名の確認はできますが、アプリケーションによるユーザーの認証は行われていません。

アプリケーションによる認証がない場合の**利点**は次のとおりです。

- 最も簡単な形式で統合し、最も短時間で実装できます。

アプリケーションによる認証が場合の**不利な点**は次のとおりです。

- 最も低いセキュリティ・レベルになります。
- OracleAS Portal との統合が最も弱くなります。

OracleAS Portal アクセス制御リスト (ACL)

OracleAS Portal インスタンスにログインすると、OracleAS Single Sign-On インスタンスによって認証が行われます。ID を確認した後、OracleAS Portal は、ACL を使用して、特定のポートレットへのアクセスが認可されているかどうかを確認し、そのポートレットをポートレット・リポジトリからページに追加します。

OracleAS Portal ACL は、次のセキュリティ特性に従って使用されます。

- **権限**は、権限が付与されているオブジェクトに対して実行できる操作を定義します。権限には、管理や実行などの操作が含まれます。
- OracleAS Portal の**ユーザーとその権限**は、ビルダーの「管理」タブで付与されます。
- OracleAS Portal の**ユーザー・グループ**は、OracleAS Portal ビルダーの「管理」タブで管理されます。グループ内のメンバーシップとそのグループに付与されている権限は、すべてこのタブで定義および保持されます。ユーザー・グループに付与された権限は、そのグループのすべてのユーザーによって継承されます。

- **プロバイダの権限**は、プロバイダとそのすべてのポートレットに適用されます。プロバイダの ACL は、OracleAS Portal ナビゲータの「プロバイダ」タブで管理されます。
- **ポートレットの権限**によって、ポートレットのプロバイダに設定された権限を上書きできます。ポートレットの ACL は、OracleAS Portal ナビゲータの「プロバイダ」タブで管理されます。プロバイダについて「開く」をクリックすると、そのプロバイダのポートレットを管理するページが表示されます。

OracleAS Portal の各オブジェクト、ユーザーおよびユーザー・グループに適用できる権限の詳細は、『Oracle Application Server Portal 構成ガイド』を参照してください。

ACL の利点は次のとおりです。

- ACL は、OracleAS Portal オブジェクトを保護するための簡単かつ強力なメカニズムを備えています。
- ユーザー・グループのメンバーシップを集中管理することによって、各オブジェクトに関連付けられている ACL の変更が不要になるため、ACL の管理が簡略化されます。

ACL の不利な点は次のとおりです。

- ACL は、プロバイダまたはポートレットのレベルで適用されます。ポートレットを配置するページに応じてポートレットのセキュリティ・ルールを変更することはできません。

ポートレットのセキュリティ・マネージャ

ポートレットのセキュリティ・マネージャはプロバイダ内部に実装され、特定のユーザーがポートレットのインスタンスを表示できることを確認します。ユーザーがポートレット・インスタンスを含むページを表示すると、セキュリティ・マネージャによって、そのユーザーにポートレットを表示する適切な権限があるかどうか判断されます。プロバイダでアクセス制御メソッドを実装することで、適切な権限のないユーザーはポートレットからコンテンツを取得できない（つまり、ポートレットを表示しない）ように制限します。ユーザーの詳細や設定項目など指定の特性が認可ロジックを通過した場合のみ、そのユーザーはコンテンツを取得できます。プロバイダでポートレットのセキュリティ・メソッドを実装しない場合は、偽名や認証されていない名前であっても、すべてのユーザー名が通過する可能性があります。

プロバイダでは、次の 2 つのポートレットのセキュリティ・メソッドを実装できます。

- ポートレットのリストの取得
- ポートレットのアクセス可能性の確認

ポートレットは、OracleAS Portal のユーザー権限、およびユーザーがメンバーとして属しているグループにアクセスできます。セキュリティ・メソッドで使用できる情報は次のとおりです。

- ユーザーのデフォルト・グループ
- ユーザーまたはグループの権限

- すべてのグループで利用できる最高のユーザー権限
- ユーザーがアクセスできるオブジェクト（データベース・プロバイダのみ）

AuthLevelSecurityManager は、認可レベルに関する次の情報にアクセスできます。

- 強い認証

ユーザーは、現行の OracleAS Portal セッションで OracleAS Single Sign-On によって認証されています。つまり、ユーザーは有効なユーザー名とパスワードを使用してログインし、そのセッションのコンテキスト内でポートレットをリクエストしました。

- 弱い認証

以前に強い認証を受けているユーザーは、OracleAS Portal セッションがアクティブではないページに戻り、そのページを表示します。永続的な Cookie（ユーザーのブラウザで保持）には、以前のセッションでユーザーが有効なユーザー名とパスワードでログインしたことが示されます。

- パブリックまたは認証なし

ユーザーは現行の OracleAS Portal セッションのコンテキスト内でログインしたことがなく、以前にログインしたことを示す永続的な Cookie もありません。

これらのセキュリティ・サービスを Java ポートレットに組み込むには、provider.xml を更新して、セキュリティ・レベルを strong、weak または public に設定します。次の XML を provider.xml の </portlet> タグの直前に挿入してください。

```
<securityManager class="oracle.portal.provider.v2.security.AuthLevelSecurityManager">
  <securityLevel>strong</securityLevel>
</securityManager>
```

provider.xml を変更した後は、プロバイダをリフレッシュします。

セキュリティ・メソッドの**利点**は次のとおりです。

- ポートレットでは、認可レベルに従って異なる出力を作成できます。

セキュリティ・メソッドの**不利な点**は次のとおりです。

- セキュリティ・マネージャのほとんどの実装では、認可レベルや受信メッセージにあるユーザー固有の他の要素が使用されます。このような確認は、OracleAS Portal インスタンスを装ったエンティティによって無視されることがあります。

ポートレットの表示

Java ポートレットに追加したセキュリティ・マネージャの動作を示す手順は、次のとおりです。

1. ページ作成とページへのポートレット追加を実行できる権限で OracleAS Portal インスタンスにログインしていることを確認します。
2. PUBLIC に対して表示可能な新しいポータル・ページを作成します。

3. Java ポートレットをページに追加します。
4. 新しいポータル・ページへの直接 URL を書き留めておきます。
5. 「ログアウト」リンクをクリックして、ポータル・インスタンスからログアウトします。
6. ブラウザのアドレス・バーに手順 4 で書き留めた URL を入力して、ポータル・ページに直接アクセスします。

手順 2 で作成したページが表示されますが、手順 3 で作成したポートレットは表示されません。ページにポートレットを追加したときはログインしていたため、強い認証を受けています。PDK ランタイムはこれを検出し、ポートレットの追加を許可しました。ログアウトしてページを表示したときには、強い認証がすでに失われていたため、PDK フレームワークはポートレットのコンテンツのレンダリングを許可しませんでした。

再度ログインしてページを表示すると、ポートレットの存在を確認できます。

独自のセキュリティ・マネージャの実装 PDK に付属の実装では提供されない特別なセキュリティ対策がポートレットに必要な場合は、独自のカスタムの `PortletSecurityManager` コントローラ・クラスを用意する必要があります。そのためには、`oracle.portal.provider.v2.security.PortletSecurityManager` クラスを拡張し、インタフェースで指定されている 2 つのメソッドに対して実装を提供します。次に、XML プロバイダ定義で `securityManager` コントローラ要素のクラス属性を新しいクラス名に置き換え、子要素を適切に構成します。

OracleAS Portal のサーバー・セキュリティ

プロバイダへの認可されていないアクセスを回避する 1 つの方法は、プロバイダへのアクセスを既知のクライアント・マシンに、サーバー・レベルで制限することです。この方法は、サービスの拒否攻撃に対する防御にある程度役立ちます。

このため、Oracle Application Server では、ホスト名または IP アドレスに基づいて `httpd.conf` ファイルのディレクティブを許可または拒否することで、クライアント・マシンへのアクセスを制御します。識別子としてホスト名が使用されている場合、サーバーはドメイン・ネーム・サーバー (DNS) でホスト名を検索する必要があり、各リクエストの処理で余分なオーバーヘッドが発生します。この問題は IP アドレスを使用することで回避できますが、リモート・クライアントの IP アドレスが警告なしに変更される可能性があります。

サーバー・セキュリティの利点は次のとおりです。

- プロバイダへのアクセスをトラステッド・ホストのみに制限できます。
- 構成を簡略化できます。

サーバー・セキュリティの不利な点は次のとおりです。

- OracleAS Web Cache には IP アドレスのチェック機能がありません。プロバイダの前に OracleAS Web Cache がある場合は、すべてのホストのクライアントが OracleAS Web Cache に表示リクエストを送信するのを防止できません。

- アクセスを特定の IP アドレスとホスト名に制限しても、偽の IP アドレスとホスト名が含まれたプロバイダにメッセージを送信することで回避される可能性があります。この方法の場合、リターン・メッセージは IP アドレスがコピーされたマシンに送信されますが、その後も問題が発生する可能性があるため、効果的に実行するのは困難です。

このトピックの詳細は、『Oracle Application Server Portal 構成ガイド』を参照してください。

メッセージ認証

PDK-Java ではメッセージ認証がサポートされているため、アクセスを指定されたプロバイダ・インスタンスまたはプロバイダ・インスタンスのグループに制限できます。プロバイダは、OracleAS Portal とプロバイダの管理者のみが知っている秘密共有キーに登録されます。

OracleAS Portal では、Hashed Message Authentication Code (HMAC) アルゴリズムを使用して計算されたデジタル署名が各メッセージとともにプロバイダに送信されます。プロバイダは、そのプロバイダ独自の共有キーのコピーを使用して署名を確認し、メッセージを認証できます。この手法は、プロバイダとの Secure Socket Layer (SSL) 通信で、クライアント証明書のかわりに使用できます。

OracleAS Portal では、ユーザー情報、共有キーおよびタイムスタンプに基づいて署名が計算されます。署名とタイムスタンプは、SOAP メッセージの一部として送信されます。このタイムスタンプは、UTC（協定世界時、グリニッジ標準時の学術名）を基にしているため、異なるタイム・ゾーンにある複数コンピュータ間のメッセージに使用できます。

プロバイダはこのメッセージを受信すると、その署名の独自のコピーを作成します。署名が合致すると、メッセージのタイムスタンプを現在の時刻と比較します。2つの時刻の差が許容範囲内の場合、メッセージは本物とみなされて処理されます。

セキュリティや管理上の問題が発生する可能性があるため、1つのプロバイダ・インスタンスで複数の共有キーはサポートできません。たとえば、共有キーのコピーの1つがなんらかの方法で侵害された場合、プロバイダの管理者は新しいキーを作成してすべての OracleAS Portal クライアントに配布し、クライアントはそのプロバイダ定義を更新する必要があります。この問題を回避するには、異なるプロバイダ・サービスをデプロイして、サービスごとに一意の共有キーを指定します。各プロバイダ・サービスには独自のデプロイ・プロパティ・ファイルがあるため、各サービスは他のサービスとは無関係に構成されます。同じプロバイダ・アダプタ内に複数のプロバイダ・サービスをデプロイすることによるオーバーヘッドは比較的小さいものです。

プロバイダの前に OracleAS Web Cache がない場合、プロバイダ・セッションの存続期間中に同じ署名 Cookie を使用するこの方法は、パフォーマンスと、リクエストを認証することによるセキュリティの間のトレードオフを意味します。署名 Cookie の値が計算されるのは、最初の SOAP リクエストによってプロバイダとのセッションが確立された後の1回のみです。プロバイダ・セッションのタイムアウトが短いほど、署名の計算頻度が高くなり、不正に再送信される表示リクエストに対するセキュリティが高まります。ただし、セッションの確立に必要な SOAP リクエストでは時間がかかります。

OracleAS Web Cache を使用して表示リクエストのレスポンスをキャッシュするプロバイダでも、同様のトレードオフが必要です。キャッシュされたコンテンツは、それを取得するための署名 Cookie が受信リクエストに含まれているという点で安全ですが、長期間にわたってコンテンツをキャッシュすると、プロバイダは不正な表示リクエストに対して無防備になります。

署名要素は、メッセージの傍受や再送信に対する保護を提供しますが、メッセージ・コンテンツの傍受や読取りを防ぐための処理は何も行いません。しかも、メッセージはプレーン・テキストで送信されます。メッセージの内容が認可されていない者に読み取られるのが心配な場合は、メッセージ認証を SSL と組み合わせて使用する必要があります。

メッセージ認証の利点は次のとおりです。

- プロバイダによって受信されたメッセージが正当な OracleAS Portal インスタンスから送信されたものであることが保証されます。

メッセージ認証の不利な点は次のとおりです。

- プロバイダが複数のポータルを処理する場合は、管理上の問題が発生します。
- セッションのタイムアウトを短くすることによってセキュリティを非常に高くした場合は、パフォーマンス上の影響があります。

このトピックの詳細は、『Oracle Application Server Portal 構成ガイド』を参照してください。

HTTPS 通信

OracleAS Portal とプロバイダとの間の通常の通信には、TCP をトランスポート・レイヤーとして使用してデータをプレーン・テキストで送信する、HTTP というネットワーク・プロトコルが使用されます。HTTPS は、TCP の上位にある保護された追加のレイヤー (SSL) を使用して、クライアントとサーバー間の通信を保護し、メッセージの傍受や読取りを困難にします。

SSL を使用して通信を受信する各エンティティ (たとえば、OracleAS Web Cache インスタンス) には、自由に利用できる公開鍵と、そのエンティティのみが知っている秘密鍵があります。エンティティに送信されるメッセージは、そのエンティティの公開鍵で暗号化されます。公開鍵で暗号化されたメッセージは秘密鍵によってのみ復号化できるため、悪意のある第三者がメッセージを傍受しても復号化はできません。

通信の署名に使用される証明書によって、公開鍵が実際に正しいエントリに属していることが保証されます。このような証明書は、認証局 (CA) として知られる信頼できるサード・パーティによって発行されます。この証明書には、エンティティの名前、公開鍵およびその他のセキュリティ資格証明が記述されており、サーバーの ID を確認するために SSL 通信のサーバー側にインストールされます。クライアントの証明書は、クライアントの ID を確認するためにクライアントにインストールすることもできます。

Oracle Wallet Manager は、公開鍵のセキュリティ資格証明を管理します。また、公開鍵と秘密鍵のペアの生成、CA に対する証明書要求の作成、およびサーバーでの証明書のインストールを行います。

このトピックの詳細は、『Oracle Application Server Portal 構成ガイド』を参照してください。

SSL の構成 プロバイダを OracleAS Portal インスタンスから登録するときは、URL を 1 つのみ入力します。これは、HTTP または HTTPS のいずれかを使用できますが、両方は使用できないことを意味します。

SSL メッセージの受信に使用される各サーバーの各ポートには、サーバー側の証明書がインストールされている必要があります。つまり、Web プロバイダとプロバイダをホスティングするサーバーの前には、OracleAS Web Cache インスタンス（存在する場合）が必要です。サーバー・ポートにインストールされている証明書によって、2 地点間の通信が暗号化されることが保証されますが、メッセージのソースは認証されません。信頼できる OracleAS Portal インスタンスとプロバイダとの間の通信を完全に保護するには、メッセージ認証もあわせて使用する必要があります。

OracleAS Portal の SSL 構成の詳細は、『Oracle Application Server Portal 構成ガイド』を参照してください。

LDAP (Oracle Internet Directory) セキュリティ

PDK-Java では、LDAP (Oracle Internet Directory) セキュリティにポートレットのセキュリティ・マネージャを使用します。また、PDK-Java では、ユーザー、グループおよび権限のリポジトリとして Oracle Internet Directory を使用します。ログイン・ユーザーに関する情報が取得され、そのユーザーにポートレットとポートレット内のデータの表示に必要な権限があるかどうか判断されます。Oracle Internet Directory セキュリティを有効にすると、プロバイダで次の処理が可能になります。

- グループに基づいてポートレットを保護できます。
- (独自のセキュリティ・マネージャを使用して) ポートレットの管理機能へのアクセスを制限します。
- 姓、名前、タイトル、電子メール、電話番号、グループ、写真など、Oracle Internet Directory に格納されているユーザー・プロパティ情報をすべて取得できます。
- OracleAS Portal のユーザーおよびグループを作成できます。

デフォルトでは、Oracle Internet Directory セキュリティは無効になっています。有効にするには、特定のプロバイダのデプロイ・プロパティ・ファイルを変更する必要があります。ポートレットを保護するために、Oracle Internet Directory を有効にして使用する手順は、次のとおりです。

1. デプロイ・プロパティ・ファイル (`provider_name.properties`) で Oracle Internet Directory マネージャを有効にします。

```
oidManager=true  
oidAdminClass=class_that_extends_oracle.portal.provider.v2.oid.OidInfo
```

2. `OidInfo` という簡単なクラスを拡張して、Oracle Internet Directory の接続情報を指定します。
3. プロバイダ定義ファイルに、ポートレットを表示できるグループのリストを指定します。

```
<group>cn=group1,cn=groups,dc=us,dc=oracle,dc=com</group>
```

プロバイダは、`OidInfo` クラスに指定した情報を使用して Oracle Internet Directory に接続します。ポートレットは、指定の資格証明（たとえば、ユーザー名とパスワード）を使用して Oracle Internet Directory にアクセスし、指定されたタスクを実行します。特にプロバイダ接続用として、ポートレットでリクエストされるタスクの実行に必要な最低限の権限を持つ Oracle Internet Directory ユーザーを作成することをお勧めします。たとえば、ポートレットでグループ情報のチェックのみを行う場合は、管理者として Oracle Internet Directory に接続しないでください。

Oracle Internet Directory セキュリティの実装 PDK-Java には、特に Oracle Internet Directory との統合用に一連のデフォルト・クラスが用意されています。これらのクラスは、ポートレットから Oracle Internet Directory への接続を処理し、OracleAS Portal グループに基づいてポートレットを保護し、さらに Oracle Internet Directory 内からユーザー・プロパティ情報にアクセスすることを可能にします。Oracle Internet Directory との統合用に Web プロバイダで使用するクラスは、次のとおりです。

- `oracle.portal.provider.v2.oid.OidInfo` は、開発者が指定した Oracle Internet Directory 接続情報を受信して、Oracle Internet Directory に接続します。独自のポートレットを作成するときは、このクラスを拡張して、プロバイダから Oracle Internet Directory に安全な接続詳細を送信する必要があります。
- `oracle.portal.sample.v2.devguide.oid.UnsafeOidInfo` は `OidInfo` の拡張機能で、ポートレットのセキュリティをテストする簡単な方法を提供します。このクラスは、PDK-Java の Oracle Internet Directory サンプルで使用され、Oracle Internet Directory 接続情報（後の説明を参照）についてデプロイ・プロパティ・ファイルを解析します。このクラスは、テストおよび開発の目的でのみ使用してください。本稼働用に使用するのは安全ではありません。

- `oidManager` は、デフォルトでは `false` に設定されています。Oracle Internet Directory を有効にするには、`provider_name.properties` でこのクラスを `true` に設定する必要があります (Web アプリケーションに存在するプロバイダが1つのみの場合は、`provider_name.properties` が `_default.properties` と同じであることを確認してください)。次に例を示します。

```
serviceClass=oracle.webdb.provider.v2.adapter.soapV1.ProviderAdapter
loaderClass=oracle.portal.provider.v2.http.DefaultProviderLoader
showTestPage=true
definition=providers/lab_provider/provider.xml
autoReload=true
oidManager=true
oidAdminClass=oracle.portal.sample.v2.devguide.oid.UnsafeOidInfo
oidHost=myhost.mydomain.com
oidPort=oidPort
oidUser=oidUser
oidPasswd=oidPassword
```

- `oidAdminClass` は、`OidInfo` を拡張するクラスに設定されます。PDK-Java では、デフォルトで `UnsafeOidInfo` を使用しますが、その名前に示すように、このクラスは本稼働用に使用しないでください。
 - `oidHost` は、Oracle Internet Directory がホスティングされているマシンです。
 - `oidPort` は、Oracle Internet Directory で使用するポートです。
 - `oidUser` は、Oracle Internet Directory のアカウントです。
 - `oidPasswd` は、Oracle Internet Directory のパスワードです。

次に例を示します。

```
serviceClass=oracle.webdb.provider.v2.adapter.soapV1.ProviderAdapter
loaderClass=oracle.portal.provider.v2.http.DefaultProviderLoader
showTestPage=true
definition=providers/lab_provider/provider.xml
autoReload=true
oidManager=true
oidAdminClass=oracle.portal.sample.v2.devguide.oid.UnsafeOidInfo
oidHost=myhost.mydomain.com
oidPort=oidPort
oidUser=oidUser
oidPasswd=oidPassword
```

- `oracle.portal.provider.v2.security.GroupSecurityManager` は、プロバイダとそのポートレットへのアクセス権限を持つグループを管理します。このクラスはポートレット固有で、プロバイダ定義ファイルから情報を取得します。プロバイダのポートレットごとに、異なるグループ設定を指定することもできます。このタグを使用して設定できるグループの数に制限はありませんが、Web プロバイダは各グループを解析して検証するため、グループ数が多いとパフォーマンスが低下する場合があります。
- `<group>` は `provider.xml` 内のタグで、グループ管理を行います。このタグは、ポートレットにアクセス可能なグループのリストを表示します。このグループ情報の大 / 小文字の区別は、Oracle Internet Directory と同じです。

注意： 次の例で参照している `portal_instance_id` は、使用しているインストールに固有な内容です。インスタンス ID を検索する場合は、『Oracle Internet Directory 管理者ガイド』を参照してください。

```
<securityManager class="oracle.portal.provider.v2.security.
  GroupSecurityManager">
  <group>cn=DBA,cn=portal_instance_id,cn=groups,
    dc=us,dc=oracle,dc=com</group>
</securityManager>
```

Oracle Internet Directory セキュリティの**利点**は次のとおりです。

- ポートレットを保護する簡単で強力な方法を提供します。
- ユーザーのグループ・メンバーシップに基づいてポートレット内のデータを保護します。
- Web プロバイダとして公開されているポートレットから直接ユーザーとグループを作成します。

Oracle Internet Directory セキュリティの**不利な点**は次のとおりです。

- Oracle Internet Directory を介してポートレットを認可するときに、パフォーマンスがわずかに低下します。LDAP サーバーからグループ情報を取得するときにコストがかかりますが、このコストが発生するのは、ユーザーがセッション内のポートレットに初めてアクセスするときのみです。
- プロバイダは Oracle Internet Directory にアクセスする必要があります。
- プロバイダで使用するすべての OracleAS Portal インスタンスは、同じ Oracle Internet Directory インスタンスを使用することが前提です。

Oracle Internet Directory を使用してプロバイダを保護する方法、または Oracle Internet Directory を使用して保護されたサンプル・ポートレットを設定する方法は、テクニカル・ノート「Installing the OID Portlets」を参照してください。

ポートレットの表示 Java ポートレットに追加したセキュリティ・マネージャの動作を示す手順は、次のとおりです。

1. `provider.xml` の `<group>` タグに指定されているグループのメンバーであるユーザーで、OracleAS Portal インスタンスにログインしていることを確認します。
2. PUBLIC に対して表示可能な既存のページを使用するか、新しいページを作成します。
3. Java ポートレットをページに追加します。
4. 新しいページへの直接 URL を書き留めておきます。
5. 「ログアウト」をクリックします。
6. 手順 4 で書き留めた URL をブラウザのアドレス・バーに入力してページに直接アクセスするか、`provider.xml` にリストされているグループに属さないユーザーを使用して OracleAS Portal にログインします。

手順 2 で作成したページが表示されますが、手順 3 で作成したポートレットは表示されません。ページにポートレットを追加したときは、ポートレット表示の認可を受けたユーザーとしてログインしています。PDK ランタイムはこれを検出し、ポートレットの追加を許可しました。ログアウトしてページを表示したときは、ポートレット表示を許可されたグループにはすでに属していなかったため、PDK フレームワークはポートレットのコンテンツのレンダリングを許可しませんでした。

図 5-44 開発者用のページとポートレット



再度ログインしてページを表示すると、ポートレットの存在を確認できます。

図 5-45 開発者 / 管理者用のページとポートレット



キャッシュを使用したポートレットのパフォーマンスの向上

この章の前の各項では、完全に機能する Java ポートレットを PDK フレームワークを使用して作成する方法を説明しました。ポートレットの基本機能の設定が完了した後は、ポートレットのパフォーマンスに注目する必要があります。

キャッシュは、高度な動的コンテンツを含む Web サイトのパフォーマンスの向上に使用される一般的な手法です。この手法では、キャッシュと呼ばれる待機時間の短い大きなデータ・ストアを備えたローカル・エージェントを使用してリクエストをプロキシ化することで、動的コンテンツのデータの取得と出力の生成に伴うオーバーヘッドを大幅に削減できます。キャッシュ・エージェントは、リクエストに対して次のいずれかの方法で応答します。

- リクエストされたコンテンツの有効なバージョンがキャッシュ内に存在する場合、エージェントはキャッシュされた既存のコピーをそのまま戻します。これによって、コンテンツの取得と生成というコストがかかるプロセスを省くことができます。この状況はキャッシュ・ヒットと呼ばれます。
- リクエストされたコンテンツの有効なバージョンがキャッシュ内に存在しない場合、エージェントはリクエストをその宛先に転送し、コンテンツが戻されるのを待機します。エージェントはコンテンツをリクエスト元に戻し、そのキャッシュ内にローカル・コピーを格納して、同じコンテンツに対するリクエストが後で発生した場合に再利用します。この状況はキャッシュ・ミスと呼ばれます。

Web プロバイダは、動的コンテンツ（つまり、ポートレット）の生成元であり、多くの場合、デプロイされている OracleAS Portal インスタンスから離れた場所にあります。このため、キャッシュによるパフォーマンス向上が可能となります。OracleAS Portal のアーキテクチャは、それ自身がキャッシュに適しています。これは、すべてのレンダリング・リクエストが、ポータルの中間層に存在する **Parallel Page Engine (PPE)** という単一のページ作成エージェントから発生するためです。Web プロバイダでレンダリングされたポートレットを PPE でキャッシュし、キャッシュされたコピーを再利用して後続のリクエストを処理することによって、Web プロバイダがページ作成にもたらすオーバーヘッドを最小限にできます。

Web プロバイダでは、どれがアプリケーションに最も適しているかによって、次の 3 種類のキャッシュ方法のいずれも使用できます。これらのキャッシュ方法の主な違いは、コンテンツが有効かどうかを判断する方法にあります。

- 1. 有効期限ベースのキャッシュ:** プロバイダはレンダリング・リクエストを受信すると、そのレスポンスに有効期限のスタンプを付けます。レンダリングされたレスポンスはキャッシュに格納され、同じコンテンツに対する後続のすべてのリクエストは、有効期限が経過するまで、このキャッシュを使用して処理されます。このキャッシュ・スキームは最も簡単でパフォーマンスが優れていると考えられます。それは、キャッシュ妥当性をテストするためのオーバーヘッドがほとんどなく、ネットワークのラウンドトリップを伴わないためです。有効期限ベースのキャッシュは、コンテンツの使用期間が効率よく定義されているアプリケーションに適しています。コンテンツの使用期間が確実ではない場合、有効期限ベースのキャッシュは効果的ではありません。詳細は、5-95 ページの「[キャッシュのアクティブ化](#)」および 5-96 ページの「[有効期限ベースのキャッシュの追加](#)」を参照してください。
- 2. 失効化ベースのキャッシュ:** 失効化ベースのキャッシュは、基本的には有効期限ベースのキャッシュと同様に機能しますが、任意の時点でキャッシュのコンテンツを期限切れ、つまり失効できる点が異なります。キャッシュのコンテンツの失効化は、通常、イベントによってトリガーされます。

たとえば、当日の予定を表示するカレンダー・ポートレットの場合、ポートレットのコンテンツが生成されるのは、初めてその日のカレンダーを表示したときの 1 回のみです。予定の追加や取消し、予定時間の変更などその日の予定に変更が発生するまで、コンテンツはキャッシュされたままです。各変更イベントは、カレンダー・アプリケーションでアクションをトリガーできます。変更イベントが発生すると、カレンダー・アプリケーションは、その変更の影響を受けるキャッシュされたポートレット・コンテンツに対して失効リクエストを生成できます。カレンダー・ポートレットを含むページを次回に表示すると、キャッシュにはエントリが含まれていません。Web プロバイダは、変更後の予定で新しいコンテンツを再生成するように連絡を受けます。

この方法は、非常に効率的にコンテンツをキャッシュできます。これは、コンテンツの生成元（つまり、Web プロバイダ）が連絡を受けるのは新しいコンテンツの生成が必要となるときのみで、固定した再生成スケジュールに拘束されないためです。詳細は、5-95 ページの「[キャッシュのアクティブ化](#)」および 5-97 ページの「[失効化ベースのキャッシュの追加](#)」を参照してください。

3. **妥当性チェック・ベースのキャッシュ**: レンダリング・リクエストを受信したプロバイダは、そのレスポンスにバージョン識別子 (または「E」タグ) のスタンプを付けます。この方法でもレスポンスはキャッシュされますが、キャッシュされたレスポンスを PPE が再利用する前に、キャッシュされたバージョンがまだ有効かどうかを判断する必要があります。そのために、キャッシュされたコンテンツのバージョン識別子を含めたレンダリング・リクエストをプロバイダに送信します。プロバイダは、このバージョン識別子が有効であるかどうかを判断します。バージョン識別子が有効である場合、プロバイダはコンテンツを含まない軽量のレスポンスを PPE に即座に送信し、キャッシュされたバージョンを再利用できることを通知します。バージョン識別子が無効な場合、プロバイダは新しいバージョン識別子を含む新しいコンテンツを生成し、以前にキャッシュされたバージョンを置き換えます。このキャッシュ方法では、PPE は、常にプロバイダに対してコンテンツが最新かどうかを確認する必要があります。キャッシュされたコピーの妥当性は、プロバイダのロジックによって判断されます。この方法では、固定期間に依存するのではなく、キャッシュされたコンテンツの使用をプロバイダが管理できるという利点があります。詳細は、5-95 ページの「[キャッシュのアクティブ化](#)」および 5-102 ページの「[妥当性チェック・ベースのキャッシュの追加](#)」を参照してください。

前提

1. 5-35 ページの「[PDK-Java ポートレットの構築](#)」を読み終えて理解していること。
2. ウィザードで作成したポートレットがページに正常に追加されていること。

キャッシュのアクティブ化

OracleAS Portal のキャッシュ機能を Web プロバイダで使用するには、最初に、中間層キャッシュをアクティブにする必要があります。このキャッシュは、`mod_plsql` (データベース・プロシージャおよびデータベース・プロバイダを HTTP でコールする Oracle HTTP Server プラグイン) で使用されるキャッシュと同じであるため、PL/SQL キャッシュと呼ばれます。

通常、キャッシュに関する構成詳細の責任は、OracleAS Portal 管理者にあります。

失効化ベースのキャッシュの場合は、Web プロバイダの前に OracleAS Web Cache を構成する必要があります。リモートの Web プロバイダには OracleAS Web Cache がインストールされていない場合が多いことに注意してください。OracleAS Web Cache の詳細は、『Oracle Application Server Web Cache 管理者ガイド』を参照してください。

OracleAS Web Cache をインストールして構成した後は、OracleAS Web Cache マネージャについて次の点を確認してください。

- Web プロバイダのホスト名とポートを指していること。
- キャッシュ・ルールによってプロバイダのコンテンツのキャッシュが発生しないこと。コンテンツは、プロバイダがそのレスポンスで生成する `Surrogate-Control` ヘッダーに従ってキャッシュされる必要があります。

有効期限ベースのキャッシュの追加

有効期限ベースのキャッシュは、実装が最も簡単なキャッシュ・スキームの1つで、XML プロバイダ定義内で宣言によってアクティブにできます。利用する `ManagedRenderer` の出力には、出力のキャッシュ時間（分）をその `pageExpires` プロパティに設定することによって、有効期限を設定できます。たとえば、ポートレット出力を1分間キャッシュすると仮定します。

1. 5-35 ページの「[PDK-Java ポートレットの構築](#)」の説明に従って、ポートレット・ウィザードを使用してポートレットを作成した後、`provider.xml` ファイルを編集し、`showPage` の `pageExpires` プロパティ・タグを1に設定します。これによって、ポートレットの1分間の期限切れエントリが設定されます。

デフォルトでは、ウィザードによって、`showPage` の圧縮された標準的なタグが生成されます。`pageExpires` のサブタグを含めるには、このタグを拡張する必要があります。

```
<showPage class="oracle.portal.provider.v2.render.http.ResourceRenderer">
  <resourcePath>/htdocs/mycacheportlet/MyCachePortletShowPage.jsp
  </resourcePath>
  <PageExpires>1</PageExpires>
</showPage>
```

2. JSP コードを表示ページに追加して、ポートレットが1分間キャッシュされることをテストします。ここでは、現在の時刻を JSP に追加します。

```
<%@page contentType="text/html; charset=windows-1252"
import="oracle.portal.provider.v2.render.PortletRenderRequest"
import="oracle.portal.provider.v2.http.HttpCommonConstants"
import="java.Util.Date"
import="java.text.DateFormat"
%>

<%
PortletRenderRequest pReq = (PortletRenderRequest)
request.getAttribute(HttpCommonConstants.PORTLET_RENDER_REQUEST);
DateFormat df = DateFormat.getDateInstance(DateFormat.LONG,
DateFormat.LONG,pReq.getLocale());
String time = df.format(new Date());
%>

<P>Hello <%=pReq.getUser().getName() %>.</P>
<P>This is the <b><i>Edit</i></b> render mode!</P>
<P>This information is correct as of <%=time%>.</P>
```

ポートレットを表示して、時間（秒を含む）が1分間で一定していることを確認します。時間が経過すると、ポートレットは最新の時刻を表示し、新しいキャッシュが設定されます。

失効化ベースのキャッシュの追加

OracleAS Web Cache を使用すると、コンテンツに対するリクエストは HTTP で送信され、コンテンツがキャッシュから戻されるか、または HTTP リクエストがコンテンツの生成元に転送されます。OracleAS Web Cache に戻されたコンテンツは、リクエスト元に戻される前にキャッシュに追加されます。キャッシュのコンテンツは、失効リクエストを OracleAS Web Cache に直接送信することで失効化されます。PDK-Java では、Java API for Web Cache (JAWC) を使用して失効リクエストを生成します。この項では、OracleAS Web Cache の構成方法、および PDK-Java に付属する失効化ベースのキャッシュのサンプルの使用方法を説明します。

OracleAS Web Cache に送信されたすべてのリクエストがキャッシュされるわけではありません。コンテンツをキャッシュするには、OracleAS Web Cache に対してコンテンツのキャッシュを指示するディレクティブがコンテンツに含まれている必要があります。通常、OracleAS Web Cache では、リクエストに関連付けられた URL をキャッシュ・キーとして使用しますが、Surrogate-Control ヘッダーと呼ばれる特別な HTTP ヘッダーをレスポンスに設定すると追加のキーを指定できます。

Java ポートレットを構成して失効化ベースのキャッシュを使用するには、次の処理を実行する必要があります。

- OracleAS Web Cache を構成します。詳細は、『Oracle Application Server Web Cache 管理者ガイド』を参照してください。
- プロバイダ・サブレット・レベルで失効化ベースのキャッシュをオンにします。
- 失効ポートを定義します。
- provider.xml ファイルで失効化ベースのキャッシュをアクティブ化します。
- ポートレットでキャッシュを失効化するには、ポートレットの JSP コードを更新して失効リクエストをトリガーします。

プロバイダ・サブレットの構成 失効化ベースのキャッシュを有効にするには、そのキャッシュをプロバイダ・サブレット・レベルでオンにします。PDK-Java の Web アプリケーション・デプロイメント・ディスクリプタである web.xml 内にある初期化パラメータにフラグが設定されます。次に例を示します。

```
<servlet-class>oracle.webdb.provider.v2.adapter.SOAPServlet</servlet-class>
  <init-param>
    <param-name>invalidation_caching</param-name>
    <param-value>true</param-value>
  </init-param>
</servlet>
```

ここにフラグが定義されていないと、失効化ベースのキャッシュはオフになります。フラグの状態は、プロバイダのテスト・ページを表示して簡単にチェックできます。次に例を示します。

```
http://<provider_hostname>:<port>/jpdk/providers/webcache/MyWebProvider
```

OracleAS Web Cache の失効ポートの定義 PDK-Java が自動的に事前インストールされている Oracle Application Server インストール・タイプ（たとえば、OracleAS Portal and Wireless タイプのインストール）を使用する場合は、OracleAS Web Cache の失効化設定がすでに MID_TIER_ORACLE_HOME/portal/conf/cache.xml に事前構成されていることを確認する必要があります。事前構成されている場合、この項の説明は無視して 5-100 ページの「XML プロバイダ定義の構成」に進んでください。そうでない場合は、OracleAS Web Cache の失効ポートレットを構成する必要があります。

最初に必要なのは、アプリケーション・サーバーに関連付けられた OracleAS Web Cache インスタンスの失効ポートのユーザー名とパスワードです。これらを取得した後は、提供されている oracle.webdb.provider.v2.cache.Obfuscate Java ユーティリティを使用して、このユーザー名とパスワードを、JAWC API に必要な書式のわかりにくい文字列に変換します。デフォルトの Oracle Application Server インストールでは、通常、失効ポートのユーザー名は invalidator で、パスワードはアプリケーション・サーバー管理者のパスワードと同じです。たとえば、失効ポートのユーザー名に invalidator を、パスワードに welcome を指定して、Oracle Application Server を D:¥as904 にインストールするとします。次のコマンドを実行します。

注意： 次に示すコマンドは、『PDK-Java installation guide』で指示されているように、pdkjava.jar が ORACLE_HOME/portal/jlib に存在し、jawc.jar が ORACLE_HOME/webcache/jlib に存在することを前提にしています。

スタンドアロンの Oracle Application Server Containers for J2EE インストールを使用している場合は、Java 実行可能ファイルへのパスを外部の Java 2 SDK インストールに置き換える必要があります。

```
D:¥as904¥jdk¥bin¥java -classpath D:¥as904¥portal¥jlib¥pdkjava.jar
oracle.webdb.provider.v2.cache.Obfuscate invalidator:welcome
```

このコマンドが正常に実行されると、次に示すような 16 進文字列が出力されます。

```
0510198d5df8efd5779406342be2528aa0cccb179ea6b77baf49f019f5075a3a11
```

この情報を使用して JAWC 構成ファイル cache.xml を作成します。この構成ファイルは、1 つ以上の OracleAS Web Cache インスタンスとその失効ポートを指定します。次に例を示します。

```
<?xml version="1.0">
<webcache>
<invalidation
  host=cache.mycompany.com
  port=4001
  authorization="0510198d5df8efd5779406342be2528aa0cccb179ea6b77baf49f019f5075a3a11"/>
</webcache>
```

複数の OracleAS Web Cache を定義できます。この場合、構成ファイルの各 OracleAS Web Cache には同じ失効リクエストが送信されます。次に例を示します。

注意： タグ名 `connectionPool`、`invalidationConnection1` および `invalidationConnection2` はユーザー定義の名前です。テンプレート・ファイルは、次のディレクトリに用意されています。

```
MID_TIER_ORACLE_HOME/j2ee/OC4Jinstance/applications/jpdk/jpdk/WEB-INF/
providers/webcache
```

```
<?xml version="1.0">
<webcache>
<connectionPool>
<invalidationConnection1
  host=cache.mycompany.com
  port=4001
authorization="0510198d5df8efd5779406342be2528aa0cccb179ea6b77baf49f019f5075a3a11"/>
<invalidationConnection2
  host=cache.mycompany.com
  port=4002
authorization="0510198d5df8efd5779406342be2528aa0cccb179ea6b77baf49f019f5075a3a11"/>
</connectionPool>
</webcache>
```

JAWC では、システム・プロパティ `oracle.http.configfile` を使用してこの構成ファイルを検索します。Oracle Application Server インストール内の Oracle Application Server Containers for J2EE インスタンスに対してこのシステム・プロパティを設定するには、PDK-Java がインストールされている特定のインスタンスの `oc4j.properties` ファイル（たとえば、`MID_TIER_ORACLE_HOME/j2ee/OC4Jinstance/config/oc4j.properties`）に適切な行を追加して、そのインスタンスを再起動します。次に例を示します。

```
oracle.http.configfile=fully_qualified_filename
```

Oracle Application Server Containers for J2EE をスタンドアロンで実行している場合は、起動するために使用するコマンドラインでオプションを指定できます。

```
java -Doracle.http.configfile=<fully_qualified_filename> -jar oc4j.jar
```

注意： キャッシュ構成ファイルの場所はシステム・プロパティとして定義されるため、サーバー・インスタンスごとに定義できるキャッシュは1つのみです。各 OracleAS Web Cache 構成の背後に、2つの異なる Web プロバイダを指定することはできません。

XML プロバイダ定義の構成 provider.xml でタグの組合せを使用すると、ポートレット全体またはそのページの一部に対して自動失効化ベースのキャッシュをアクティブ化できます。自動失効化ベースのキャッシュをオンにするには、それを、次に示すように各ページまたはレンダラのレベルで宣言して、失効化ベースのキャッシュがすべてのページに対してアクティブ化されていることを示す必要があります。

```
<useInvalidationCaching>true</useInvalidationCaching>
```

ポートレットでカスタマイズ（編集モードまたはデフォルト編集モードで）をサポートしている場合は、レンダラに対しても <autoInvalidate>true</autoInvalidate> を宣言できます。編集モードで「OK」または「適用」をクリックすると、この宣言によって、キャッシュされたポートレット・コンテンツがすべて失効化されます。したがって、カスタマイズされた設定に基づいて、新しいマークアップが生成されます。

ページをキャッシュに保持する最大時間は、次の時間についての最小時間となります。

- 「グローバル設定」ページの「キャッシュ」タブに定義されている OracleAS Portal の最大有効期限
- OracleAS Portal で最大有効期限が指定されていない場合は、Web プロバイダのデフォルト（24 時間）
- <pageExpires> タグがある場合はその時間（分）

provider.xml の次の抜粋部分では、このポートレットは最大 5 分間キャッシュされ、カスタマイズ時には自動的に失効化されるように指定されています。

```
<renderer class="oracle.portal.provider.v2.render.RenderManager">
  <contentType>text/html</contentType>
  <renderContainer>true</renderContainer>
  <autoRedirect>true</autoRedirect>
  <autoInvalidate>true</autoInvalidate>
  <showPage class="oracle.portal.provider.v2.render.http.ResourceRenderer">
    <resourcePath>/htdocs/invalidation/invalidation1.jsp</resourcePath>
    <useInvalidationCaching>true</useInvalidationCaching>
    <pageExpires>5</pageExpires>
  </showPage>
  <editPage class="oracle.portal.sample.v2.devguide.invalidation.
    InvalidationEditRenderer"/>
</renderer>
```

注意： pageExpires タグは、通常の有効期限ベースのキャッシュでも使用します。これら 2 つの形式のキャッシュは、相互に排他的です。失効化ベースのキャッシュは、Web プロバイダと同じ場所にある OracleAS Web Cache インスタンスで発生します。有効期限ベースのキャッシュを使用して格納されたページは、OracleAS Portal の中間層にキャッシュされます。

手動によるキャッシュの失効化 リクエストが処理されたり、なんらかの情報が更新されたときに、ポートレットのキャッシュされたバージョンを失効化する必要が生じる場合があります。このような場合は、手動でキャッシュを失効化できます。

PDK-Java に付属する失効化ベースのキャッシュのポートレット・サンプルには、コンテンツがキャッシュされた時刻、および失効リクエストをトリガーするためのリンクを表示する単一のポートレットが含まれています。Web プロバイダに対するページ・リクエストがキャッシュを介して初めて行われたときは、そのレスポンスがキャッシュされます。ポートレットのコンテンツに対する後続のリクエストは、OracleAS Web Cache からコンテンツを戻すことによって実行されます。ポートレット下部にあるリンクをクリックすると、キャッシュからポートレットを削除する失効リクエストがプロバイダで生成されます。ポートレットに対する次のリクエストはプロバイダに転送され、プロバイダは現在の時刻で新しいポートレットを生成します。

OracleAS Web Cache に対する失効コールを実行するには、最初に、ServletInvalidationContext オブジェクトへのハンドルを用意する必要があります。このハンドルは、ServletInvalidationContextFactory クラスの static `getServletInvalidationContext()` メソッドをコールすると取得できます。

ハンドルを取得した後は、キャッシュ・キーを指定する必要があります。キャッシュ・キーでは、失効化する対象が、特定のポートレット・インスタンスか、あるポートレットのすべてのインスタンスか、またはあるプロバイダが管理するすべてのポートレットかを指定する必要があります。このタスクを実行するには、ServletInvalidationContext クラスのメソッド、またはそのスーパー・クラス InvalidationContext のメソッドを使用します。ここでは、特定のユーザー（ユーザー・レベル）についてポートレットをキャッシュするかどうかを指定できます。キャッシュ・キーにユーザーが指定されていない場合、キャッシュされたコンテンツはすべてのユーザーに戻されます。これは、システム・レベルのキャッシュを使用していることを意味します。

次のサンプル・コードでは、ポートレット・インスタンスを失効化し、`setFullProviderPath()` メソッドと `setPortletReference()` メソッドをコールします。キャッシュ・キーが設定されている場合は、InvalidationContext オブジェクトの `invalidate()` メソッドをコールします。このメソッドは、失効メッセージを OracleAS Web Cache に送信します。

```
ServletInvalidationContext inv =
    ServletInvalidationContextFactory.getServletInvalidationContext();
inv.setFullProviderPath
    ((ServletPortletRenderRequest)pReq);
inv.setPortletReference
    (pReq.getPortletReference());
int num = inv.invalidate();
```

詳細は、Web Cache のサンプル・プロバイダを参照してください。

妥当性チェック・ベースのキャッシュの追加

妥当性チェック・ベースのキャッシュの追加は多少複雑ですが、プロバイダに対するどのリクエストがキャッシュ・ヒットであるかを厳密に制御できます。ここでは、ポートレット内のデータが変更された場合のみキャッシュを更新する例を説明します。このアルゴリズムを実装するには、`prepareResponse` メソッドを上書きする必要があります。

`BaseManagedRenderer.prepareResponse` メソッドの署名は次のとおりです。

```
public boolean prepareResponse(PortletRenderRequest pr)
    throws PortletException,
           PortletNotFoundException
```

使用しているバージョンの `prepareResponse()` で、次の処理を実行する必要があります。

- キャッシュされたバージョン識別子を取得します。この識別子は、`HttpPortletRendererUtil.getCachedVersion()` メソッドをコールすることで、PPE によってレンダリング・リクエストに設定されます。

```
public static java.lang.String getCachedVersion
(PortletRenderRequest request)
```

- 以前にキャッシュされたバージョンが有効であることをポートレットが検出した場合は、`HttpPortletRendererUtil.useCachedVersion()` メソッドをコールして適切なヘッダーを設定する必要があります。また、`RenderManager` に対しては、ポートレット本体をレンダリングするために `renderBody()` をコールする必要がないことが指示されます。

```
public static void useCachedVersion(PortletRenderRequest request)
```

これ以外の場合は、`HttpPortletRendererUtil.setCachedVersion()` を使用して、新しいバージョンのポートレットを生成します。このポートレットがキャッシュされます。また、PPE に対しては、`renderBody()` メソッドをコールして、ポートレットのコンテンツを再生成する必要があることが指示されます。

```
public static void setCachedVersion(PortletRenderRequest request,
    java.lang.String version,
    int level)
    throws java.lang.IllegalArgumentException
```

妥当性チェック・ベースのキャッシュの場合、`provider.xml` を更新する必要はありません。ポートレットを表示するには、ページをリフレッシュするか、ポートレットをページに追加してコンテンツを更新します。コンテンツが変更されている場合、ポートレットは新しいコンテンツを表示します。コンテンツが変更されていない場合は、キャッシュされたバージョンのポートレットが表示されます。

多言語ポートレットの作成

この項では、異なる言語でレンダリングできる Java ポートレットの作成方法を説明します。ポートレットで使用する言語は、そのポートレットを表示しているポータル言語設定によって決まります。

この項を終了すると、必要な数の言語をサポートするポートレットを作成できるようになります。また、既存のポートレットを多言語をサポートするように変換することもできます。ポートレットが複数の言語をサポートするように作成されていると、新しい言語をプラグ・インするのが簡単になります。多言語の Java ポートレットの基本モデルは、標準の Java 国際化モデルと似ています。Java の国際化に関する知識がある場合は、容易にこのプロセスを習得できます。

前提

1. 5-35 ページの「[PDK-Java ポートレットの構築](#)」を読み終えて理解していること。
2. ウィザードで作成したポートレットがページに正常に追加されていること。

ポートレットの国際化

- [ポートレット・コンテンツの翻訳の提供](#)
- [ポートレット属性の翻訳の提供](#)

ポートレット・コンテンツの翻訳の提供 5-35 ページの「[PDK-Java ポートレットの構築](#)」の項では、Java ポートレット・ウィザードを使用してポートレットを作成しました。ウィザードで作成した基本的なメッセージは 1 つの言語でのみ利用でき、表示するテキストはポートレットのレンダラ・クラスにハードコーディングされています。ポートレットを複数の言語で利用できるようにするには、このような言語依存要素を、リソース・バンドルと呼ばれる独自のクラスに格納する必要があります。

リソース・バンドルの作成

リソース・バンドルは、ポートレットで利用する言語ごとに必要です。また、ポータルで選択された言語設定に対応するリソース・バンドルがない場合に使用する、デフォルトのリソース・バンドルも作成する必要があります。

- **デフォルト・リソース・バンドルの作成**
 1. Oracle JDeveloper で、`java.util` パッケージの `ListResourceBundle` を拡張する `MyProviderBundle` という Java クラスを作成します。このクラスには、オブジェクトの多次元配列が含まれており、この配列には、JSP の表示ページにある各言語依存要素を表すキー値のペアが保持されます。この実装は次のコードで示されます。

```
package mypackage2;
import java.util.ListResourceBundle;
public class MyProviderBundle extends ListResourceBundle
{
```

```
public static String HELLO_MSG = "MyPortletHelloMessage";
public static String INFO_MSG = "MyPortletInfoMessage";
public Object[] [] getContents()
{
    return contents;
}
static private final Object[] [] contents =
{
    {HELLO_MSG, "Hello"},
    {INFO_MSG, "This is the show page of the portlet and it is being generated
in the default language!"}
};
}
```

2. MyProviderBundle を保存します。

■ サポートする他の言語のためのリソース・バンドルの作成

ここで、ポートレットでサポートする各言語について、リソース・バンドル・クラスを作成する必要があります。それぞれのクラスの名前は、デフォルトのリソース・バンドル・クラスと同じである必要がありますが、末尾に言語コードを付加します。たとえば、フランス語をサポートする場合は、MyProviderBundle_fr という Java クラスを作成します。言語コード fr は、言語設定がフランス語に設定されている場合にポータルロケール・オブジェクトによって使用されるコードと同じです（ロケールの詳細は、java.util.Locale の JavaDoc を参照）。OracleAS Portal で言語設定を変更する場合は、現行のロケール・オブジェクトの値、つまりロケール・オブジェクトの言語コードを変更します。言語コードは、言語の呼称に関する ISO:639 コードに準拠しています。

1. フランス語のリソース・バンドルを作成するには、前述したように、MyProviderBundle_fr という Java クラスを作成します。
2. デフォルトのリソース・バンドルをテンプレートとして使用し、英語文字列を対応するフランス語に置き換えます。次に例を示します。

```
package mypackage2;

import java.util.ListResourceBundle;
public class MyProviderBundle_fr extends
ListResourceBundle
{
    public Object[] [] getContents()
    {
        return contents;
    }
    static private final Object[] [] contents =
    {
        {MyProviderBundle.HELLO_MSG, "Bonjour"},
        {MyProviderBundle.INFO_MSG, "Cette page est le 'show mode' de la portlet
et est generee dans la langue par default."}
    }
}
```

```
};
}
```

3. MyProviderBundle_fr を保存します。
4. リソース・バンドルを作成する各言語について、手順 1～3 を繰り返します。その際、クラス名を適切な言語コードを持つ名前に変更し、メッセージ文字列を適切な言語の同等のメッセージに変更します。

レンダラの更新

ここで作成したリソース・バンドルを使用するには、JSP の表示ページを更新し、ハードコーディングされたメッセージを、実行時にポータルロケール・オブジェクトと最も密接に対応するリソース・バンドルからメッセージを取り出す参照に置き換えます。

1. 表示ページを表す JSP を開き、次のように変更します。

```
<%@page contentType="text/html; charset=windows-1252"
    import="oracle.portal.provider.v2.render.PortletRenderRequest"
    import="oracle.portal.provider.v2.http.HttpCommonConstants"
    import="java.util.ResourceBundle"
%>

<%
    PortletRenderRequest pReq = (PortletRenderRequest)
        request.getAttribute(HttpCommonConstants.PORTLET_RENDER_REQUEST);

    <-- Get a resource bundle object for the current language. -->
    ResourceBundle b =
        ResourceBundle.getBundle("mypackage2.MyProviderBundle", pReq.getLocale());
%>

    <-- Pull the message from the appropriate resource bundle. -->
    <P> <%= b.getString(mypackage2.MyProviderBundle.HELLO_MSG) %>
        <%= pReq.getUser().getName() %>.</P>
    <P> <%= b.getString(mypackage2.MyProviderBundle.INFO_MSG) %></P>
```

2. JSP ページを変更します。

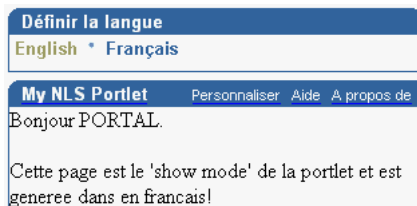
これで、ポートレットをリフレッシュして変更を表示できます。

図 5-46 英語のポートレット



フランス語のメッセージを表示するには、「言語の設定」ポートレットで言語を英語からフランス語に設定します。

図 5-47 フランス語のポートレット



ポートレット内のテキストは変更されますが、ポートレット・タイトルはデフォルト言語の英語のままであることを注意してください。また、`provider.xml` からリソース・バンドルを指し示すことによって、ポートレットで適切なポートレット属性（ポートレット名、ポートレット・タイトル、ポートレットの説明など）を設定できます。

ポートレット属性の翻訳の提供 プロバイダの定義ファイル `provider.xml` には、ポートレット名や説明などポートレットを記述する多数の属性が定義されています。これらの属性は、表示モードでポートレットのタイトル・バーなどで使用されるため、翻訳する必要があります。翻訳を提供する方法は 2 通りあり、いずれかを選択します。両方の方法について、次に概略を説明します。

- 方法 1: プロバイダ・レベルでリソース・バンドルを使用
- 方法 2: ポートレット・レベルでリソース・バンドルを作成

方法 1: プロバイダ・レベルでリソース・バンドルを使用

リソース・バンドルでポートレット属性に関する翻訳を提供し、リソース・バンドルで使用したキーを指定すると、`provider.xml` でこれらのリソース・バンドルを使用するように指定できます。この方法では、任意のキーを使用できるため、プロバイダの各種ポートレットで対応する各属性に対して異なるキーを使用することによって、プロバイダのすべてのポートレットが使用できるリソース・バンドルのセットを 1 つのみにできます。

■ リソース・バンドルの更新

1. デフォルトのリソース・バンドル `MyProviderBundle.java` を開きます。
2. ポートレット属性を表すリソース・バンドルに文字列を追加し、その文字列のテキストを追加します。

```
package mypackage2;

import java.util.ListResourceBundle;
public class MyProviderBundle extends ListResourceBundle
{
    public static String HELLO_MSG = "MyPortletHelloMessage";
    public static String INFO_MSG = "MyPortletInfoMessage";
    public static String PORTLET_NAME = "FirstPortletName";
    public static String PORTLET_TITLE = "FirstPortletTitle";
    public static String PORTLET_SHORT_TITLE = "FirstPortletShortTitle";
    public static String PORTLET_DESCRIPTION = "FirstPortletDescription";
    public static String TIMEOUT_MESSAGE = "FirstPortletTimeoutMessage";

    public Object[][] getContents()
    {
        return contents;
    }
    static private final Object[][] contents =
    {
        {HELLO_MSG, "Hi"},
        {INFO_MSG, "This is the show page of the portlet and it is being generated
            in the default language!"},
        {PORTLET_NAME, "MyNLSPortlet"},
        {PORTLET_TITLE, "My NLS Portlet"},
        {PORTLET_SHORT_TITLE, "MyNLSPortlet"},
        {PORTLET_DESCRIPTION, "My first ever NLS portlet, using my
            MyPortletShowPage.jsp"},
        {TIMEOUT_MESSAGE, "Timed out waiting for MyNLSPortlet"}
    };
}
```

3. `MyProviderBundle.java` を保存します。
4. `MyProviderBundle_fr.java` を開きます。`MyProviderBundle` で宣言した文字列と一致するフランス語文字列を含むように変更します。

```
package mypackage2;

import java.util.ListResourceBundle;
public class MyProviderBundle_fr extends ListResourceBundle
{
    public Object[][] getContents()
```

```
{
return contents;
}
static private final Object[][] contents =
{
{MyProviderBundle.HELLO_MSG, "Bonjour"},
{MyProviderBundle.INFO_MSG, "Cette page est le 'show mode' de la portlet
  et est generee en francais!"},
{MyProviderBundle.PORTLET_NAME, "MaPremierePortlet"},
{MyProviderBundle.PORTLET_TITLE, "Ma Portlet Multi-Langue"},
{MyProviderBundle.PORTLET_SHORT_TITLE, "Ma NLS Portlet"},
{MyProviderBundle.PORTLET_DESCRIPTION, "Ma premiere portlet
  multi-langue, utilisant mon renderer"},
{MyProviderBundle.TIMEOUT_MESSAGE, "Temps d'accès a la portlet
  demandee expire"}
};
}
```

5. MyProviderBundle_fr.java を保存します。

■ provider.xml の更新

1. XML プロバイダ定義ファイルを開き、ハードコーディングしたポートレット属性値を使用するかわりにリソース・バンドルを指すように変更します。

```
<portlet class="oracle.portal.provider.v2.DefaultPortletDefinition">
  <id>3</id>
  <resource>MyProviderBundle</resource>
  <nameKey>FirstPortletName</nameKey>
  <titleKey>FirstPortletTitle</titleKey>
  <ShortTitleKey>FirstPortletShortTitle</ShortTitleKey>
  <descriptionKey>FirstPortletDescription</descriptionKey>
  <timeout>10</timeout>
  <timeoutMessageKey>FirstPortletTimeoutMessage</timeoutMessageKey>
  <showEditToPublic>>false</showEditToPublic>
  <hasAbout>>true</hasAbout>
```

方法 2: ポートレット・レベルでリソース・バンドルを作成

PDK-Java では、ポートレット属性の翻訳を提供するために使用できるリソース・バンドル・キーのセットが定義されます。これらのキーを使用すると、「方法 1: プロバイダ・レベルでリソース・バンドルを使用」で実行したように provider.xml ファイルでリソース・バンドル・キーを指定する必要がなくなります。ただし、プロバイダの各ポートレットに対して独立した一連のリソース・バンドルを提供する必要があります。これは、各ポートレットで使用するキーは同じである必要がありますが、その値は異なるためです。

■ リソース・バンドルの更新

1. デフォルトのリソース・バンドル `MyProviderBundle.java` を開きます。
2. 前の項で実行した変更を削除し、`oracle.portal.provider.v2.PortletConstants` をインポートします。これで、文字列のかわりに次の定数を参照できます。`PortletConstants` を使用するときは、静的文字列を宣言する必要はありません。

```
{PortletConstants.NAME, "MyNLSPortlet"},
{PortletConstants.TITLE, "My NLS portlet"},
{PortletConstants.SHORTTITLE, "MyNLSPortlet"},
{PortletConstants.DESCRPTION, "My first ever NLS portlet"},
{PortletConstants.TIMEOUTMSG, "Timed out waiting for MyNLSPortlet"}
```

3. `MyProviderBundle.java` を保存します。
4. `MyProviderBundle_fr.java` を開きます。前の項で追加したポートレット属性を削除し、`oracle.portal.provider.v2.PortletConstants` をインポートして、文字列のかわりに定数を参照します。

```
{PortletConstants.NAME, "MaPremierePortlet"},
{PortletConstants.TITLE, "Ma Portlet Multi-Langue"},
{PortletConstants.SHORTTITLE, "Ma NLS Portlet"},
{PortletConstants.DESCRPTION, "Ma premiere portlet multi-langue,
    utilisant mon renderer"},
{PortletConstants.TIMEOUTMSG, "Temps d'accès à la portlet demandée
    expire"}
```

5. `MyProviderBundle_fr.java` を保存します。

■ provider.xml の更新

1. `provider.xml` で、「方法 1: プロバイダ・レベルでリソース・バンドルを使用」で実行したように文字列ごとに 1 つのタグを使用するのではなく、1 つのタグのみを使用する必要があります。ポートレット ID とタイムアウト秒数値の間にそのタグを追加します。

```
<resource>mypackage2.MyProviderBundle</resource>
```

Java 国際化の詳細は、Java チュートリアル「[Internationalization trail](#)」を参照してください。

ポートレットの表示

プロバイダを更新して Oracle Application Server Containers for J2EE にデプロイした後、ポートレットを含んだポータル・ページをリフレッシュします。リソース・バンドルが機能していることを確認するには、ページに「言語の設定」ポートレットを追加し、言語設定をフランス語に変更してみてください。デフォルトのリソース・バンドルは英語です。対応するリソース・バンドルのない他の言語を選択した場合は、ポートレットは英語で表示されることに注意してください。

Oracle JDeveloper を使用した Struts ポートレットの作成

この項では、OracleAS Portal で使用する Struts ポートレットを Oracle JDeveloper で作成するフレームワークについて説明します。Struts タグ・ライブラリを Oracle Application Server Portal Developer Kit (バージョン 9.0.4.0.2 以降) から Oracle JDeveloper に追加して既存のアプリケーションから Struts ポートレットを作成し、次に Oracle PDK Java ポートレット・ウィザードを使用して Java ポートレットを作成する方法を説明します。

OracleAS Portal および Apache Struts のフレームワーク

この項では、Apache Struts を OracleAS Portal とともに使用方法を説明します。Struts は、Model-View-Controller (MVC) 設計パターンの実装です。

- [Model-View-Controller の概要](#)
- [OracleAS Portal と Struts の統合](#)
- [まとめ](#)

Model-View-Controller の概要

データベース・アプリケーションは、次のような複数の異なるタスクを実行します。

- データ・アクセス
- ビジネス・ロジックの実装
- ユーザー・インタフェースの表示
- ユーザーの相互作用
- アプリケーション (ページ) フロー

MVC (Model-View-Controller) アーキテクチャは、データ表示などのアクティビティはデータ・アクセスから分離する必要があるという前提に基づいて、これらのタスクを区分する方法を提供します。このアーキテクチャによって、データソースをアプリケーションに簡単にプラグインでき、ユーザー・インタフェースをリライトする必要はありません。MVC では、アプリケーションを論理的にモデル、ビューおよびコントローラの 3 つのレイヤーに区分できます。

モデル

モデルは、アプリケーション・データとビジネス・ロジック用のリポジトリです。モデルの目的の1つは、データベースからデータを取得し、データをデータベースに保持することです。また、モデルでは、ビューからデータにアクセス可能にするなどの方法でデータの公開を行い、ビューから入力されたデータの検証と消込みを行うためにビジネス・ロジック・レイヤーを実装します。アプリケーション・レベルでは、モデルは、ユーザー・インタフェースと表示されるビジネス・データの間で検証および抽象化のレイヤーとして機能します。データベース・サーバー自体は、モデルの永続的なレイヤーとして機能します。

ビュー

ビューは、JSP を使用して、モデルのデータをレンダリングします。ビューのコードには、ハードコーディングされたアプリケーションやナビゲーション・ロジックは含まれていませんが、ユーザー・ロールに基づいて条件付きデータを表示するなどのタスクを実行するロジックが含まれている場合があります。エンド・ユーザーが、ビューでレンダリングされた HTML ページ内で操作を実行すると、イベントがコントローラに送信されます。これによって、コントローラは次のステップを決定します。

コントローラ

コントローラは MVC パターンの重要な基軸です。ビューで実行されるすべてのユーザー操作は、コントローラを介して送信されます。コントローラは、ブラウザからのリクエストの内容に基づいて、次の操作を実行します。

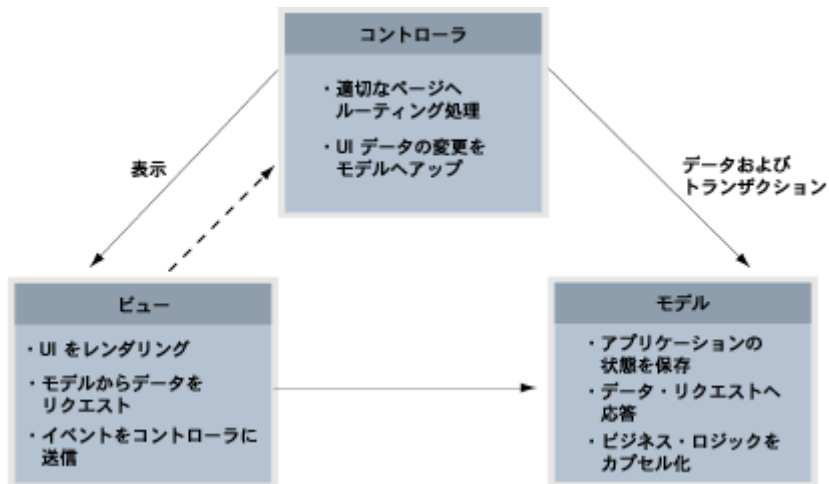
コントローラを起動する方法はいくつかあります。たとえば、URL 引数を使用して、リクエストを適切なコードにルーティングできます。MVC パターンでは、コントローラの機能を決定しますが、その動作は決定しません。

利点

MVC アーキテクチャによって、アプリケーションとその設計をモジュール単位で明確に表示できます。アプリケーション・ロジックの様々なコンポーネントとロールを区分することによって、開発者はアプリケーションを簡単で異なるコンポーネント（モデル、ビューおよびコントローラ）の集まりとして設計できます。このパターンは、保持や拡張が容易なアプリケーションの作成に役立ちます。たとえば、1つのビューを作成すると、同じビジネス・ロジックを使用して別のビューを簡単に作成できます。その使いやすさと再利用性のため、MVC パターンは、Web ベースのアプリケーション開発で最も広く使用されているパターンです。

次の図は、MVC パターンを従来の Thin クライアントの Web アプリケーションに適用する方法を示します。

図 5-48 MVC パターン



Apache Struts の概要

Apache Struts フレームワーク (<http://struts.apache.org>) は、Web アプリケーション作成用の最も一般的なフレームワークの 1 つで、MVC 設計パラダイムの JSP モデル 2 に基づくアーキテクチャを提供します。モデル 2 では、エンド・ユーザーのリクエストは、フローを制御し、JavaBeans、EJB、BC4J などのコンポーネントを使用してデータにアクセスして操作するサーブレットで管理されます。さらにモデル 2 では、JSP を使用して、アプリケーションのコンテンツを Web ブラウザにレンダリングします。このモデルは、JSP がブラウザ・リクエストとデータ・アクセスを管理する JSP モデル 1 とは異なります。

Struts フレームワークは、コントローラ・コンポーネントとして独自の HTTP サーブレットを提供します。Struts フレームワークは、アプリケーションのページ・フローが含まれる XML 構成ファイルによって起動されます。Struts はモデルを提供しませんが、開発者は、EJB、TopLink、BC4J、JD BC などのデータ・アクセス・メカニズムに Struts を統合できます。Struts のビューは、デフォルトで、Java 標準タグ・ライブラリ (JSTL) および JavaServer Faces (JSF) も含めて JavaServer Pages (JSP) を使用します。Struts が提供する一連の JavaBeans および JSP タグによって、MVC の様々なコンポーネントを使用できます。

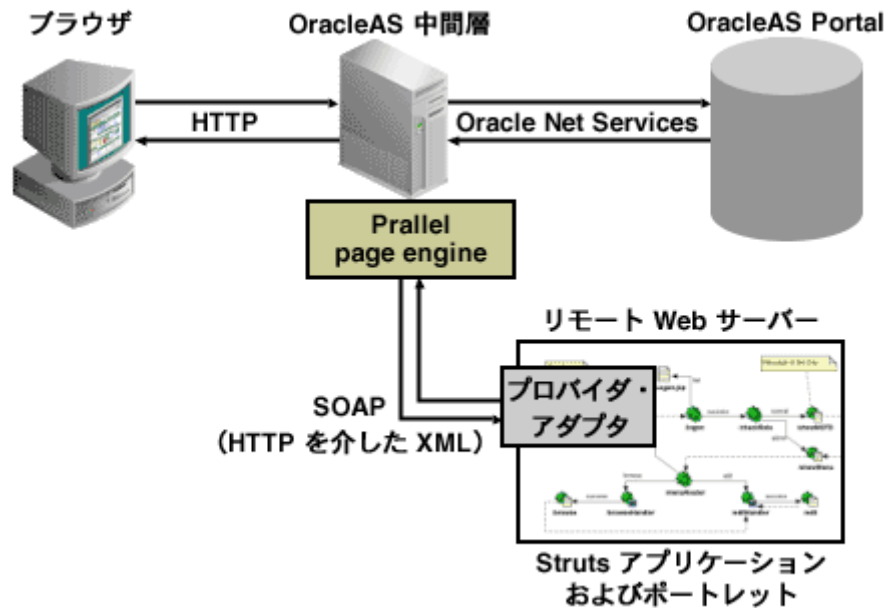
注意： JSTL および JSF の詳細は、Apache Software Foundation の Web サイト (<http://jakarta.apache.org/struts/faqs/kickstart.html>) で「FAQ」を参照してください。

OracleAS Portal と Struts の統合

Oracle Application Server Portal Developer Kit には、パーソナライズやキャッシュなど OracleAS Portal API の使用方法に関する多数の例やドキュメントが含まれています。ポートレット API では、アプリケーション・ロジックとビジネス・ロジックを統合できません。一方、Struts フレームワークを使用すると、MVC アーキテクチャを利用して、企業ポータル内でアプリケーションを作成および公開できます。

Oracle Struts ポートレット Struts フレームワークを使用してポートレットを作成したり、既存の Struts アプリケーションからポートレットを生成するには、すべてのコンポーネントを J2EE コンテナにデプロイする必要があります。OracleAS Portal のコンテキストでは、Struts アプリケーションは PPE によってコールされ、スタンドアロンの Struts アプリケーションのようにブラウザではコールされません。リクエストが送信されると、Oracle Parallel Page Engine (PPE) は Struts ポートレットをコールし、Struts ポートレットはそのリクエストを Apache Struts コントローラ・サーブレットに転送します。

図 5-49 Struts アプリケーションと OracleAS Portal の統合



次に示すコードは、プロバイダ定義ファイル (provider.xml) の一部です。

```
...
<renderContainer>true</renderContainer>
  <renderCustomize>true</renderCustomize>
  <autoRedirect>true</autoRedirect>
  <contentType>text/html</contentType>
  <showPage class="oracle.portal.provider.v2.render.http.StrutsRenderer">
    <defaultAction>showCustomer.do</defaultAction>
  </showPage>
</renderer>
...
```

showPage タグは、ポートレットの表示モードで実行されるビジネス・ロジックを定義します。Struts ポートレットの showPage には、次の 2 つの重要なコンポーネントが含まれません。

1. レンダラ・クラス (oracle.portal.provider.v2.render.http.StrutsRenderer)。PPE からポートレット・リクエストを受信し、プロキシとして機能してリクエストを Struts アクション・サーブレットに転送します。
2. defaultAction タグ。ポートレットが初めてコールされたときにデフォルトで使用する Struts アクションを定義します。

Oracle Application Server Portal Developer Kit を使用すると、Struts アプリケーションのビュー (Portal ビュー) を容易に開発できます。このビューによって、Portal スタイルを使用する Struts ポートレットでは整合性のとれた外観が維持され、エンド・ユーザーはポータル内でアプリケーションを使用できます。

Struts ポートレットを作成するには、デフォルトの Struts JSP タグの拡張である OracleAS Portal JSP タグを使用する必要があります。この開発プロセスは、スタンドアロンの Struts アプリケーションを作成するプロセスに似ています。Struts ポートレットの作成方法は、5-115 ページの「[Struts ポートレットの作成](#)」を参照してください。また、ポートレットと Struts アプリケーションは同じサーブレット・コンテキスト内に存在する必要があるため、両方の要素を含む単一の Web アプリケーションを作成する必要があります。Oracle JDeveloper でこのような Web アプリケーションを簡単に作成する方法は、次の項の 5-115 ページの「[Struts ポートレットの作成](#)」を参照してください。

まとめ

Apache Struts は、MVC 設計パラダイムの明確で簡単な実装を提供しているため、事実上、MVC ベースの J2EE アプリケーションを開発する際の標準になっています。このフレームワークによって、ポートレット開発者は、アプリケーションの様々なコンポーネントを区分し、Struts コントローラを利用して既存の Struts アプリケーションを OracleAS Portal に容易に公開できるため、既存のビジネス・ロジックをすべて変更する必要はありません。

Oracle Application Development Framework (ADF) は、Struts アクション・サーブレットをコントローラとして使用したり、独自のコントローラ (ADF コントローラ、ADFc) を使用することもできます。このフレームワークによって、Struts ポートレットと ADF ポートレットを同じ方法で作成できます。

注意： Oracle Application Server Portal Developer Kit については、Portal Center (<http://portalcenter.oracle.com>) を参照してください。

Struts ポートレットの作成

OracleAS PDK (バージョン 9.0.4.0.2) では、Apache Struts アプリケーションを統合するための新しい拡張機能が導入されました。この項では、既存の Struts アプリケーションからポートレットを作成する方法を説明します。また、その作成手順に従うと、Model-View-Controller パラダイムを使用するポートレットも作成できます。Apache Struts フレームワークの詳細は、5-110 ページの「[OracleAS Portal および Apache Struts のフレームワーク](#)」を参照してください。

この項で説明する PDK-Java の拡張機能は、Apache Struts プロジェクトの 1.1 本稼働バージョンに準拠しています。Oracle JDeveloper バージョン 9.0.3.x には、デフォルトで Apache Struts バージョン 1.1b2 (1.1 ベータ 2) が含まれています。Oracle JDeveloper 9.0.3.x を使用して Struts アプリケーションおよびポートレットを作成する場合は、Oracle JDeveloper の Struts ライブラリを本稼働バージョンの Struts にアップグレードする必要があります。Apache Struts 1.1 本稼働バージョンは、Oracle JDeveloper 10g (9.0.4) 以降に付属していません。

この項では、次の手順を説明します。

- [Struts ポートレットの作成](#)
- [プロバイダの登録](#)
- [まとめ](#)

Struts ポートレットの作成

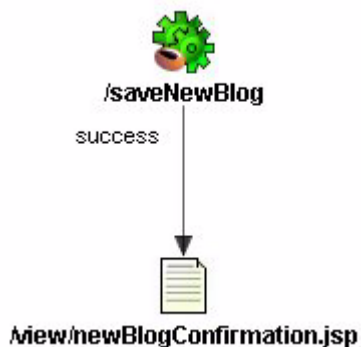
既存の Struts アプリケーションの一部をポートレットとして公開するには、最初に、アプリケーションの Portal ビューとして機能する新しいビューを作成する必要があります。このビューでは、既存のオブジェクト (Actions、ActionForm など) とともに新しいマッピングと JavaServer Pages を使用します。

次の例では、新しいエントリを Web ログ出力 (Blog) に追加できるポートレットを作成します。

図 5-50 Blog の送信



図 5-51 Blog エントリの保存



prepareNewBlog は、リクエストを enterNewBlog.jsp ページにリダイレクトする簡単な空のアクションです。このページには、新しい Blog を送信するためのフォームが表示されます。

struts-config.xml で対応するエントリは次のとおりです。

```
<action path="/prepareNewBlog" scope="request"
type="view.PrepareNewBlogAction" >
  <forward name="success" path="/view/enterNewBlog.jsp"/>
</action>
<action path="/saveNewBlog" name="blogForm" scope="request"
type="view.SaveNewBlogAction" input="/view/enterNewBlog.jsp" >
  <forward name="success" path="/view/newBlogConfirmation.jsp"/>
</action>
```


ポートレット・アクションをホスティングするための新しいフローとビューの作成 新しいビューを作成するには、最初に、様々なアクションとリクエストを Portal 固有の JSP にリダイレクトする ActionMappings (ページ・フロー) の新しいセットを作成します。

```
<action path="/portal/prepareNewBlog" scope="request"
  type="view.PrepareNewBlogAction" >
  <forward name="success" path="/view/portal/enterNewBlog.jsp"/>
</action>
<action path="/portal/saveNewBlog" name="blogForm" scope="request"
type="view.SaveNewBlogAction" input="/view/enterNewBlog.jsp" >
  <forward name="success" path="/view/portal/newBlogConfirmation.jsp"/>
</action>
```

変更するのはパス属性のみです。FormBean アクションは、アプリケーションのビジネス・ロジックが変更されないようにします。

新しい JSP の作成 前述の手順で指定したように、アクションによってリクエストが新しい JSP に転送され、JSP でポートレットのコンテンツがレンダリングされます。新しいポートレット・ビューの JSP は、HTML をスタンドアロン・ビューと共有できますが、ポートレットに関する次の点を確認してください。

- 他のポータル・ページと整合性のとれた外観を維持する Portal スタイルが使用されていること。
- HTML の表のセルに格納できる HTML コードが含まれていること (つまり、<html>、<body> および <frame> タグ以外)。
- ポータルで認識できるリンクおよびフォームをレンダリングしていること。これは、Struts ポートレットがコンテンツをインラインでレンダリングするために必要です。これによって、同じポートレット・コンテナ内でリクエストされたコンテンツがレンダリングされ、ユーザーをポータル・ページのコンテキスト内で管理できます。

Struts ポートレット内でインラインでレンダリングするには、OracleAS PDK タグを使用する必要があります。

```
<pdk-struts-html:form action="/portal/saveNewBlog.do">
...
...
</pdk-struts-html:form>
```

ポートレットのレンダリング中に、いずれかの JSP タグ (たとえば、pdk-struts-html:form タグ) でフォームを Parallel Page Engine (PPE) に送信し、PPE ではそのパラメータを Struts ポートレットに送信します。Struts コントローラは、これらのアクションのロジックを実行し、次の JSP をポータル・ページ内のポートレットに戻します。

PDK にはすべての Struts タグが含まれており、URL に関連するすべてのタグを拡張します。次に、PDK によって拡張されるタグを示します。

- **form:** HTML フォームを作成し、フォーム内にポータル・ページのコンテキストを埋め込んで、インラインのレンダリングを保証します。
- **link** および **rewrite:** ポータル・ページへのリンクを作成します。これらはインラインのレンダリングに必要です。
- **img:** Web プロバイダを指す絶対リンクを作成します。このタグを、ファイアウォールを備えたインターネット Web サイトのコンテキストで使用する場合は、プロバイダにインターネットから直接アクセスできることを確認する必要があります。これが不可能な場合は、OracleAS Portal 中間層へのイメージをデプロイし、Apache Struts のイメージ・リンクを使用して関連リンク（アプリケーションではなくポータルに関連付けられた）を生成できます。

注意： OracleAS PDK は Oracle JDeveloper に登録できるため、Oracle JDeveloper のコンポーネント・パレットからタグを削除できます。詳細は、Oracle JDeveloper オンライン・ヘルプで JDeveloper のカスタム・タグ・ライブラリの登録に関する項を参照してください。

ポートレットの作成 Struts ポートレットは、手動または Java ポートレット・ウィザードを使用して作成できます。ウィザードでは明示的に Struts サポートを提供していませんが、ウィザードを使用して Struts ポートレットを作成できます。

ポートレットを作成する手順は、次のとおりです。

1. Oracle JDeveloper で、OracleAS PDK の Java ポートレット・ウィザードを開きます。

注意： ウィザードの使用方法は、5-36 ページの「[ポートレットとプロバイダの作成](#)」を参照してください。

2. 表示ページの実装スタイルは、「**Java クラス**」を選択します。
3. 「パッケージ名」には、`oracle.portal.provider.v2.render.http` と入力します。
4. クラス名には、`StrutsRenderer` と入力します。
5. Java ポートレット・ウィザードでは、ポートレットのレンダラ・クラスのスケルトンとして `StrutsRenderer` を生成します。`StrutsRenderer` は PDK に含まれているため、この生成されたファイルは必要ありません。したがって、ウィザードが完了した後は、ウィザードで生成されたファイルを削除する必要があります。これを行うには、「システム・ナビゲータ」ウィンドウでファイルをクリックし、Oracle JDeveloper の「ファイル」メニューで「**ディスクから削除**」オプションを選択します。

6. provider.xml を編集して、次のプロパティを変更します。

プロバイダ・レベルで次の変更を行います。

- セッション処理を有効にします。これは、Struts アプリケーションでセッション管理を使用するためです。

```
<session>true</session>
```

- URL パラメータの引渡しを有効にします。これによって、フォーム送信されたすべてのパラメータを Struts コントローラに渡すことができます。

```
<passAllUrlParams>true</passAllUrlParams>
```

ポートレット・レベルで次の変更を行います。

- ポートレットがコールされたときに起動する最初のアクションを指定します。次のコードを使用します。

```
<showPage class="oracle.portal.provider.v2.render.http.StrutsRenderer">  
<defaultAction>/portal/prepareNewBlog.do</defaultAction>  
</showPage>
```

ポートレットを拡張して Portal のビジネス・ロジックを追加 アプリケーションで、ユーザーの情報、パーソナライズ、ローカライゼーションなど、ポータル固有のコードを追加する必要があります。これを行うために、Portal のコンテキストでのみコールでき、Portal 固有のビジネス・ロジックを処理する新しい Action クラスを作成できます。

プロバイダの登録

ポートレットを OracleAS Portal で使用する準備が完了すると、そのポートレットを OracleAS Portal に登録してアクセス可能にする必要があります。プロバイダを登録するとき、プロバイダの「ログイン周期」を「ユーザー・セッションごとに1回」に指定して、セッション情報が Struts ポートレットに渡されることを確認してください。

まとめ

Oracle Application Server を使用すると、Oracle JDeveloper を使用して Struts ポートレットを容易に作成でき、既存の Struts アプリケーションを OracleAS Portal に公開できます。Oracle JDeveloper の Java ポートレット・ウィザードの使用方法は、この章の前半を参照してください。OracleAS Portal の使用方法は、『Oracle Application Server Portal ユーザーズ・ガイド』および OracleAS Portal オンライン・ヘルプを参照してください。

第 IV 部

付録

第 IV 部の内容は次のとおりです。

- [付録 A 「ポートレット・ビルダーを使用したポートレットの構築」](#)

ポートレット・ビルダーを使用した ポートレットの構築

コンポーネントを迅速に構築する手段を必要とする開発者のために、OracleAS Portal にはポートレット・ビルダーが用意されています。ポートレット・ビルダーのウィザードでは、ポートレットを作成する各手順をガイドします。ウィザードを使用することで、ビジネス開発の初心者もすぐにポートレットの作成を開始できます。

いくつかの基本的な質問に回答した後、ウィザードを続行するか、すでに収集されている情報に基づいてポートレットを作成するかを決定できます。十分な情報が収集されると「完了」ボタンが表示され、ウィザードの最後のページまで移動しなくてもポートレットを作成できます。他のユーザーに対してポートレットを実行する準備が整った際に必要な作業は、ポートレットのアクセス権限を割り当て、ポートレットを公開し、そのポートレットをページに追加することのみです。

この章では、ウィザードを使用してポートレットを作成する方法、および異なる種類のポートレットを作成、編集、管理および実行する手順について説明します。この章の内容は、次のとおりです。

- [ウィザードによるポートレットの構築](#)
- [ポートレット・ビルダーのコンポーネントの編集](#)
- [ポートレットの管理](#)
- [バージョンの管理](#)
- [ポートレットのセキュリティ管理](#)
- [ポートレットに対するテスト・ランの実施](#)
- [OracleAS Portal スキーマの参照](#)
- [追加機能のコーディング](#)
- [共有コンポーネントによる外観の作成](#)
- [例: チャートおよびレポートの構築](#)

ウィザードによるポートレットの構築

ポートレットは、プロバイダの所有権で構築する必要があります。これは、ポートレットを構築するには、ポートレットをホスティング（所有または収容）するプロバイダを最初に作成（または既存のプロバイダを選択）する必要があることを意味します。新規プロバイダを作成する際は、その前にプロバイダを構築するスキーマを作成することが必要な場合があります。

OracleAS Portal を使用して作成するポートレットはすべてプロバイダの所有権でポートレットを構築する必要がありますが、ポートレットを構築するたびにプロバイダを構築する必要はありません。実際には、既存のプロバイダを使用するか、またはローカルで構築されたポートレットの所有権の専用プロバイダを作成します。その後、別のプロバイダを再度作成する必要はありません。

1つのプロバイダは、多数のポートレットを管理するコンテナとして使用できます。たとえば、レポート、フォームおよびチャートを含む人事管理アプリケーションなど、論理的なアプリケーションごとに1つのプロバイダを使用することは適切な方法です。

この項では、OracleAS Portal で、スキーマを作成してプロバイダを作成し、そのプロバイダの所有権でポートレットを構築する手順について説明します。この項の内容は、次のとおりです。

- [OracleAS Portal でのスキーマの作成](#)
- [ローカルで構築されたポートレットに対するプロバイダの作成](#)
- [OracleAS Portal のウィザードを使用したポートレットの作成](#)

OracleAS Portal でのスキーマの作成

スキーマとは、データベース・オブジェクト、アプリケーションおよびコンポーネントを格納し、スキーマのデータベース権限を制御できる所有権に基づく Oracle データベース・ユーザー・アカウントです。Portal スキーマを使用してスキーマへのアクセスにパスワードを課し、ローカルで構築されたポートレット専用の表領域を確保して、スキーマが使用する一時表領域を識別します。

スキーマの作成には、次の3つの主なタスクが含まれます。

- [スキーマの作成](#)
- [データベース・オブジェクトに対する権限の付与および取消し](#)
- [1つ以上のロールへのスキーマの登録](#)

次に、これらの各タスクの手順について説明します。

スキーマの作成

OracleAS Portal でスキーマを作成する手順は、次のとおりです。

1. OracleAS Portal にログインします。
2. ページ上部の「ナビゲータ」リンクをクリックします。
3. ナビゲータで、「データベース・オブジェクト」タブをクリックして前面に表示します。
4. ページ上部で、「新規作成 ...」の横にある「スキーマ」リンクをクリックします (図 A-1)。

図 A-1 Portal ナビゲータの「データベース・オブジェクト」タブの「スキーマ」リンク



5. 「スキーマの作成」ページ (図 A-2) で、「スキーマ」フィールドにスキーマを識別するための名前を入力します。

名前はポータル内で一意にする必要があります。名前に空白および特殊文字は使用できません。名前に空白を追加するには、アンダースコアを入力してください。たとえば、PORTLETS_SCHEMA ではなく PORTLETS_SCHEMA というスキーマ名を指定できます。また、PORTLETS*SCHEMA および PORTLETS%SCHEMA というスキーマ名は指定できません。

図 A-2 「スキーマの作成」ページ

スキーマの作成

スキーマの作成

スキーマのユーザー名とパスワードを入力してください。新しいスキーマがオブジェクトの格納と一時操作の実行に使用する表領域を選択してください。Portalユーザーがマップできるデータベース・スキーマのリストにこのスキーマを追加するには、「このスキーマをPortalユーザーに使用する」をチェックします。

スキーマ デフォルト表領域

パスワード 一時表領域

パスワードの確認 ORACLEプロファイル

このスキーマをPortalユーザーに使用する

新規作成 閉じる

6. 「パスワード」フィールドに、スキーマに割り当てるパスワードを入力します。

文字を入力するたびにアスタリスクが表示されます。ユーザーは、このスキーマとして Oracle データベースにログインする際にこのパスワードを入力します。
7. 「パスワードの確認」フィールドにパスワードを再度入力します。
8. 「デフォルト表領域」リストからデフォルト表領域を選択します。

この領域は、スキーマによって作成されるデータベース・オブジェクトを格納するために使用されます。ローカルで構築されたポートレット・アプリケーションを格納する専用の目的で作成された表領域を選択することをお勧めします。必要な場合は、データベース管理者と相談してください。
9. 「一時表領域」リストから一時表領域を選択します。

この領域は、表の行をソートするなど、スキーマが操作用の一時記憶域を作成するために使用します。
10. 「ORACLE プロファイル」リストから新しいスキーマの Oracle リソース・プロファイルを選択します。

デフォルトのプロファイルは DEFAULT です。このプロファイルを自由に使用してください。新しいプロファイルの作成を計画している場合は、Oracle SQL コマンドを使用して作成する必要があります。詳細は、Oracle データベースのドキュメントを参照してください。ドキュメントは、Oracle Technology Network Japan (<http://otn.oracle.co.jp>) から入手できます。
11. 管理目的で、ポータル・ユーザーがマップできるデータベース・スキーマのリストに作成中のスキーマを追加するには、「このスキーマを Portal ユーザーに使用する」チェック・ボックスを選択します。

OracleAS Portal ユーザーはすべて、データベース・スキーマに関連付けられている必要があります。デフォルトでは、このスキーマの名前は `<portal>_public` です。<portal> には、OracleAS Portal がインストールされているスキーマの名前が入ります。このチェック・ボックスを選択すると、ここで作成しているスキーマにポータル・ユーザーをマップできます。
12. 変更内容を保存して Portal ナビゲータに戻るには、「新規作成」をクリックします。

あるいは、新しいスキーマのプロパティを編集する場合は、ページ上部に表示される新しいスキーマへのリンクをクリックします。詳細は、A-5 ページの「データベース・オブジェクトに対する権限の付与および取消し」および A-8 ページの「1 つ以上のロールへのスキーマの登録」を参照してください。

データベース・オブジェクトに対する権限の付与および取消し

このスキーマに関連付けられているいくつかのデータベース・オブジェクトに対して、ユーザーに付与されているアクセス権限を拡大または制限することが必要になる場合があります。ユーザーがこのスキーマにログインすると、ユーザーのアクセス権限は、このスキーマが属しているロールによって影響を受けます（A-8 ページの「[1つ以上のロールへのスキーマの登録](#)」を参照）。権限付与を使用すると、特定のデータベース・オブジェクトを選択して、アクセスをさらに制限または拡大する権限を指定できます。

たとえば、BASIC_USER ロールがあると仮定します。このロールは、データベース内のすべての表について SELECT を実行できます。一部の表に対して選択を制限する必要性が発生しました。この場合、権限付与を使用して1つ以上のデータベース・オブジェクト（特定の表、ビューなど）を選択し、SELECT 権限を削除できます。反対に、1つ以上のデータベース・オブジェクトに対してスキーマが保持する必要のあるすべてのアクセス権限がロールに含まれていない場合は、権限付与を使用してアクセス権限を拡大できます。

注意： OracleAS Portal では、データベース・オブジェクトに対する SELECT 権限のみがデフォルトで許可されています。他の権限は、明示的に付与する必要があります。

この項では、スキーマに関連付けられているオブジェクトに対する権限をさらに制御する方法について説明します。

注意： 「スキーマの作成」 ページの上部に表示された「スキーマの編集」リンクをクリックしてこのタスクに至った場合は、手順 6 に進んでください。

スキーマに関連付けられているデータベース・オブジェクトに権限を付与する手順は、次のとおりです。

1. OracleAS Portal にログインします。
2. ページ上部の「ナビゲータ」リンクをクリックします。
3. 「データベース・オブジェクト」タブをクリックして前面に表示します。
4. 「データベース・オブジェクト」タブで、「名前」列に移動して該当するスキーマを検索し、横にある「編集」リンクをクリックします。

図 A-3 Portal ナビゲータでスキーマの横にある「編集」リンク

Oracle Application Server Portal ナビゲータ

ホーム ビルダー ナビゲータ ヘルプ
カスタマイズ アカウント情報 ログアウト

ページグループ プロバイダ データベース・オブジェクト

使用可能なスキーマをブラウズし、許可された操作を実行します。

新規作成... [スキーマ](#) 検索: 実行

パス: データベース・スキーマ

タイプ ▲▼	名前 ▲▼	操作	作成者 ▲▼	最終変更 ▲▼ ?
スキーマ	AURORA\$JIS\$UTILITY\$	編集 , 削除 , アクセス権限付与	AURORA\$JIS\$UTILITY\$	2004-03-26
スキーマ	AURORA\$ORB\$UNAUTHENTICATED	編集 , 削除 , アクセス権限付与	AURORA\$ORB\$UNAUTHENTICATED	2004-03-26
スキーマ	CTXSYS	編集 , 削除 , アクセス権限付与	CTXSYS	2004-03-26
スキーマ	DBSNMP	編集 , 削除 , アクセス権限付与	DBSNMP	2004-03-26
スキーマ	DCM	編集 , 削除 , アクセス権限付与	DCM	2004-03-26
スキーマ	DISCOVERER5	編集 , 削除 , アクセス権限付与	DISCOVERER5	2004-03-26
スキーマ	DSGATEWAY	編集 , 削除 , アクセス権限付与	DSGATEWAY	2004-03-26
スキーマ	HR	編集 , 削除 , アクセス権限付与	HR	2004-06-25
スキーマ	INTERNET_APPSERVER_REGISTRY	編集 , 削除 , アクセス権限付与	INTERNET_APPSERVER_REGISTRY	2004-03-26

- 表示されたページで、「**権限付与**」タブをクリックします (図 A-4)。

図 A-4 「権限付与」 タブ



- 「オブジェクト」フィールドに、権限を制御するデータベース・オブジェクトの名前を入力します。
オブジェクトのリストから選択するには、「ブラウズ」アイコンをクリックします。オブジェクト名の前に、そのオブジェクトを所有しているスキーマを付加します。たとえば、SCOTT スキーマにある EMP 表を示すには、SCOTT を EMP に追加します (SCOTT.EMP)。
- 「追加」をクリックして、ページ下部のリストにオブジェクトを追加します。
オブジェクトがオブジェクト・タイプ別にグループ化されます。つまり、表は一連の表で、ビューは一連のビューでリストされます。
- オブジェクトに対するアクセス権限を拡大または制限するオブジェクトの横にあるチェック・ボックスを選択、または選択を解除します。

図 A-5 表オブジェクトの横にあるチェック・ボックス



9. 変更内容を保存するには、「適用」をクリックします。変更内容を保存して Portal ナビゲータに戻るには、「OK」をクリックします。

1 つ以上のロールへのスキーマの登録

必要なレベルのすべてのデータベース・アクセス権限を提供するには、スキーマを 1 つ以上のロールに登録することが必要になります。スーパー・ユーザーとしてのスキーマが、そのスキーマへのパスワード・アクセス権限を持つユーザーの集団から作成されていると考える場合は、そのスキーマに順にログイン・アクセスするすべてのユーザーに対してデータベースへのアクセス権限を提供するために、スキーマを 1 つ以上のロールに登録します。

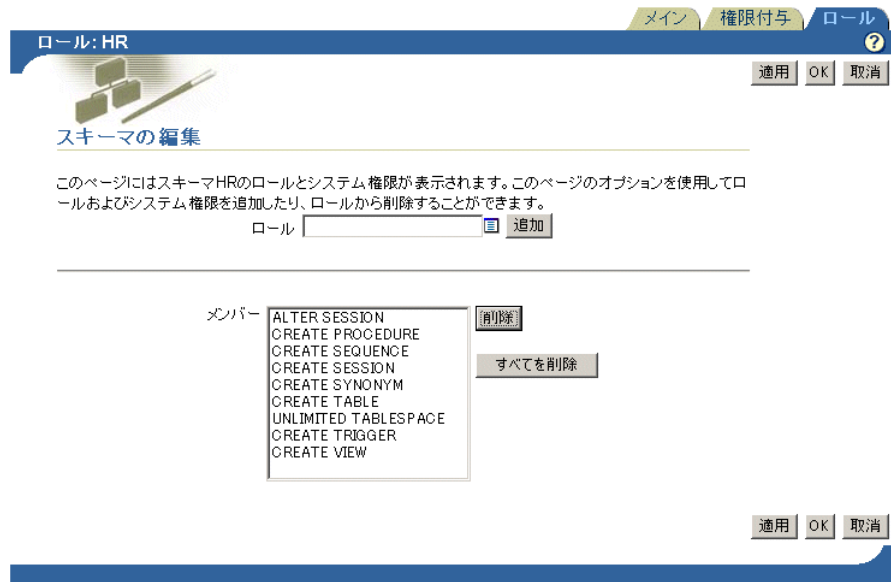
権限付与を使用した特定のデータベース・オブジェクトについては、1 つ以上のロールのスキーマのメンバーシップを介して提供されるアクセス権限を無視できます (A-5 ページの「データベース・オブジェクトに対する権限の付与および取消し」を参照)。

注意：「スキーマの作成」ページの上部に表示された「スキーマの編集」リンクをクリックしてこのタスクに至った場合、またはスキーマの「メイン」タブまたは「権限付与」タブの編集から続ける場合は、手順 6 に進んでください。

1つ以上のロールにスキーマを登録する手順は、次のとおりです。

1. OracleAS Portal にログインします。
2. ページ上部の「ナビゲータ」リンクをクリックします。
3. 「データベース・オブジェクト」タブをクリックして前面に表示します。
4. 「データベース・オブジェクト」タブで、「名前」列に移動して該当するスキーマを検索し、横にある「編集」リンクをクリックします。
5. 表示されたページで、「ロール」タブをクリックして前面に表示します。

図 A-6 「ロール」タブ



デフォルトでは、スキーマは CONNECT ロールに登録されていますが、他のロールに登録することもできます。

6. 「ロール」フィールドの横にある「ブラウズ」アイコンをクリックします。
7. スキーマを登録するロールを選択します。

8. 「**ロール**」フィールドの横にある「**追加**」ボタンをクリックし、選択したロールを「**メンバー**」リストに追加します。

目的のロールすべてにスキーマが登録されるまで、引き続きロールを選択して「**追加**」ボタンをクリックします。

9. 変更内容を保存するには、「**適用**」をクリックします。変更内容を保存して Portal ナビゲータに戻るには、「**OK**」をクリックします。

ローカルで構築されたポートレットに対するプロバイダの作成

プロバイダを所有することになるスキーマを作成、または既存のスキーマを識別した後は、ローカルで構築されたポートレットをホスティング（所有または収容）するプロバイダを作成できます。

ローカルで構築されたポートレットのプロバイダを作成する手順は、次のとおりです。

1. OracleAS Portal にログインします。
2. ページ上部の「**ナビゲータ**」リンクをクリックします。
3. Portal ナビゲータで、「**プロバイダ**」タブをクリックして前面に表示します。
4. 「**プロバイダ**」タブで、「**ローカルで構築されたプロバイダ**」リンクをクリックします。
5. 表示されたページで、「**新規作成 ...**」の横にある「**データベース・プロバイダ**」リンクをクリックします。
6. 「**Portal DB プロバイダ名**」フィールドに、ポータルがこのプロバイダを識別するための内部名を入力します。
空白および特殊文字は使用できません。空白を追加するには、`portlet_provider` のようにアンダースコアを入力します。
7. 「**Portal DB プロバイダの表示名**」フィールドに、ユーザーがこのプロバイダを識別するための表示名を入力します。
8. 「**Portal DB プロバイダのスキーマ**」ドロップダウン・リストからスキーマ（たとえば、先ほど作成したスキーマ）を選択します。
このスキーマのリストには、「**このスキーマを Portal ユーザーに使用する**」チェック・ボックスが選択されているスキーマのみが表示されます。このチェック・ボックスは、「**メイン**」タブにあります。
9. 「**OK**」をクリックして変更内容を保存し、Portal ナビゲータに戻ります。

プロバイダの公開

プロバイダの保護下でポートレットを作成および公開するには、そのプロバイダが OracleAS Portal に対するプロバイダとして識別されている必要があります。これを確認する手順は、次のとおりです。

1. (たとえば、A-10 ページの「ローカルで構築されたポートレットに対するプロバイダの作成」の説明に従って) プロバイダを作成します。
2. Portal ナビゲータのリストで、新しいプロバイダを検索します。
3. 新しいプロバイダの横にある「アクセス権限付与」リンクをクリックします (図 A-7)。

図 A-7 Portal ナビゲータでプロバイダの横にある「アクセス権限付与」リンク

Oracle Application Server Portal ナビゲータ

ホーム ビルダー ナビゲータ ヘルプ
カスタマイズ アカウント情報 ログアウト

ページグループ プロバイダ データベース・オブジェクト

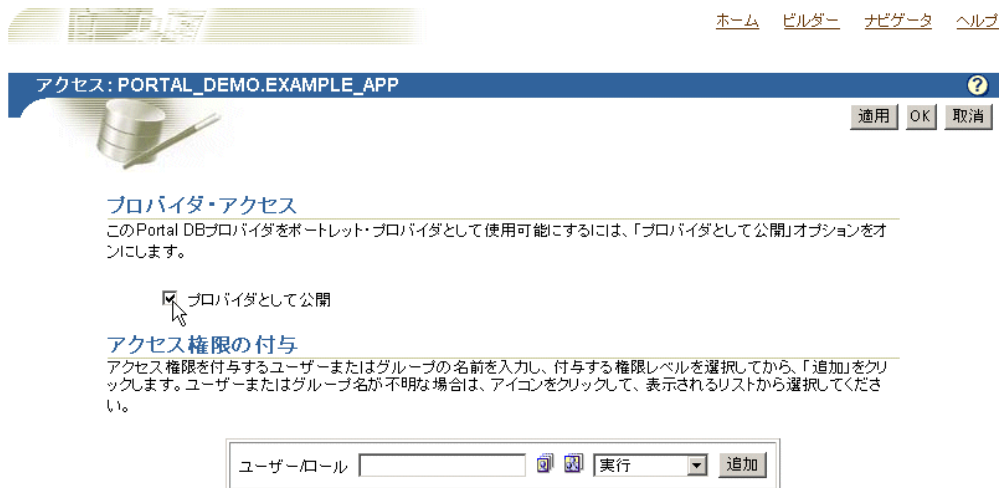
使用可能なローカルで構築されたプロバイダをブラウズします。
新規作成... [Webプロバイダ](#), [データベース・プロバイダ](#) 検索: 実行

パス: [プロバイダ](#) > ローカルで構築されたプロバイダ

タイプ ▲▼	名前 ▲▼	操作	作成者 ▲▼	最終変更 ▲▼ ?
データベース・プロバイダ	Example Application	編集 , 削除 , アクセス権限付与 , エクスポート	PORTAL_DEMO	2004-03-26
データベース・プロバイダ	My company DB provider	編集 , 削除 , アクセス権限付与 , エクスポート	SCOTT	2004-06-25
データベース・プロバイダ	PEOPLE_APP	編集 , 削除 , アクセス権限付与 , エクスポート	PORTAL_DEMO	2004-03-26
データベース・プロバイダ	Portal Application	編集 , 削除 , アクセス権限付与 , エクスポート	PORTAL	2004-03-26
データベース・プロバイダ	Task Application	編集 , 削除 , アクセス権限付与 , エクスポート	PORTAL_DEMO	2004-03-26
共有コンポーネント	共有コンポーネント	アクセス権限の付与 , エクスポート	PORTAL	

4. 「アクセス権限の付与」ページで、「プロバイダとして公開」チェック・ボックスが選択されていることを確認します。

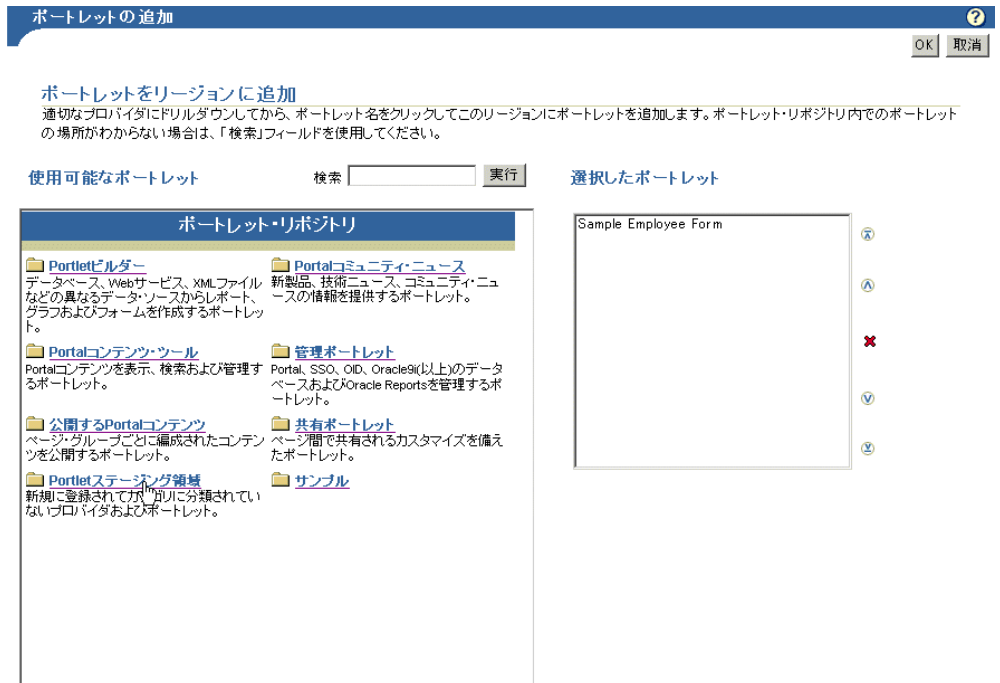
図 A-8 プロバイダの「アクセス権限の付与」ページの「プロバイダとして公開」チェック・ボックス



5. 「OK」をクリックして変更内容を保存し、「アクセス権限の付与」ページを終了します。

このチェック・ボックスを選択すると、このプロバイダの保護下でユーザーがポートレットをポートレット・リポジトリに公開できるようになります。このチェック・ボックスを選択し、ポートレット・ビルダーを使用してポートレットを作成すると、そのポートレットの作成が完了したときにポートレット・リポジトリ（「Portlet ステージング領域」ノード）に自動的に公開されます。

図 A-9 ポートレット・リポジトリの「Portlet ステージング領域」ノード



OracleAS Portal のウィザードを使用したポートレットの作成

この項では、OracleAS Portal で使用可能なポートレット構築ウィザードを使用して構築できるポートレットのタイプをリストし、これらのタイプについて説明します。ポートレットの構築方法に関する一般的な情報やフォーム、レポート、チャートおよび値リストの構築方法に関する特定の情報も提供します。

以前のリリースの OracleAS Portal では、OracleAS Portal 内の他の場所にポートレットを作成した後、そのポートレットを選択してページに組み込むという手順でした。たとえば、あるページにチャート・ポートレットを追加するには、OracleAS Portal の適切な領域にナビゲートし、チャート・ポートレットを定義してポートレット・メタデータ・リポジトリ (PMR) に追加し、ページに戻ってからポートレット・リポジトリにナビゲートし、ページに追加するチャート・ポートレットを選択していました。現在は、これらのポートレットをその場で定義できます。つまり、ポートレット・ビルダーを使用して、ページを移動せずにチャート・ポートレットをページに追加して、完全なポートレットを定義できるようになりました。

この項の内容は、次のとおりです。

- [宣言によるポートレットの構築](#)
- [宣言によるフォームの構築](#)
- [宣言によるレポートの構築](#)
- [interMedia リッチ・コンテンツに対するフォームおよびレポートの構築](#)
- [宣言によるチャートの構築](#)
- [宣言による値リストの構築](#)

ポートレット構築ウィザードを使用して、様々なタイプのポートレットを構築できます。[表 A-1](#) に、構築できるポートレットのタイプをリストし、これらのタイプについて説明します。

表 A-1 ポートレット構築ウィザードで作成可能なポートレットのタイプ

ポートレット・タイプ	説明
フォーム	<p>表の更新、ストアド・プロセスの実行、および他のカスタマイズ済フォームの生成を実行するためのインタフェースとして使用できるカスタマイズ済フォームを表示します。</p> <p>OracleAS Portal のポートレット構築ウィザードでは、次の 3 種類のタイプのフォームを構築できます。</p> <ul style="list-style-type: none"> ■ 「表またはビューをベースとするフォーム」では、ユーザーがデータベース表またはビューのデータを挿入、更新および削除できます。 ■ 「マスター・ディテール・フォーム」は、単一の HTML ページ内にマスター行と複数の詳細行を表示します。このフォームには、2 つのデータベース表またはビューの値を更新するためのフィールドがあります。 ■ 「プロセスをベースとするフォーム」では、データベース・ストアド・プロセスを介して、ユーザーがデータを挿入、更新および削除できます。
レポート	<p>データベース表またはビューで選択したデータをレポート形式で表示します。レポートには、表、フォームまたはカスタムのレイアウトを設定できます。次の 3 種類のタイプのレポートを構築できます。</p> <ul style="list-style-type: none"> ■ 「例による問合せ (QBE) レポート」では、ユーザーが表およびビューのデータを問合せ、挿入、更新および削除できます。QBE レポートの構築ウィザードで、レポートに表示するデータを選択します。または、QBE レポートのカスタマイズ・フォームでエンド・ユーザーが独自の問合せを実行できます。 ■ 「問合せウィザードからのレポート」では、SELECT 文を構成するウィザードを使用して、レポートの構築がガイドされます。 ■ 「SQL 問合せからのレポート」では、手動で構成した SQL 問合せからレポートを構築します。
チャート	データベース表またはビューで選択したデータを棒グラフで表示します。
カレンダー	データベース表またはビューで選択したデータをカレンダー形式で表示します。
動的ページ	動的に生成された HTML コンテンツを Web ページに表示します。
XML コンポーネント	XML ページを表示します。
階層	データベース表またはビューで選択したデータを、3 レベルまでのアイテムのグラフィカルな階層で表示します。

表 A-1 ポートレット構築ウィザードで作成可能なポートレットのタイプ (続き)

ポートレット・タイプ	説明
メニュー	他のメニュー、OracleAS Portal データベース・ポートレットまたは URL にリンクされるオプションが含まれた HTML ベースのメニューを表示します。
URL	ポートレット内の URL ターゲットのコンテンツをレンダリングします。
フレーム・ドライバ	Web ページを 2 つのフレームで表示します。一方のフレームに入力された問合せが、他方のフレームのコンテンツを制御します。
リンク	OracleAS Portal データベース・ポートレットと、他のデータベース・ポートレット、データベース・ポートレットのカスタマイズ・フォームまたは HTML ページとの間のハイパーテキスト・ジャンプを提供するリンクを表示します。
値リスト (LOV)	データベース・ポートレットの作成時に LOV を使用して、入力フィールドに使用できる値を事前に設定します。ユーザーは、値を手動で入力するのではなく、リストから値を選択します。他の LOV をベースにして新しい LOV を構築できます。
データ・コンポーネント	データをスプレッドシート形式で表示します。

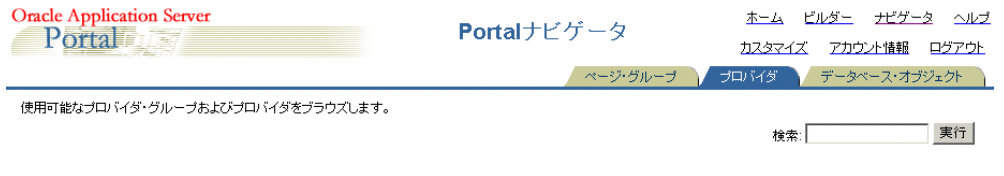
宣言によるポートレットの構築

注意： ポートレットを構築するには、ポートレットを所有するプロバイダに対して少なくとも編集権限が付与されている必要があります。詳細は、A-10 ページの「[ローカルで構築されたポートレットに対するプロバイダの作成](#)」を参照してください。

ウィザードを使用してポートレットを構築する手順は、次のとおりです。

1. OracleAS Portal にログインします。
2. ページ上部の「ナビゲータ」リンクをクリックします。
3. 「プロバイダ」タブをクリックして前面に表示します。
4. 「ローカルで構築されたプロバイダ」リンクをクリックします。

図 A-10 Portal ナビゲータの「ローカルで構築されたプロバイダ」リンク



- パス: [プロバイダ](#)
- | タイプ ▲ ▼ | 名前 ▲ ▼ | 操作 | 作成者 ▲ ▼ | 最終変更 ▲ ▼ ? |
|--------------------------|-----------------|---------------------------------|---------|------------|
| <input type="checkbox"/> | ローカルで構築されたプロバイダ | ローカルで構築されたプロバイダ | | 2004-03-26 |
| <input type="checkbox"/> | 登録されたプロバイダ | 登録されたプロバイダ | | 2004-06-21 |
| <input type="checkbox"/> | プロバイダ・グループ | プロバイダ・グループ | | 2004-06-21 |
- 新しいポートレットをホスティングする OracleAS Portal データベース・プロバイダの名前をクリックします。
権限が付与されているすべてのプロバイダ名が「名前」列に表示されます。
 - 「新規作成 ...」の横にある「チャート」、「フォーム」など、構築するポートレット・タイプのリンクをクリックします (図 A-11)。

図 A-11 Portal ナビゲータで「新規作成 ...」の横にある「フォーム」リンク



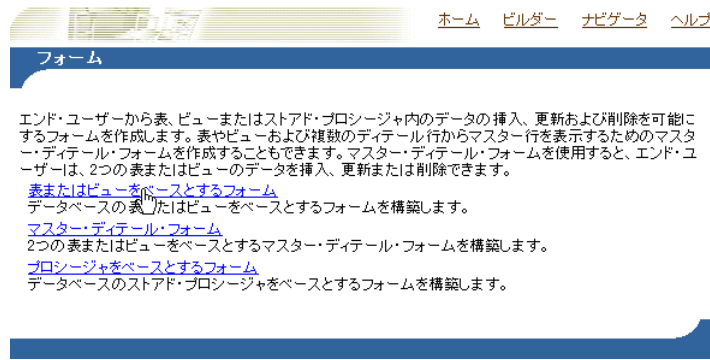
パス: [プロバイダ](#) > [ローカルで構築されたプロバイダ](#) > [Example Application](#)

タイプ ▲ ▼	名前 ▲ ▼	操作	作成者 ▲ ▼	最終変更 ▲ ▼ ?
<input type="checkbox"/>	動的ページ	Dynamic Page		EXAMPLE_APP 2004-03-26
<input type="checkbox"/>	値リスト	EXAMPLE_DEPTNO		EXAMPLE_APP 2004-03-26
<input type="checkbox"/>	値リスト	EXAMPLE_JOB		EXAMPLE_APP 2004-03-26

構築できるポートレットのタイプとその説明については、表 A-1 を参照してください。

構築するポートレットのタイプに応じて、OracleAS Portal により、ポートレット構築ウィザードのステップ 1 またはその他の選択メニューが表示されます。たとえば、「フォーム」をクリックした場合は、表またはビューをベースとするフォーム、マスター・ディテール・フォームまたはプロシージャをベースとするフォームを作成するオプション (図 A-12) があります。

図 A-12 ポートレット・ビルダーでフォームの作成に使用できるオプション



7. ポートレット構築ウィザードのステップ 1 が表示された場合は、各ステップで表示される指示に従います。

ウィザードを完了するために必要な残りのステップ数がプログレス・インジケータに表示されます。ポートレットを構築できる十分な情報が揃った場合は、「完了」ボタンが使用可能になります。指定しなかったフィールドは、デフォルト値が設定されるか、または NULL のままになります。

ウィザードの特定のステップでフィールドを完了する方法が不明な場合は、OracleAS Portal オンライン・ヘルプのアイコンをクリックして詳細を確認してください。

次の項では、特定のタイプのポートレットを構築するために必要な固有のステップについて説明します。詳細は、次の各項を参照してください。

- [宣言によるフォームの構築](#)
- [宣言によるレポートの構築](#)
- [interMedia リッチ・コンテンツに対するフォームおよびレポートの構築](#)
- [宣言によるチャートの構築](#)
- [宣言による値リストの構築](#)

宣言によるフォームの構築

フォームには、わかりやすいインタフェースを介してデータベースの内容を挿入、更新および削除する手段が用意されています。フォーム構築ウィザードで使用可能なオプションを使用して、フォームのレイアウトを制御します。ウィザードのオプションを使用することで、フォーム全体およびフォームの個々の入力フィールドやボタンの外観を制御できます。

フォームのテキスト・フィールドに入力されたユーザー指定の値に対し、JavaScript 妥当性チェックを実行できます。また、ユーザーがフォーム上のボタンをクリックしたときに実行する PL/SQL イベント・ハンドラを追加できます。フォームのコンテンツが正常に送信された後、実行する PL/SQL ブロックまたはプロシージャを指定できます。

OracleAS Portal のポートレット構築ウィザードには、フォームを構築するための 3 種類の構成方式が用意されています。

- 表またはビューをベースとするフォーム
- マスター・ディテール・フォーム
- プロシージャをベースとするフォーム

これらすべてのフォームについては表 A-1 を参照してください。この項では、表またはビューをベースとするフォーム (図 A-13) の作成方法について説明します。

図 A-13 表をベースとするフォームのサンプル

サンプルのEmployeeフォーム カスタマイズ

以下のフォームから新規従業員データを入力してください。

挿入 問合せ リセット

Department

Manager

Employee

Job

Hire Date

挿入 問合せ リセット

データを入力するには「挿入」を、フォームをクリアするには「リセット」をクリックしてください。

表またはビューをベースとするフォームを作成する手順は、次のとおりです。

1. 「宣言によるポートレットの構築」に記載されている手順に従います。
手順 6 を完了した後、この項に戻ります。
2. 「新規作成 ...」の横にある「フォーム」リンクをクリックします (図 A-14)。

図 A-14 Portal ナビゲータで「新規作成 ...」の横にある「フォーム」リンク

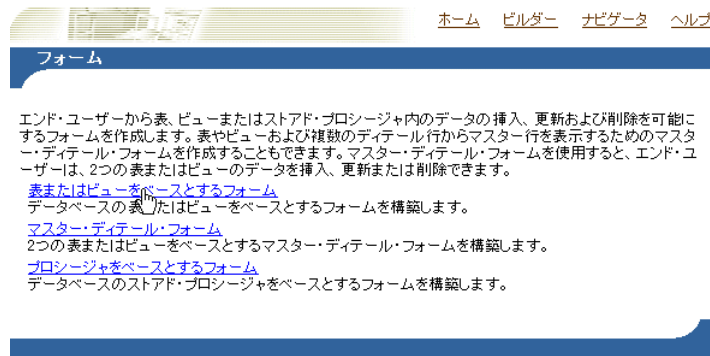


パス: [プロバイダ](#) > [ローカルで構築されたプロバイダ](#) > [Example Application](#)

タイプ ▲▼	名前 ▲▼	操作	作成者 ▲▼	最終変更 ▲▼ ?
動的ページ	Dynamic Page	管理 , 実行 , 編集 , 削除 , アクセス権限付与 , エクスポート	EXAMPLE_APP	2004-03-26
値リスト	EXAMPLE_DEPTNO	管理 , 実行 , 編集 , 削除 , アクセス権限付与 , エクスポート	EXAMPLE_APP	2004-03-26
値リスト	EXAMPLE_JOB	管理 , 実行 , 編集 , 削除 , アクセス権限付与 , エクスポート	EXAMPLE_APP	2004-03-26

- 「フォーム」 ページで、「**表またはビューをベースとするフォーム**」をクリックします。

図 A-15 ポートレット・ビルダーの「表またはビューをベースとするフォーム」リンク



- 「名前」 フィールドに、フォームの内部名を入力します (図 A-16)。

図 A-16 フォーム・ウィザードのステップ 1

FORM_0625164403

ステップ 1 / 8

フォーム名とPortal DBプロバイダ

フォームの名前を入力し、それを所有するPortal DBプロバイダを選択します。このステップで選択できるPortal DBプロバイダは、コンポーネントを構築する権限が付与されているPortal DBプロバイダです。

名前: sample_employee_form

表示名: サンプルのEmployeeフォーム

説明: サンプルのEmployeeフォーム

Portal DBプロバイダ: EXAMPLE_APP

次へ > 取消

内部名はユーザーには公開されません。

5. 「表示名」フィールドに、ユーザーがこのフォームを識別するための名前を入力します (図 A-16)。

表示名は実行時に使用されます。表示名は、フォームのバナーと同様にブラウザ・ウィンドウのタイトルとして表示されます。フォームをポートレットとして公開すると、表示名は、ポートレットのタイトルおよびポートレット・リポジトリの「使用可能なポートレット」リスト内で使用されます。

6. 「説明」フィールドに、このフォームの説明を入力します (図 A-16)。

説明は、ポートレット・リポジトリのポートレットの下に表示されます。説明には、ポートレットの目的の概要、ポートレットのタイプの分類、またはその他の説明的な情報を記載します。開発組織内で、説明に記載する内容について標準的なアプローチを設定すると便利です。

7. 「Portal DB プロバイダ」のリストから、このフォームを所有するプロバイダを選択します (図 A-16)。

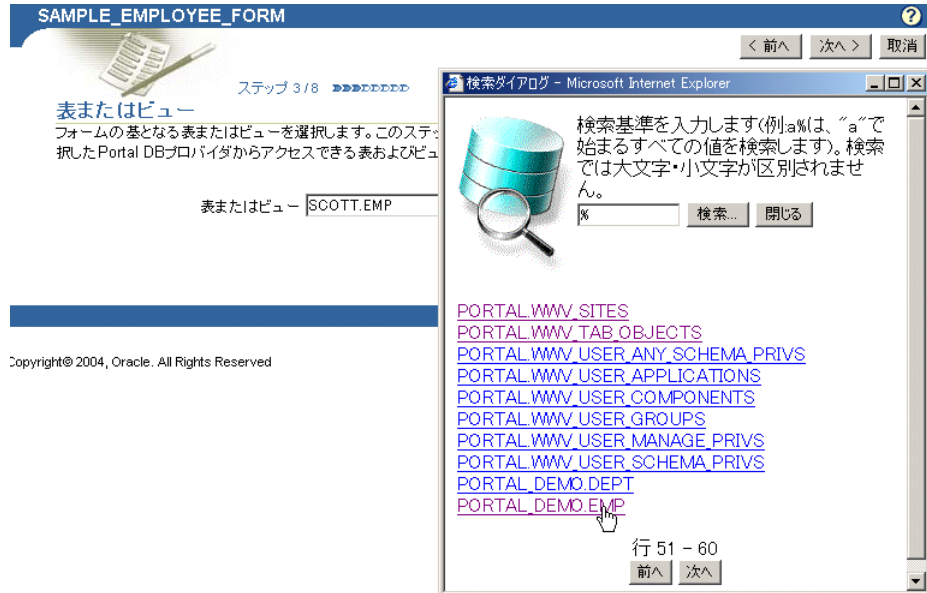
宣言的に構築したポートレットを所有またはホスティングするために作成した専用のプロバイダを使用することをお勧めします。このリストには、ポートレットを構築する権限が付与されているプロバイダのみが表示されます。

8. 「次へ」をクリックします。

9. 「**表またはビュー**」の横にある「**ブラウズ**」アイコンをクリックし、フォームのベースとする表またはビューを選択します (図 A-17)。

このフィールドには、固定値のかわりに定数 (たとえば、#APP_SCHEMA#.EMP、#PORTAL_SCHEMA#.DEPT) を使用できます。詳細は、A-111 ページの「[OracleAS Portal スキーマの参照](#)」を参照してください。

図 A-17 表またはビューの選択

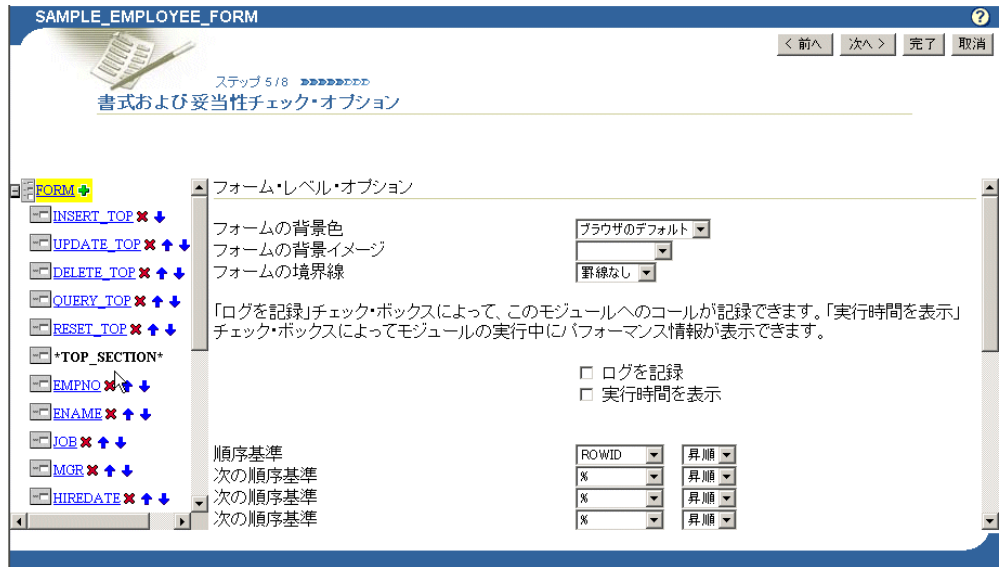


ここで選択できる表およびビューは、前の手順で選択した Portal DB プロバイダからアクセス可能な表およびビューです。フィールドに表名を直接入力することもできます。

10. 「次へ」をクリックします。
11. フォーム・レイアウトを選択します。

「表」または「カスタム」を選択します。表レイアウトは自動的に作成されます。カスタム・レイアウトは、後の手順で指定する HTML コードに基づきます。
12. 「次へ」をクリックします。
13. 「書式および妥当性チェック・オプション」を設定します (図 A-18)。

図 A-18 フォームの書式および妥当性チェック・オプション



- a. 左側の列の「FORM」をクリックし、フォーム全体に対して書式および妥当性チェック・オプションを設定します (図 A-18)。

右側の列に表示される書式および妥当性チェック・オプションは、左側の列で選択したオブジェクトに関するものです。

表 A-2 に、フォーム・レベルの書式および妥当性チェック・オプションをリストし、これらのオプションについて説明します。

表 A-2 フォーム・レベルの書式および妥当性チェック・オプション

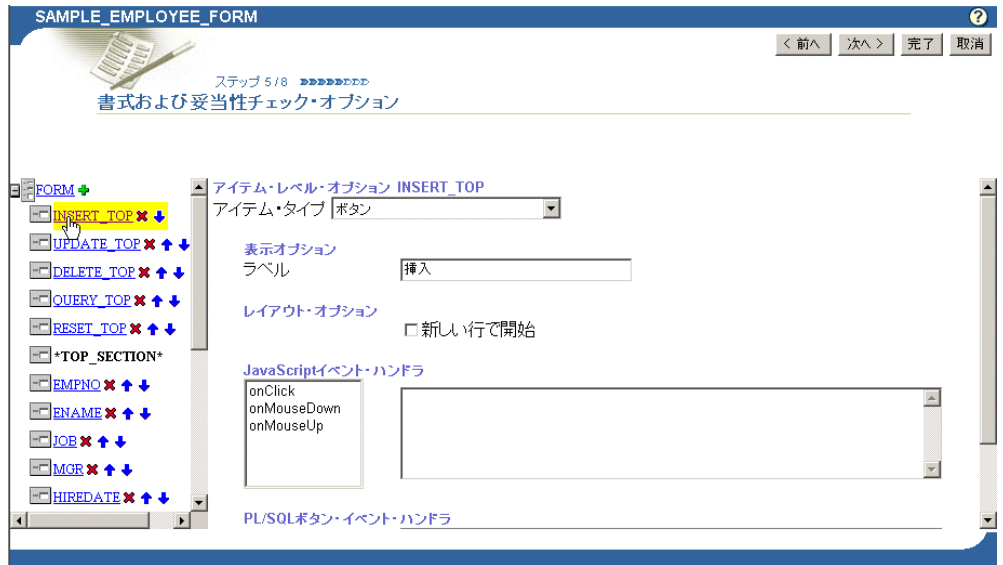
オプション	説明
フォームの背景色	フォームの背景色を選択します。
フォームの背景イメージ	フォームの背景に表示するイメージを選択します。
フォームの境界線	フォームの背景の周囲に配置する境界線のスタイルを選択します。
ログを記録	OracleAS Portal のログにアクティビティを記録する場合は、このチェック・ボックスを選択します。ログ情報には、パフォーマンス統計およびフォームをリクエストするユーザー名が記録されます。
実行時間を表示	フォームに対して HTML を生成するリクエストをサーバーが受け取ってからの経過時間を表示する場合は、このチェック・ボックスを選択します。実行時間はフォームの下部に表示されます。

表 A-2 フォーム・レベルの書式および妥当性チェック・オプション (続き)

オプション	説明
順序基準	SELECT 文で戻されるデータを並べる基準となる列を選択します。並べ替える列を指定しない場合は、パーセント記号 (%) を選択します。
次の順序基準	SELECT 文で戻されるデータを次に並べる基準となる列を選択します。並べ替える列を指定しない場合は、パーセント記号 (%) を選択します。
昇順	列値を並べる方法に従って、「昇順」(a から z、0 から 9 の順) または「降順」(z から a、9 から 0 の順) を選択します。
フォーム送信成功時に 実行する PL/SQL	ユーザーがフォーム上のボタンをクリックした後に実行する、オプションの PL/SQL コードを入力します。ボタンをクリックすることで、フォームのベースとなっている表またはビューに対する操作 (INSERT など) が実行される必要があります。たとえば、表の列が正常に更新されたときに、ユーザーにメッセージが表示されるようにコードを入力できます。

- b. 左側の列で TOP_SECTION アイテム (つまり、INSERT_TOP、UPDATE_TOP など) を選択し、フォームの上部に表示される INSERT、UPDATE などのアイテムについて、右側の列で書式および妥当性チェック・オプションを設定します (図 A-19)。

図 A-19 ボタンの書式および妥当性チェック・オプション



デフォルトでは、挿入、更新、削除、問合せおよびリセットするボタンが各フォームの上部および下部に表示されます。これらのボタンが表示されないようにするには、左側の列のアイテムの横に表示される「削除」アイコンをクリックします。

表 A-3 に、ボタンの書式および妥当性チェック・オプションをリストし、これらのオプションについて説明します。

表 A-3 ボタンの書式および妥当性チェック・オプション

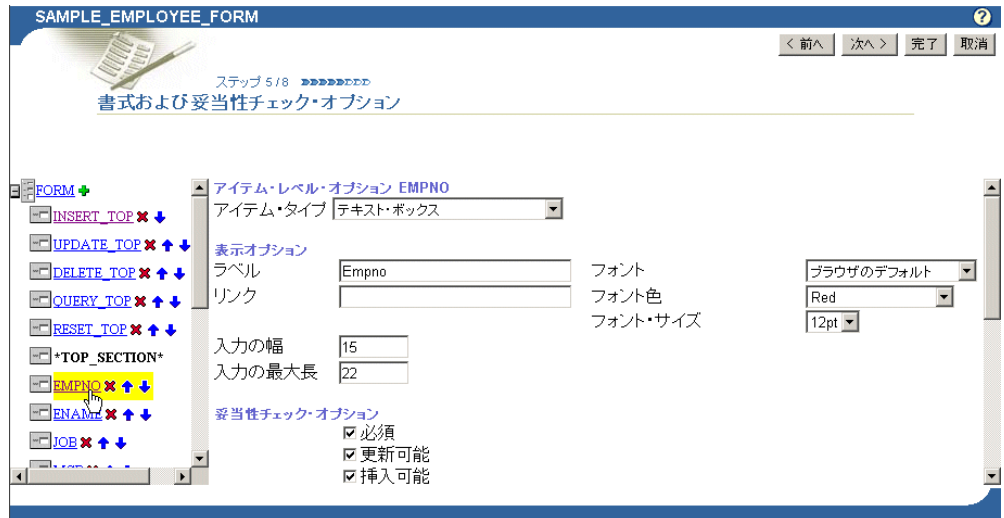
オプション	説明
アイテム・タイプ	<p>選択したアイテム・タイプをフォーム上に表示する方法を選択します。INSERT_TOP などのアイテムのデフォルトはボタンです。INSERT、UPDATE、DELETE、QUERY および RESET では、ほとんどの場合にデフォルトを使用すると考えられます。別のアイテム・タイプを選択した場合は、JavaScript または PL/SQL イベント・ハンドラを介して、アイテムの機能（挿入など）を起動するためのコードを追加する必要があります。アイテム・タイプには、次のものがあります。</p> <ul style="list-style-type: none"> ■ 空白：フィールド間に改行を挿入します。 ■ ボタン：ボタンを挿入します。INSERT、UPDATE、DELETE、QUERY および RESET アイテムには、このページの下にあるイベント・ハンドラを選択する必要があります。 ■ チェック・ボックス：INSERT、UPDATE などの場合は、意味のあるものとするために JavaScript ハンドラを記述する必要があります。列の場合、基礎となる列は TRUE（選択状態）および FALSE（選択解除状態）の入力に対応するために、少なくとも varchar2(5) にする必要があります。 ■ コンボ・ボックス：手動による値の入力を考慮して余分な空白を備えた、値のポップリストを挿入します。選択するには、値リストが定義されている必要があります。「デフォルト値」フィールドに定数を入力し、「デフォルト値タイプ」フィールドでデフォルト値を定数として定義できます。詳細は、A-111 ページの「OracleAS Portal スキーマの参照」を参照してください。 ■ ファイルのアップロード (バイナリ)：BLOB 列の挿入に使用します。 ■ ファイルのアップロード (interMedia)：イメージ、オーディオ・クリップ、ビデオ・クリップなどの interMedia リッチ・コンテンツには、このタイプを選択します。表には、ORDIMAGE、ORDAUDIO または ORDVIDEO タイプの列が必要です。詳細は、A-62 ページの「interMedia リッチ・コンテンツに対するフォームおよびレポートの構築」を参照してください。 ■ 非表示：非表示のフォーム・フィールドを作成します。 ■ 水平定規：<hr> と類似した水平線を引きます。 ■ イメージ：イメージの表示に使用します。 ■ ラベルのみ：プロンプトのみを表示します。 ■ パスワード：パスワード・フィールドとして機能し、入力内容は表示されません。 ■ テキスト領域：複数行のテキストを入力するための大きいテキスト領域を提供します。 ■ テキスト・ボックス：テキスト・ボックスを提供します。

表 A-3 ボタンの書式および妥当性チェック・オプション (続き)

オプション	説明
表示オプション	ラベル: ボタンの横に表示するテキストを入力します。
レイアウト・オプション	このセクションは、「フォーム・レイアウト」ステップで「表」を選択した場合のみ表示されます。「カスタム」を選択した場合は、次のステップで HTML コードを使用して独自の高度なレイアウトを指定できます。 新しい行で開始: フォームの新しい行にアイテムを表示するには、このオプションを選択します。選択が解除されている場合は、前のアイテムまたは列フィールドと同じ行にアイテムが表示されます。
JavaScript イベント・ハンドラ	イベントを選択し、そのイベントが発生したときに実行する処理の JavaScript を入力します。このリストに表示されるイベントの詳細は、JavaScript のドキュメントを参照してください。
PL/SQL ボタン・イベント・ハンドラ	ボタン・イベントを選択し、そのイベントが発生したときに実行する処理の PL/SQL コードを入力します。詳細にカスタマイズする場合は、ドロップダウン・リストから「カスタム」を選択し、カスタム・イベントの PL/SQL コードを入力します。

- c. 左側の列で列名を選択し、フォームに表示する表またはビューの列について、右側の列で書式および妥当性チェック・オプションを設定します (図 A-20)。

図 A-20 列の書式および妥当性チェック・オプション



列が表示されないようにするには、その列の横にある「削除」アイコンをクリックします。

表 A-4 に、表またはビューの列に使用可能な書式および妥当性チェック・オプションをリストし、これらのオプションについて説明します。

表 A-4 表またはビュー列の書式および妥当性チェック・オプション

オプション	説明
アイテム・タイプ	左側で選択した列をフォームに表示する方法を定義します。オプションは、表 A-3 を参照してください。
表示オプション	<p>ラベル: フォーム上のこの列のラベルに表示するテキストを入力します。テキストは単なる表示で、フォーム上でこの表またはビューの列を表すフィールドの横に表示されます。たとえば、EMPNO 列のフィールドの横に Employee Identification Number というラベルを追加できません。</p> <p>リンク: 別の OracleAS Portal ポートレットまたは URL へのリンクを指定します。このオプションを指定すると、フォーム上でラベルがハイパーテキスト・リンクとして表示されます。フォームを別の URL にリンクする場合は、「リンク」テキスト・ボックスに URL の位置を入力します。OracleAS Portal のポートレットにリンクする場合は、次のように、ポートレットが含まれているパッケージ名を入力できます。</p> <p><code>SCHEMA.portlet.SHOW</code></p> <p>SCHEMA はポートレットを所有しているスキーマ名、portlet はポートレット名、SHOW はポートレットを表示するために使用するプロシージャです。また、SHOW_PARMS を指定すると、ポートレットのカスタマイズ・フォームを表示できます。</p> <p>フォント、フォント色、フォント・サイズ: 選択された列に関連付けられているテキストをフォームに表示するためのフォントの特性を指定します。</p> <p>入力の幅: 選択された列に関連付けられているフィールドの文字幅を入力します。</p> <p>入力の最大長: 選択された列に関連付けられているフォーム・フィールドに入力できるデータの最大長を指定します。</p>

表 A-4 表またはビュー列の書式および妥当性チェック・オプション (続き)

オプション	説明
妥当性チェック・オプション	<p>必須: 選択された列に関連付けられているフィールドにユーザーが値を指定してからフォームを送信する必要がある場合は、このチェック・ボックスを選択します。</p> <p>更新可能: ユーザーが列を更新できるようにする場合は、このチェック・ボックスを選択します。選択が解除されている場合は、更新不可になります。この機能は、フォーム上で表の一部の列をユーザーが更新できるようにし、それ以外の列は表示のみにする場合に便利です。たとえば、従業員名と ID 番号については「更新可能」の選択を解除し、従業員の部門については「更新可能」を選択することで、(従業員の異動時など) ユーザーが従業員の部門を更新でき、従業員名と ID 番号は更新できないフォームを作成できます。ユーザーがビューを表示すると、更新可能な列は黒で表示され、更新禁止の列は青で表示され、必須の列は赤で表示されます。</p> <p>挿入可能: ユーザーが新しいレコードを作成する場合に列の値をデータベースに格納する場合は、このチェック・ボックスを選択します。選択が解除されている場合は、値がデータベースに格納されないようになります (つまり、INSERT 文から列が削除されます)。</p> <p>デフォルト値: 選択された列に関連付けられているフィールドのデフォルト値を入力します。ユーザーはこの値をそのまま使用できます。独自の値を指定することもできます。定数、関数、式または SQL 問合せを指定します。</p> <p>デフォルト値タイプ: 前述のフィールド (デフォルト値) に入力したデフォルト値のタイプを指定します。</p> <p>書式マスク: 数値および日付データ型の列に対して Oracle 表示書式を入力します。たとえば、DD/MM/YYYY と入力すると、このパターンに従って日付が表示され、999,999,999.99 と入力すると、このパターンに従ってカンマおよび小数点が配置されます。</p> <p>注意: 日付および数値の書式オプションの詳細は、Oracle データベースのドキュメント (http://otn.oracle.co.jp) を参照してください。</p> <p>フィールド・レベルの妥当性チェック: ユーザーがフィールドに有効な値を入力したかどうかを確認する JavaScript 妥当性チェック・ルーチンを選択します。たとえば、SALARY フィールドに数値が入力されたことを確認する IsNumber という JavaScript ルーチンを選択できます。フィールド妥当性チェック・プロバイダは JavaScript で実装され、OnChange 条件が発生したとき (たとえば、ユーザーがフィールドに値を入力してから [Enter] キーを押したとき) に実行されます。</p>

表 A-4 表またはビュー列の書式および妥当性チェック・オプション (続き)

オプション	説明
	<p>フォーム・レベルの妥当性チェック: ユーザーがフィールドに有効な値を入力したかどうかを確認する JavaScript 妥当性チェック・プロバイダを選択します。フォーム妥当性チェック・プロバイダは、ユーザーが情報を送信したとき (たとえば、フォーム上で「挿入」ボタンをクリックした後) に実行されます。</p>
レイアウト・オプション	<p>このセクションは、「フォーム・レイアウト」ステップで「表」を選択した場合のみ表示されます。「カスタム」を選択した場合は、次のステップで HTML コードを使用して独自の高度なレイアウトを指定できます。</p> <p>新しい行で開始: 選択された列に関連付けられているフィールドの前に改行を作成する場合は、このチェック・ボックスを選択します。選択が解除されている場合は、前の列フィールドと同じ行に列フィールドが表示されます。</p> <p>行範囲: ブラウザ・ページにフィールドを垂直に表示するために使用できる HTML セルの数を入力します。</p> <p>列範囲: ブラウザ・ページにフィールドを水平に表示するために使用できる HTML セルの数を入力します。</p>
JavaScript イベント・ハンドラ	<p>イベントを選択し、そのイベントが発生したときに実行する処理の JavaScript を入力します。</p> <p>リストに表示されるイベントの詳細は、JavaScript のドキュメントを参照してください。</p>

- d. 左側の列で `BOTTOM_SECTION` アイテム (つまり、`PREVIOUS`、`NEXT`、`INSERT_BOTTOM`、`UPDATE_BOTTOM` など) を選択し、フォームの下部に表示される `PREVIOUS`、`NEXT`、`INSERT`、`UPDATE` などのアイテムについて、右側の列で書式および妥当性チェック・オプションを設定します。

詳細は、このタスクの手順 **b** を参照してください。

14. 「次へ」をクリックします。

15. 「カスタム」レイアウトを選択した場合は、次のようにします。

フォームのレイアウトを制御する HTML コードを入力します。このページのテキスト・ボックスには、表を作成する HTML コードのサンプルが含まれています。このモデルを使用すると、このウィザードの「表またはビュー」ステップで選択した各列の書式が設定され、フォーム上の表の行に表示されます。

列ラベルは、サンプル・コードの接尾辞 `.LABEL` で指定されます。列値は、サンプル・コードの接尾辞 `.ITEM` で指定されます。

列値（.ITEM 接尾辞）またはラベル（.LABEL 接尾辞）を変更しないことを前提に、サンプル・コードを独自の HTML で更新できます。ただし、このコードから列値を削除することはできません。

「表」レイアウトを選択した場合は、次のようにします。

フォームの上部または下部に表示する説明文を入力します（図 A-21）。フォームのフォーム・タイトルとヘルプ・テキストを追加できます。フォームが表示されるページの外観を制御するテンプレートも選択できます。

図 A-21 フォームのヘッダー、フッターおよび他のテキストの入力

SAMPLE_EMPLOYEE_FORM

ステップ 7 / 8

フォーム・テキスト

フォームの一番上または一番下に表示する説明文を入力します。フォームのタイトルおよびヘルプ・テキストを追加できます。フォームが表示されるページの外観を制御するテンプレートも選択できます。

テンプレート	PUBLIC.TEMPLATE_3	テンプレートのプレビュー
表示名	Employeeフォームのサンプル	
説明	Employeeフォームのサンプル	
ヘッダー・テキスト	以下のフォームから新規従業員データを入力してください。	
フッター・テキスト	データを入力するには「挿入」を、フォームをクリアするには「リセット」をクリックしてください。	
ヘルプ・テキスト		
情報テキスト		

表 A-5 に、このページで使用可能なオプションをリストし、これらのオプションについて説明します。

表 A-5 表形式のフォームのフォーム・テキストのオプション

オプション	説明
テンプレート	背景色やイメージおよびページの左上隅に表示するイメージなど、フォーム要素の外観を設定するためのテンプレートを選択します。 テンプレートは、フォームがポートレットとして表示されるのではなく、フル・ページ表示で表示される場合のみ使用されます。フォームがポートレットとして表示される場合、外観はホスト・ページのスタイルによって制御されます。
テンプレートのプレビュー	クリックすると、「 テンプレート 」ドロップダウン・リストで現在選択されているテンプレートの外観が表示されます。
表示名	フォームの表示名を編集します。このフィールドには HTML を指定できます。
ヘッダー・テキスト	フォームの上部（ボタンよりも上のタイトルの直下）に表示するフォームの紹介テキストを入力します。 このフィールドには HTML を指定できます。
フッター・テキスト	フォームの下部に表示するテキストを入力します。 このフィールドには HTML を指定できます。
ヘルプ・テキスト	フォームのヘルプ・ページに表示するテキストを入力します。「ヘルプ」ボタンは、フォームに自動的に追加されます。ユーザーがこのボタンをクリックすると、ここに入力したヘルプ・テキストが表示されるページにリンクできます。 このフィールドには HTML を指定できます。

16. 「次へ」をクリックします。

17. 必要に応じて、このフォームを作成する HTML コードの実行中に、様々な時点で実行する PL/SQL を入力します (図 A-22)。

これらのフィールドに入力する PL/SQL では、OracleAS Portal スキーマ、アプリケーション・スキーマおよびアプリケーション名に定数を使用できます。詳細は、A-111 ページの「[OracleAS Portal スキーマの参照](#)」を参照してください。

図 A-22 追加 PL/SQL コードを入力するオプション・フィールド

SAMPLE_EMPLOYEE_FORM

ステップ 8/8

追加PL/SQLコード

フォームの実行時に、様々な箇所で実行されるPL/SQLコードを入力します。

...ページ表示前

...フォーム表示前

...フォーム表示後

...ページ表示後

...フォーム処理前

...フォーム処理後

表 A-6 に、このページのオプションをリストし、これらのオプションについて説明します。

表 A-6 追加 PL/SQL コードのオプション

オプション	説明
... ページ表示前	フォームを生成する HTML の前（つまり、<FORM> タグが生成される前）に実行する PL/SQL プロシージャを入力します。このフィールドの名前とは異なり、PL/SQL プロシージャが実際に実行されるのは、ページ自体が表示された後になります。
... フォーム表示前	<FORM> タグの後、フォーム要素を表示する前に実行する PL/SQL プロシージャを入力します。
... フォーム表示後	すべてのフォーム要素が表示された後、</FORM> 終了タグの前に実行する PL/SQL プロシージャを入力します。

表 A-6 追加 PL/SQL コードのオプション (続き)

オプション	説明
... ページ表示後	</FORM> 終了タグの後に実行する PL/SQL プロシージャを入力します。
... フォーム処理前	フォームが処理される前に実行する PL/SQL プロシージャを入力します。
... フォーム処理後	フォームが処理された後に実行する PL/SQL プロシージャを入力します。

注意: PL/SQL プロシージャからの出力を表示する場合は、次の 4 つのフィールドの 1 つに PL/SQL を入力する必要があります。

- ページ表示前
- フォーム表示前
- フォーム表示後
- ページ表示後

出力が表示されないフォーム処理前およびフォーム処理後のフィールドでは、PL/SQL プロシージャからの出力を表示できません。

また、PL/SQL プロシージャからの出力を表示する場合は、プロシージャでスキーマ名を指定する必要があります。フォームとプロシージャが同じスキーマにある場合でも指定する必要があります。

18. 「完了」をクリックして変更内容を保存します。

「完了」をクリックすると、フォームを管理できるサマリー・ページにジャンプします (図 A-23)。これらのタスクについては、この章の後半で説明します。

図 A-23 フォームのサマリー・ページ

開発: APP_0622135148.SAMPLE_EMPLOYEE_FORM

開発 管理 アクセス ?

閉じる

表のフォーム SAMPLE_EMPLOYEE_FORM
Portal DBプロバイダ APP_0622135148
アーカイブ・バージョン
本稼働バージョン状態 1 (PRODUCTION with VALID Package)
最終更新 火曜日 2004/06/22 15:22:33 by ORCLADMIN
実行リンク PORTAL:www_app_module.new_instance?p_moduleid=1091083698
PL/SQLソース [パッケージ仕様](#), [パッケージ本体](#)
コール・インタフェース [表示](#)

[編集](#) [新規として編集](#) [実行](#) [ポートレットとして実行](#) [カスタマイズ](#) [お気に入り追加](#) [情報](#) [削除](#)

閉じる

19. 「閉じる」をクリックし、Portal ナビゲータに戻ります。

宣言によるレポートの構築

図 A-24 サンプル・レポート

Department Number	Manager	Employee	Job
20	7902 SMITH		CLERK
30	7698 ALLEN		SALESMAN
30	7698 WARD		SALESMAN
20	7839 JONES		MANAGER
30	7698 MARTIN		SALESMAN
30	7839 BLAKE		MANAGER
10	7839 CLARK		MANAGER
20	7566 SCOTT		ANALYST
10	(NULL) KING		PRESIDENT
30	7698 TURNER		SALESMAN
20	7788 ADAMS		CLERK
30	7698 JAMES		CLERK
20	7566 FORD		ANALYST
10	7782 MILLER		CLERK

JOB=PRESIDENTの場合、列データはすべて背景色=黄色で表示されます。

レポートには、表およびビューからのデータが表示されます (図 A-24)。JOIN 条件を使用して、複数の表またはビューに基づいたレポートを作成できます。指定した条件を満たすデータをハイライトできます。たとえば、雇用期間が2年未満のすべての従業員のレコードを赤の太字テキストで表示できます。レポートに表示された値から他の OracleAS Portal データベース・ポートレットまたは URL へのハイパーテキスト・リンクを作成できます。

また、ウィザードを使用して、次の処理を実行するその他のオプションを指定できます。

- レポートに表示されるデータを制限する
- 列の値を合計する
- 論理演算子を使用して列のデータを選択する
- レポートのデータの書式を設定する
- レポートのカスタマイズ・フォームの書式を設定する
- レポートの様々な時点で実行する PL/SQL コードを指定する

OracleAS Portal のポートレット・ビルダー・ウィザードには、レポートを構築するための 3 種類の構成方式が用意されています。

- 例による問合せ (QBE) レポート
- 問合せウィザードからのレポート
- SQL 問合せからのレポート

これらのレポートについては、表 A-1 を参照してください。この項では、問合せウィザードを使用してレポートを構築する方法について説明します。

問合せウィザードを使用してレポートを構築する手順は、次のとおりです。

1. A-16 ページの「宣言によるポートレットの構築」に記載されている手順に従います。
手順 6 を完了した後、この項に戻ります。
2. 「新規作成 ...」の横にある「レポート」リンクをクリックします (図 A-25)。

図 A-25 Portal DB プロバイダの Example Application の「レポート」リンク

これらは、使用可能なポートレットおよび操作です。

新規作成... [フォーム](#), [レポート](#), [チャート](#), [カレンダー](#), [動的ページ](#),
[XMLコンポーネント](#), [階層](#), [メニュー](#), [URL](#), [フレーム](#),
[ドキュメント](#), [リンク](#), [値リスト](#), [データコンポーネント](#)

パス: [プロバイダ](#) > [ローカルで構築されたプロバイダ](#) > [Application 0622135148](#)

3. 「レポート」ページで、「問合せウィザードからのレポート」をクリックします (図 A-26)。

図 A-26 ポートレット・ビルダーの「問合せウィザードからのレポート」リンク

レポート

レポート(エンド・ユーザーが表またはビュー内のデータの問合せを実行できるようにする)、または例による問合せレポート(エンド・ユーザーがデータの問合せおよび管理を実行できるようにする)を作成します。

[例による問合せ\(QBE\)レポート](#)

エンド・ユーザーが表およびビュー内のデータの問合せ、挿入、更新および削除を実行できるようにする QBE レポートを構築します。

[問合せウィザードからのレポート](#)

ウィザードを使用して SQL 問合せを構築します。

[SQL 問合せからのレポート](#)

独自の SQL 問合せを記述します。

4. 「名前」フィールドに、レポートの内部名を入力します (図 A-27)。

図 A-27 「問合せウィザードからのレポート」のステップ 1

RPT_0622155433

ステップ 1 / 16

レポート名とPortal DBプロバイダ

レポートの名前を入力し、それを所有するPortal DBプロバイダを選択します。このステップで選択できるPortal DBプロバイダは、コンポーネントを構築する権限が付与されているPortal DBプロバイダです。

名前: sample_report

表示名: サンプル・レポート

説明: サンプル・レポート

Portal DBプロバイダ: EXAMPLE_APP

内部名はユーザーには公開されません。

5. 「表示名」フィールドに、レポートの表示名を入力します (図 A-27)。

表示名は実行時に使用されます。表示名は、レポートのバナーと同様にブラウザ・ウィンドウのタイトルとして表示されます。レポートをポートレットとして公開すると、表示名は、ポートレットのタイトルおよびポートレット・リポジトリの「使用可能なポートレット」リスト内で使用されます。

6. 「Portal DB プロバイダ」のリストから、このレポートを所有するプロバイダを選択します (図 A-27)。

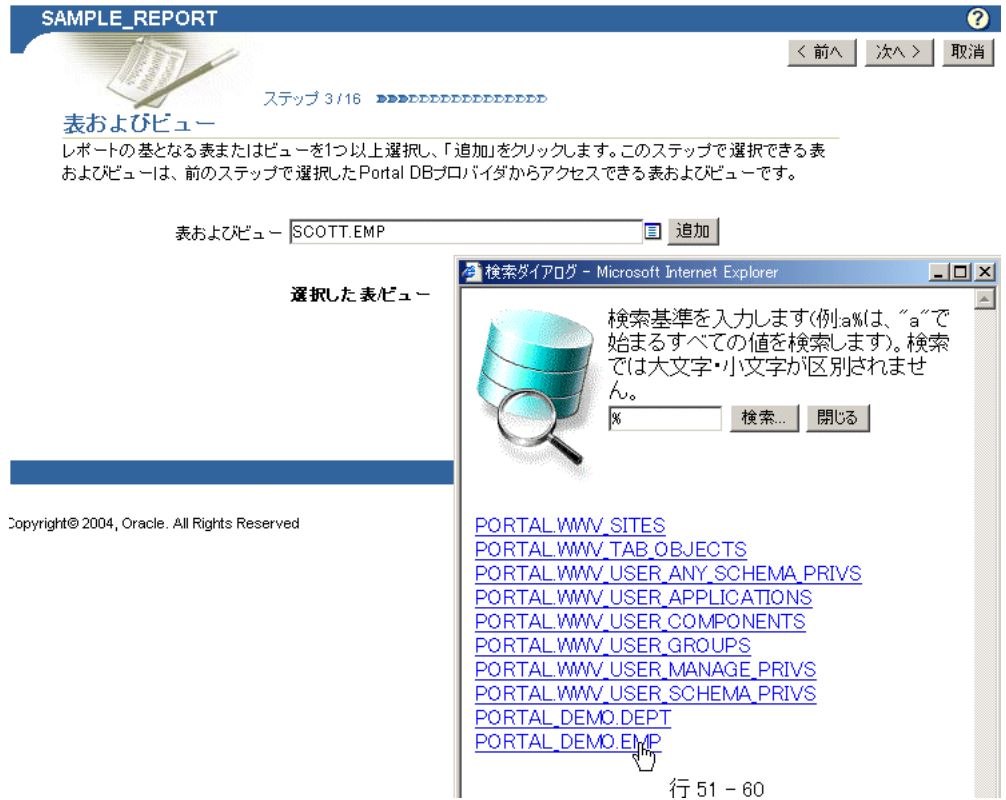
宣言的に構築したポートレットを所有またはホスティングするために作成した専用のプロバイダを使用することをお勧めします。このリストには、ポートレットを構築する権限が付与されているプロバイダのみが表示されます。

7. 「次へ」をクリックします。

8. 「表およびビュー」リストから、このレポートのベースとする表またはビューを1つ以上選択します (図 A-28)。

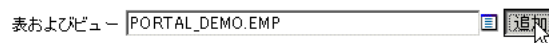
注意: 「表およびビュー」フィールドでは、OracleAS Portal スキーマおよびアプリケーション・スキーマに定数を使用できます (たとえば、#APP_SCHEMA#.EMP を入力できます)。詳細は、A-111 ページの「OracleAS Portal スキーマの参照」を参照してください。

図 A-28 レポートのデータソースの選択



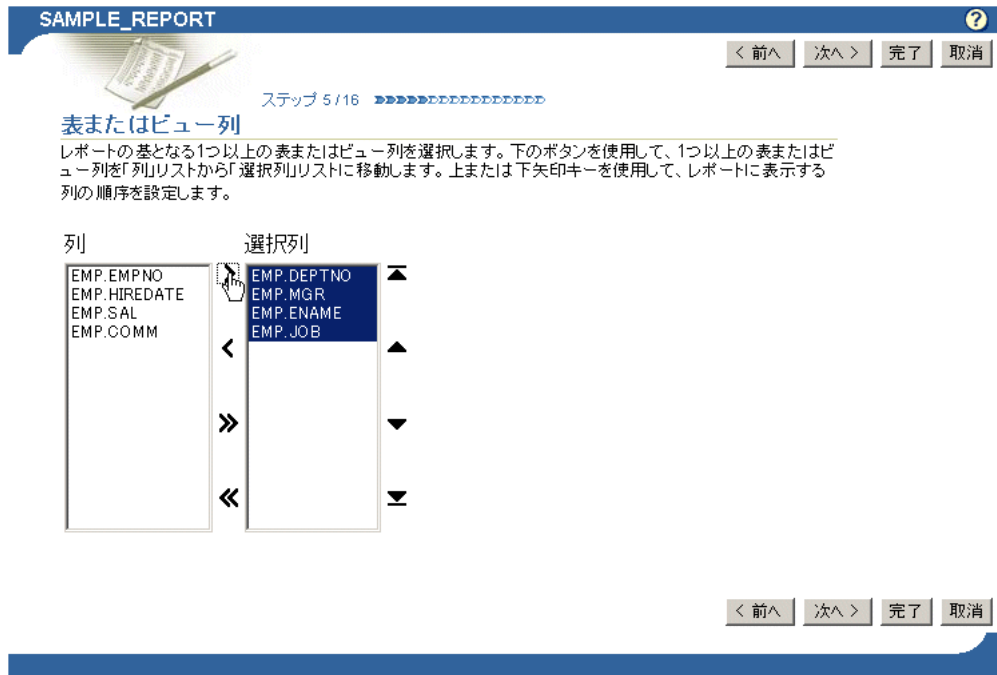
1つ選択するたびに「追加」をクリックします (図 A-29)。

図 A-29 「追加」のクリックによるデータソースの追加



9. 「次へ」をクリックします。
10. 複数の表またはビューをレポートのベースとする場合は、相互に結合する列を識別します。
このページは、前述の手順で複数の列またはビューを選択した場合のみ表示されます。デフォルトの結合条件が自動的に生成されます。これらのデフォルトは、そのまま使用または変更できます。
11. 「次へ」をクリックします。
12. レポートに表示する列を選択します (図 A-30)。

図 A-30 レポートに表示する列の選択



2つの列の間の矢印をクリックして、一方の列から他方の列に個々の選択肢またはリスト全体を移動します。右側の列の右にある再配置アイコンを使用して、選択した列の表示順序を設定します。選択できる列は最大 255 列です。

13. 「次へ」をクリックします。
14. 必要に応じて、レポートに表示されるデータを制限する条件を指定します (図 A-31)。

図 A-31 列条件の設定

SAMPLE_REPORT

ステップ 6 / 16

列条件

レポートに表示される列データを絞り込む条件を選択します。

列名	条件	値
EMP.SAL	>	200

追加...

表 A-7 に、このページのオプションをリストし、これらのオプションについて説明します。

表 A-7 「列条件」 ページのオプション

オプション	説明
列名	<p>レポートのデータに対して条件を指定する場合は、列を選択します。列は、表示対象として選択していない列でもかまいません。たとえば、給与が 200 ドルを超えるなど、指定した給与条件を満たす従業員のみを表示するようにデータを制限するには、表示対象として選択していない給与列を選択できます。この条件を設定するには、たとえば、「列名」で SAL を選択し、「条件」で > を選択し、「値」に 200 を入力します。</p>
条件	<p>「列名」と「値」との関係を定義する条件を選択します。オプションは次のとおりです。</p> <ul style="list-style-type: none"> ■ 等しい (=) ■ より大きい (>) ■ 以上 (>=) ■ より小さい (<) ■ 以下 (<=) ■ LIKE ■ 等しくない (!=) ■ NULL ■ NOT NULL ■ IN ■ NOT IN <p>IN または NOT IN 条件の後に複数の値を指定するには、10:20:30 のように各値の間にコロン (:) を入力します。</p>
値	<p>指定された列の中で、レポートに表示する行を制限するための値を入力します。</p>
追加 ...	<p>クリックすると、このページに条件を追加するためのフィールドがさらに表示されます。</p>

15. 「次へ」をクリックします。

16. レポート・レイアウト・タイプを「表」、「フォーム」または「カスタム」の3種類から選択します。
- 「表」レイアウトは、レポート構築ウィザードで選択するオプションに基づきます。結果が表形式で表示されます。各レコードは、基礎となる表またはビューの行を表します。各データ列は表の列として表示されます。
 - 「フォーム」レイアウトは、レポート構築ウィザードで選択するオプションに基づきます。結果が行に表示されます。各データ列は個別の行に表示されます。
 - 「カスタム」レイアウトは、ウィザードの後のステップで指定する HTML コードに基づきます。独自の HTML コードを指定するため、構造化された（表またはフォーム）レイアウトを使用する場合よりも、レポートの外観を高度に制御できます。
- ここでは、表レイアウトを選択したと仮定して説明します。表レポートの構成には、他のレイアウト・タイプでは実行することになるウィザードの一部のステップがありません。たとえば、カスタム・レイアウトでは、カスタム HTML コードを指定するステップがあります。このステップは、表レイアウトを選択した場合には表示されません。
17. 「次へ」をクリックします。
18. 列の表示書式を指定します (図 A-32)。

図 A-32 「列の表示書式」 ページ

列	列ヘッダー	合計	データ型	文字位置	表示内容	書式マスク	リンク	リンクの編集	幅のタイプ	幅
EMP.DEPTNO	Department Number	<input type="checkbox"/>	?99	右揃え	テキスト		%		ピクセル	
EMP.MGR	Manager	<input type="checkbox"/>	?99	右揃え	テキスト		%		ピクセル	
EMP.ENAME	Employee	<input type="checkbox"/>	A	左揃え	テキスト		%		ピクセル	
EMP.JOB	Job	<input type="checkbox"/>	A	左揃え	テキスト		%		ピクセル	

表 A-8 に、このページのオプションをリストし、これらのオプションについて説明します。

表 A-8 表レポートに対する列の表示書式オプション

オプション	説明
列	ウィザードの前のステップで選択した表またはビューの列名を表示します。
列ヘッダー	レポートでこの列を識別するために使用する名前を入力します。たとえば、表に EMPNO 列がある場合は、列の実際の名前かわりに Employee ID Number と入力できます。
合計	値を合計して結果をレポートに表示する場合は、このチェック・ボックスを選択します。
データ型	対応する列のデータの型を表示します。
文字位置	レポート列のデータを揃えるマージン（左、中央または右）を選択します。デフォルトでは、数値データは右揃え、アルファベット・データは左揃えです。
表示内容	<p>オプションは次のとおりです。</p> <ul style="list-style-type: none"> ■ テキスト: 対応する HTML タグを解析せずに、レポートに値を表示します。たとえば、<i>Name</i> は <i>Name</i> と表示されます。 ■ HTML: 対応する HTML タグを使用してレポートに値を表示します。たとえば、<i>Name</i> は Name と表示されます。 ■ 非表示: レポートに値を表示しません。たとえば、リンク・パラメータをレポートに表示する必要がない場合は、その値が含まれている列に対して「非表示」を選択します。
書式マスク	日付および数値データ型の列に対して Oracle 表示書式を入力します。日付と数値の書式マスクの例は、表 A-9 および表 A-10 を参照してください。
リンク	列から別の OracleAS Portal のポートレットへのリンクを作成します。列値はハイパーテキスト・リンクとして表示されます。この列に使用するリンクをあらかじめ作成し、データベースに格納されている場合のみ、リンクを指定できます。
リンクの編集	クリックすると、「リンク・パラメータの設定」ウィンドウが表示されます。このウィンドウで、ターゲット・ポートレットに渡すリンク・パラメータの値を設定できます。10 または KING などの静的な値を入力するか、またはこのウィザードの「表またはビュー列」ステップで識別した列のリストから選択します。リンク・パラメータには列の値が必要です。その列をレポートに表示しないようにする場合は、「表示内容」の値を「非表示」に設定します。

表 A-8 表レポートに対する列の表示書式オプション (続き)

オプション	説明
幅のタイプ	次のいずれかを指定します。 <ul style="list-style-type: none"> ■ 文字: 1行当たりの指定文字数で出力を表示します。たとえば、「幅」に 20 と入力すると、レポートは各行 20 文字の列データで表示されます。文字数が指定した数を超えると、残りの文字は次の行に折り返されます。ただし、出力が ASCII フォーマットの場合、「文字」は列の幅が基準となります。 ■ ピクセル: 1行当たりの指定ピクセル数で出力を表示します。たとえば、「幅」に 10 と入力すると、列データは 1行当たり 10 ピクセルのデータで表示されます。 ■ パーセント: 表に対して指定のパーセントを占める列で出力を表示します。たとえば、「幅」に 25 と入力すると、列データは表全体の 25 パーセントを占める幅の列で表示されます。
幅	幅のタイプに対応する数値を入力します。

Excel 形式で生成したレポートには、書式マスクを使用できません。表 A-9 および表 A-10 に、日付と数値の書式マスクの例を示します。これらは、このウィザード・ページの「書式マスク」フィールドで使用できます。

表 A-9 日付書式マスクの例

サンプル日付書式	表示される日付
MM/DD/RR	03/04/85
Mon. DD, RRRR	Mar. 4, 1985
Day Month DD fmHH:MI AM	Monday March 4 11:35 PM
Dy Mon ddth fmHH24:MI:SS	Mon Mar 4th 23:35:22
Day "the" ddthsp "of" Month	Monday the fourth of March

表 A-10 数値書式マスクの例

サンプル数値書式	数値	表示される数値
-0000	7934	"7934"
	-7934	"-7934"
-00000	7934	"07934"
-NNNN	7639	"7639"
	535	"535"

表 A-11 に、このページのオプションをリストし、これらのオプションについて説明します。

表 A-11 レポート・ウィザードの「書式条件」ページのオプション

オプション	説明
列	条件を適用する列を選択します。列は、ウィザードの最初の方のステップで表示対象として選択した列を指定する必要があります。
条件	「列」と「値」との関係を定義する条件を選択します。オプションは次のとおりです。 <ul style="list-style-type: none"> ■ 等しい (=) ■ より大きい (>) ■ 以上 (>=) ■ より小さい (<) ■ 以下 (<=) ■ LIKE ■ 等しくない (!=) ■ NULL ■ NOT NULL ■ IN ■ NOT IN
値	条件をトリガーする列値を入力します。数値に書式マスクを適用する必要はありません。たとえば、表に EMPNO 列があり、3000 より大きい EMPNO 値に対して特別な書式を適用する場合は、「 条件 」で > を選択し、このフィールドに 3000 と入力します。 IN または NOT IN 条件の後に複数の値を指定するには、10:20:30 のように各値の間にコロン (:) を入力します。
行 / 列	条件付書式を行全体に適用するか、特定の列データのみ適用するかを選択します。条件を指定する場合は、ここで <ROW> または列名を選択する必要があります。
色	指定の条件を満たすデータを表示するテキストの色を選択します。
背景色	指定の条件を満たすデータが含まれている列または行の背景色を選択します。
フォント	指定の条件を満たすデータを表示するフォントを選択します。
A	指定の条件を満たす結果を太字で表示する場合は、このチェック・ボックスを選択します。

表 A-11 レポート・ウィザードの「書式条件」ページのオプション（続き）

オプション	説明
A	指定の条件を満たす結果をイタリックで表示する場合は、このチェック・ボックスを選択します。
<u>A</u>	指定の条件を満たす結果にアンダスコアを表示する場合は、このチェック・ボックスを選択します。
点減	指定の条件を満たす結果を点減させる場合は、このチェック・ボックスを選択します。
順序	連番を入力し、書式設定を実行する優先順位を設定します。たとえば、1つの行に複数の条件を満たすデータが含まれており、それぞれの書式ルールが異なる場合に役立ちます。順序が最も高い（つまり、1）条件が、他のすべてよりも優先されます。

21. 「次へ」をクリックします。

22. レポートに使用できる様々なビューに対する表示オプションを定義します。

- 表 A-12 に、すべてのビューに共通する表示オプションを示します (図 A-34)。
- 表 A-13 に、フル・ページ・ビューに共通する表示オプションを示します (図 A-35)。
- ポートレット・ビューのオプションは、いくつかの例外を除き、フル・ページ・ビューのオプションと同じです。表 A-13 および図 A-35 を参照してください。
- 表 A-14 に、列ブレイクの表示オプションを示します (図 A-36)。
- 表 A-15 に、行順序の表示オプションを示します (図 A-37)。
- 表 A-16 に、モバイル・デバイスの表示オプションを示します (図 A-38)。

図 A-34 すべてのビューに共通する表示オプション

共通オプション

ポートレット・モードおよびフルWebページに表示されるレポートの外観に影響する共通オプションを選択します。

総行数の表示 NULL値を置き換えて表示

レポートにinterMediaリッチ・コンテンツを埋め込む

有効期限までの時間(分)

デフォルト書式

表 A-12 すべてのビューに共通するレポート表示オプション

オプション	説明
総行数の表示	レポートに総行数を表示する場合は、このチェック・ボックスを選択します。
NULL 値を置き換えて表示	たとえば、(Null) など、レポート内のすべての NULL 値に対して表示するテキスト文字列を入力します。
レポートに <i>interMedia</i> リッチ・コンテンツを埋め込む	オーディオ、ビデオ、イメージなど、 <i>interMedia</i> データ値を表示するレポート出力を生成する場合は、このチェック・ボックスを選択します。 詳細は、A-62 ページの「 interMedia リッチ・コンテンツに対するフォームおよびレポートの構築 」を参照してください。
有効期限までの時間 (分)	最初に表示した後、キャッシュにレポートを保持する時間 (分) を入力します。期限が切れると、レポート・データはデータベースからリフレッシュされます。
デフォルト書式	レポートの表示書式を選択します。 <ul style="list-style-type: none"> ■ HTML: HTML 表を使用してレポートを書式設定し、Web ブラウザの新しいページに出力を表示します。大量のデータが含まれているポートレットは、この書式で表示すると時間がかかる可能性があります。 ■ Excel: Microsoft Excel で表示するようにレポートを書式設定します。完成したレポートを実行する場合、ローカルの Excel 設定では、最初にレポートをローカルに保存しないと、レポートを開いて表示できないことがあります。 ■ ASCII: HTML PRE タグを使用してレポートを書式設定し、レポートのヘッダーおよび値を ASCII テキストとして表示します。このオプションは、大量のデータを表示する場合に便利です。

図 A-35 フル Web ページ・ビューとポートレット・ビューのレポート表示オプション

フル・ページ・オプション

フル Web ページに表示されるレポートの外観に影響するオプションを選択します。

	フォント	フォント色	サイズ
ヘッダー	Tahoma		12pt
行テキスト	Tahoma		12pt
ヘッダーの背景色		表の行の色	Aquamarine Black
境界線	罫線なし	最大行数/ページ	20
<input type="checkbox"/> 行間の線を表示	<input type="checkbox"/> ログを記録	<input type="checkbox"/> 実行時間を表示	<input type="checkbox"/> 問合せ条件の表示

ポートレット・オプション

ポートレット・モードに表示されるレポートの外観に影響するオプションを選択します。

	ページ・スタイル	フォント	フォント色	サイズ
ヘッダー	ポートレット・ヘッダー1	<デフォルト>	<デフォルト>	<デフォルト>
行テキスト	ポートレット・テキスト1	<デフォルト>	<デフォルト>	<デフォルト>
注意: フォント名、色、およびサイズはページ・スタイルの設定よりも優先されます。				
ヘッダーの背景色		表の行の色	Aquamarine Black	
境界線	罫線なし	最大行数/ページ	20	

表 A-13 フル Web ページ・ビューとポートレット・ビューのレポート表示オプション

オプション	説明
フォント	レポート・テキストのフォントを指定します。
フォント・サイズ	レポート・テキストのフォント・サイズを指定します。+1、+2 など、相対値を使用します。 相対的なフォント・サイズは、ページに対して HTML コードに指定した最後のフォント・サイズに相対値を加えたサイズになります。たとえば、HTML コードに指定した最後のフォント・サイズが 14 ポイントの場合、新しいヘッダーを +2 のフォント・サイズでコード化すると、ヘッダーは 16 ポイントのタイプで表示されます。
境界線	レポートの周囲に表示する罫線の太さを指定します。「罫線なし」、「細い罫線」または「太い罫線」を選択します。
フォント色	レポート・テキストのフォント色を指定します。
ヘッダーの背景色	レポートの列ヘッダーの背景色を指定します。
表の行の色	レポート行の背景色を指定します。複数の色を選択するには、[Ctrl] キーを押しながらクリックします (Windows の場合)。行には色が交互に表示されます。

表 A-13 フル Web ページ・ビューとポートレット・ビューのレポート表示オプション (続き)

オプション	説明
最大行数 / ページ	レポートに表示する最大行数を入力します。
行間の線を表示	レポートの行間に線を表示する場合は、このチェック・ボックスを選択します。
ログを記録	(フル・ページのみ) パフォーマンス情報およびレポートをリクエストするユーザー名を OracleAS Portal アクティビティ・ログに登録する場合は、このチェック・ボックスを選択します。
実行時間を表示	(フル・ページのみ) レポートの下部にレポートの実行時間を表示する場合は、このチェック・ボックスを選択します。実行時間は、レポートを生成するリクエストをサーバーが受け取った時点から、レポートの HTML が生成されるまでの時間です。
問合せ条件の表示	(フル・ページのみ) レポートを作成した問合せに渡されるユーザー指定のパラメータ、およびレポートが作成された時間を表示する場合は、このチェック・ボックスを選択します。

図 A-36 列ブレイクのレポート表示オプション

ブレイク・オプション

1から3列の値で、レポートをブレイクするかどうかを選択します。

ブレイク・スタイル

列の第1ブレイク

列の第2ブレイク

列の第3ブレイク

表 A-14 列ブレイクのレポート表示オプション

オプション	説明
ブレイク・スタイル	ブレイク・スタイルを選択します。その後、1、2または3つの列の値で、レポートをブレイクするかどうかを選択します。
列の第1ブレイク	ブレイクする第1の列を選択します。たとえば、DEPTNOを選択すると、レポートには、最初の部門番号に関連付けられているすべての行が表示され、その後、次の部門番号に関連付けられているすべての行が表示され、同様に後続する部門番号について表示されます。 ブレイク列を指定した場合は、同じ順序で列に「 順序基準 」値を指定する必要があります (表 A-15 を参照)。

表 A-14 列ブレイクのレポート表示オプション (続き)

オプション	説明
列の第 2 ブレイク	ブレイクする第 2 の列を選択します。第 2 レベルのブレイクが不要な場合は、% を選択します。
列の第 3 ブレイク	ブレイクする第 3 の列を選択します。第 3 レベルのブレイクが不要な場合は、% を選択します。

図 A-37 行順序のレポート表示オプション

行順序オプション

レポート内の行のソートに使用する値を含む列を選択します。

順序基準	<input type="text" value="%"/>	<input type="text" value="昇順"/>
次の基準	<input type="text" value="%"/>	<input type="text" value="昇順"/>
次の基準	<input type="text" value="%"/>	<input type="text" value="昇順"/>
次の基準	<input type="text" value="%"/>	<input type="text" value="昇順"/>
次の基準	<input type="text" value="%"/>	<input type="text" value="昇順"/>
次の基準	<input type="text" value="%"/>	<input type="text" value="昇順"/>

表 A-15 行順序のレポート表示オプション

オプション	説明
順序基準	レポートの行のソートに使用する値を含む表またはビューの列を選択します。このオプションを選択することは、SQL の ORDER BY 句を指定することと同じです。A から Z または 0 から 9 にソートするには「昇順」を選択します。Z から A または 9 から 0 にソートするには「降順」を選択します。
次の基準	レポートの行のソートに使用する値を含む追加の列を選択します。たとえば、「順序基準」に部門 ID、「次の基準」に従業員名を選択すると、部門 ID を使用してレポートの行が番号順にソートされません。次に、同じ部門 ID の行は、従業員名を使用してアルファベット順にソートされます。

図 A-38 モバイル・デバイスのレポート表示オプション

モバイル表示オプション

モバイル機器でレポートが表示された場合に値を表示する列を選択します。

表示する列	<div style="border: 1px solid black; padding: 2px;"> EMP.DEPTNO EMP.MGR EMP.ENAME EMP.JOB </div>	最大行数/ページ	<input type="text" value="5"/>
-------	---	----------	--------------------------------

表 A-17 「カスタマイズ・フォームの表示オプション」 ページの表示オプション

オプション	説明
必須の値	ユーザーがレポートのカスタマイズ・フォームの列の入力フィールドに値を指定する必要がある場合は、このチェック・ボックスを選択します。このチェック・ボックスの選択を解除すると、ユーザーには値の指定が強制されません。
列名	表またはビューの列を選択します。入力フィールドがレポートのカスタマイズ・フォームに追加されます。この結果、ユーザーはレポートに表示されるデータを制限するための値を指定できます。入力フィールドは、ここで選択した列に対してのみ表示されます。
プロンプト	入力フィールドの横に表示するプロンプト・テキストを入力します。プロンプト・テキストを使用して、フィールドに入力する内容をユーザーに伝えます。次に例を示します。 部門番号を入力してください
LOV	列の入力フィールドに対する値リスト (LOV) を選択します。カスタマイズ・フォームのユーザーは、このリストから値を選択することで、レポートに表示されるデータを制限できます。LOV を検索するには、このフィールドの横にある「ブラウズ」アイコンをクリックします。LOV は、事前に作成し、Portal リポジトリに格納されている必要があります。詳細は、A-88 ページの「 宣言による値リストの構築 」を参照してください。 「LOV」フィールドでは、アプリケーション名の定数 (たとえば、enter #APP_NAME#.dept_lov) を使用できます。詳細は、A-111 ページの「 OracleAS Portal スキーマの参照 」を参照してください。
LOV の表示	値リスト (LOV) を表示する書式を選択します。オプションは次のとおりです。 <ul style="list-style-type: none"> ■ < 空白 >: オプションは選択されていません。 ■ チェック・ボックス: 各選択肢をチェック・ボックスとともに表示します。ユーザーは、希望する値を示すボックスを選択、または選択を解除します。 ■ コンボ・ボックス: ポップアップ・リストと手動による入力フィールドの組合せで表示します。ユーザーは、提供された値を選択するか、または独自に入力できます。 ■ ポップアップ: ユーザーが値を選択するポップアップ・リストを提供します。 ■ ラジオ・グループ: 各選択肢をラジオ・ボタンとともにリストします。ユーザーは、希望する値を示すボタンを選択、または選択を解除します。 ■ 複数選択: 複数の値の選択を許可します。

表 A-17 「カスタマイズ・フォームの表示オプション」 ページの表示オプション (続き)

オプション	説明
パブリックにする	選択すると、値をパブリックにできます。つまり、認証されていないユーザーにデータを表示できます。 ページ・パラメータに関連付けられているポートレット・パラメータのリストで変数になる要素は必ずパブリックにします。
パラメータ追加	クリックすると、レポートのカスタマイズ・フォームに表またはビューの列をさらに挿入するためのフィールドが、このページに追加表示されます。

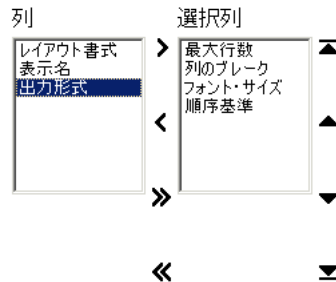
25. 「書式オプション」 セクションで、レポートのカスタマイズ・フォームに挿入する書式オプションを選択します (図 A-40)。

これらの1つ以上のオプションを挿入することで、ユーザーはレポートの最終的な外観を高度に制御できます。

図 A-40 「カスタマイズ・フォームの表示オプション」 ページの「書式オプション」 セクション

書式オプション

レポートのカスタマイズ・フォームに表示する書式オプションを選択します。エンド・ユーザーは、これらのオプションを選択して、レポートの書式を設定できます。

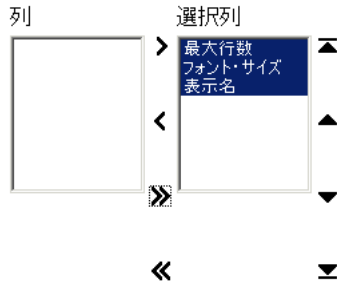


26. 「パブリック書式オプション」 セクションで、レポートが表示されるページのパラメータ・タブに挿入する書式オプションを選択します (図 A-41)。

図 A-41 「カスタマイズ・フォームの表示オプション」ページの「パブリック書式オプション」セクション

パブリック書式オプション

ページのパラメータ・タブに表示する書式オプションを選択します。エンド・ユーザーは、これらのオプションを選択して、レポートの書式を設定できます。



27. 「ボタン・オプション」セクションで、レポートのカスタマイズ・フォームに表示するボタンを選択します (図 A-42)。

表 A-18 に、レポートのカスタマイズ・フォームのボタン表示オプションをリストし、これらのオプションについて説明します。

図 A-42 「カスタマイズ・フォームの表示オプション」ページの「ボタン・オプション」セクション

ボタン・オプション

レポートのカスタマイズ・フォームに表示するボタンを選択します。エンド・ユーザーは、これらのボタンをクリックして、レポートに対する様々な操作を実行できます。

ボタン	名前	位置	文字位置
<input checked="" type="checkbox"/> 実行	レポートの実行	一番上	中央揃え
<input checked="" type="checkbox"/> 保存	保存	一番上	中央揃え
<input type="checkbox"/> バッチ	バッチ	一番上	中央揃え
<input checked="" type="checkbox"/> リセット	リセット	一番上	中央揃え

表 A-18 「カスタマイズ・フォームの表示オプション」 ページのボタン表示オプション

オプション	説明
ボタン	<p>レポートのカスタマイズ・フォームに表示する 1 つ以上のボタンを選択します。</p> <ul style="list-style-type: none"> ■ 実行: ユーザーがカスタマイズ・フォームに指定したオプションを使用して URL ポートレットを表示します。 ■ 保存: レポートのカスタマイズ・フォームにパラメータを再度入力せずに、同じパラメータを使用してレポートを再実行できるように、ユーザーの現在の選択内容を保存します。このボタンを指定すると、レポートのカスタマイズ・フォームには「デフォルトに戻す」ボタンも表示されます。このボタンを使用すると、レポートのカスタマイズ・フォームのすべての値がデフォルト値にリセットされます。 ■ バッチ: URL ポートレットをバッチ・モードで実行し、結果をデータベースに保存します。 ■ リセット: 最後の「保存」、「リセット」または「デフォルトに戻す」イベント以降、レポートのカスタマイズ・フォームに行われた変更を取り消します。
名前	<p>選択したボタンに表示するラベルを入力します。ボタンが大きく表示されないように、短い名前を指定してください。「デフォルトに戻す」ボタンのラベルは変更できません。</p>
位置	<p>レポートのカスタマイズ・フォームに配置するボタンの位置を指定します。「一番上」、「一番上と一番下」または「一番下」から選択します。</p>
文字位置	<p>レポートのカスタマイズ・フォームに配置するボタンの文字位置を指定します。「左揃え」、「中央揃え」または「右揃え」から選択します。</p>

28. 「次へ」をクリックします。

29. 必要に応じて、フル Web ページに表示する際にレポートおよびレポートのカスタマイズ・フォームの外観を設定するためのテンプレートを選択します。

レポートおよびレポートのカスタマイズ・フォームがフル Web ページに表示される場合、ここで選択したテンプレートはこれらの両方に適用されます。レポートおよびレポートのカスタマイズ・フォームがポートレットとして表示される場合、テンプレートは使用されません。その際はページのスタイルが使用されます。

テンプレートは、レポートを表示するページの外観を制御します。対照的に、このウィザードの他のページで指定する表示オプションは、レポート自体の外観を制御します。

テンプレートの要素の表示を確認するには、「**プレビュー**」ボタンをクリックします (図 A-43)。

表 A-19 レポートまたはレポートのカスタマイズ・フォームに挿入するテキストのオプション

オプション	説明
表示名	レポートおよびレポートのカスタマイズ・フォームの表示名を編集します。 このフィールドには HTML を指定できます。
説明	(レポートのみ) このレポートの目的について説明を記載します。これは、レポートまたはレポート・ポートレットが配置される領域に表示対象として選択した属性に従って、表示または非表示となる属性です。
ヘッダー・テキスト	レポートまたはレポートのカスタマイズ・フォームの上部に表示する紹介テキストを入力します。ヘッダー・テキストは、表示名またはタイトルの下に表示されます。 このフィールドには HTML を指定できます。
フッター・テキスト	レポートまたはレポートのカスタマイズ・フォームの下部に表示するテキストを入力します。このフィールドには HTML を指定できます。
ヘルプ・テキスト	レポートまたはレポートのカスタマイズ・フォームのヘルプ・ページに挿入するテキストを入力します。レポートおよびレポートのカスタマイズ・フォームがフル・ページに表示される場合は、「ヘルプ」ボタンが自動的に挿入されます。ポートレットとして表示される場合、「ヘルプ」ボタンを表示するかどうかは、ホスト・ポートレット領域の構成方法によって制御されます。ユーザーが「ヘルプ」ボタンをクリックすると、ここで入力したテキストが表示されます。 このフィールドには HTML を指定できます。
情報テキスト	「情報」ボックスに挿入するテキストを入力します。レポートおよびレポートのカスタマイズ・フォームがフル・ページに表示される場合は、「情報」ボタンが自動的に挿入されます。ポートレットとして表示される場合、「情報」ボタンを表示するかどうかは、ホスト・ポートレット領域の構成方法によって制御されます。ユーザーが「情報」ボタンをクリックすると、ここで入力したテキストが表示されます。

31. 「次へ」をクリックします。

32. 必要に応じて、レポートまたはレポートのカスタマイズ・フォーム HTML の実行中に、特定の時点で実行する PL/SQL コードを入力します (図 A-44)。

表 A-20 に、レポートまたはレポートのカスタマイズ・フォームを作成する HTML の実行中に、様々な時点で実行する PL/SQL コードに使用できるオプションをリストし、これらのオプションについて説明します。

注意： これらのフィールドに入力する PL/SQL では、OracleAS Portal スキーマ、アプリケーション・スキーマおよびアプリケーション名に定数を使用できます。詳細は、A-111 ページの「OracleAS Portal スキーマの参照」を参照してください。

図 A-44 「追加 PL/SQL コード」 ページ

SAMPLE_REPORT

ステップ 16 / 16

追加PL/SQLコード

レポートおよびカスタマイズ・フォーム実行時に、様々な箇所で実行されるPL/SQLコードを入力します。

	レポート	カスタマイズ・フォーム
...ページ表示前	<input type="text"/>	<input type="text"/>
...ヘッダー表示後	<input type="text"/>	<input type="text"/>
...フッター表示前	<input type="text"/>	<input type="text"/>
...ページ表示後	<input type="text"/>	<input type="text"/>

< 前へ 完了 取消

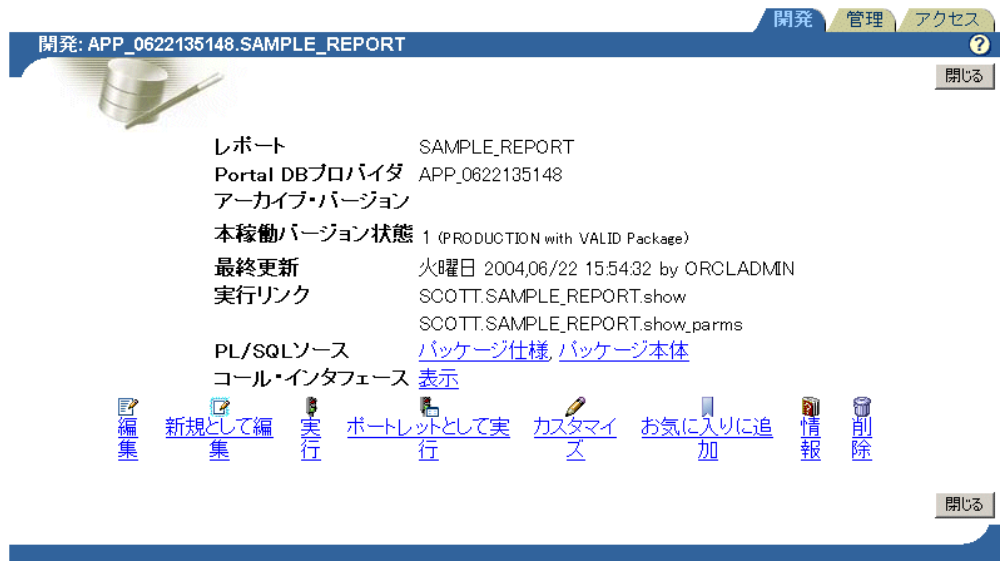
表 A-20 レポート公開時に実行する PL/SQL コードのオプション

オプション	説明
... ページ表示前	レポートまたはレポートのカスタマイズ・フォームを含むページが表示される前に実行する PL/SQL プロシージャを入力します。
... ヘッダー表示後	レポートまたはレポートのカスタマイズ・フォームのヘッダーが表示された後に実行する PL/SQL プロシージャを入力します。
... フッター表示前	レポートまたはレポートのカスタマイズ・フォームのフッターが表示される前に実行する PL/SQL プロシージャを入力します。
... ページ表示後	レポートまたはレポートのカスタマイズ・フォームを含むページが表示された後に実行する PL/SQL プロシージャを入力します。

33. 「完了」をクリックします。

レポートが保存され、レポートを管理およびテスト・ランできるサマリー・ページが表示されます (図 A-45)。以降の項では、これらのタスクの詳細について説明します。

図 A-45 レポートのサマリー・ページ



interMedia リッチ・コンテンツに対するフォームおよびレポートの構築

イメージ、オーディオ、ビデオなどの *interMedia* リッチ・コンテンツは、OracleAS Portal で構築したレポートおよびフォームに統合できます。これらのオブジェクトは、Oracle9i Database Server に格納されます。

表示できる *interMedia* リッチ・コンテンツのデータベース列のタイプは、次のとおりです。

- **ORDIMAGE**: このオブジェクト・タイプは、GIF、JPEG および BMP などのイメージ・データの格納、管理および操作をサポートします。
- **ORDAUDIO**: このオブジェクト・タイプは、MP3、AU、WAV および MPEG などのオーディオ・データの格納、管理および操作をサポートします。
- **ORDVIDEO**: このオブジェクト・タイプは、REAL、QuickTime 3/4、AVI および MPEG などのビデオ・データの格納、管理および操作をサポートします。

OracleAS Portal のレポートまたはフォームでは、*interMedia* リッチ・コンテンツのこれらのタイプの列を使用できることに加え、結合条件、書式、列条件などを使用してオブジェクト属性を表示できます。たとえば、ビデオ・クリップのサイズや期間をレポートに表示したり、特定の日付以降に変更されたオブジェクトのみを表示するようにレポート条件を設定できます。

interMedia は様々なコンテンツ・タイプと書式をサポートしていますが、これらを表示するために使用するブラウザは、対応する MIME タイプをネイティブにサポートしているか、または Web では一般的にサポートされていないリッチ・コンテンツを表示するためのプラグインがインストールされている必要があります。たとえば、ほとんどのブラウザは、GIF および JPEG イメージをネイティブに表示できますが、TIFF イメージはプラグインがインストールされていないと表示されません。

interMedia とレポート

OracleAS Portal の 2 種類のレポート・ウィザードには、*interMedia* リッチ・コンテンツを組み込むステップがあります。

- 問合せウィザードからのレポート
- 例による問合せ (QBE) レポート

QBE レポートでは、オブジェクト・タイプの属性は表示されません。したがって、レポートのカスタマイズ・フォームで *interMedia* ベースの列に値を指定しようとするとうエラーになります。これは、*interMedia* オブジェクト・タイプの列に対して実行する QBE レポートには、レポートのカスタマイズ・フォームを使用できないことを意味します。

2 つの方法の 1 つを使用して、*interMedia* リッチ・コンテンツをレポートに表示できます。

- 埋込み (インライン)

レポート内のその場でイメージ、オーディオまたはビデオを実行します。

- リンク・アイコン

ユーザーがリンクをクリックすると、コンテンツが新しいブラウザ・ウィンドウに表示されるか、または対応するソース・アプリケーションでコンテンツが処理されます。たとえば、RealPlayer のウィンドウにコンテンツが表示されます。

これらの表示オプションは、OracleAS Portal のレポートの *interMedia* コンテンツが含まれているすべての列に適用されます。これらのオプションは、OracleAS Portal レポート・ウィザードの「表示オプション」タブで選択できます。デフォルトでは、*interMedia* リッチ・コンテンツは、アイコンとしてレポートに表示されます。ユーザーがアイコンをクリックすると、実際のコンテンツが表示されます。デフォルトの使用を予定している場合は、これ以上必要な設定はありません。*interMedia* リッチ・コンテンツの埋込みを計画している場合は、レポート・ウィザードの「表示オプション」ページで「レポートに *interMedia* リッチ・コンテンツを埋め込む」チェック・ボックスを選択します。

レポート・ウィザードの詳細は、A-36 ページの「宣言によるレポートの構築」を参照してください。

***interMedia* とフォーム**

OracleAS Portal の 2 種類のフォーム・ウィザードには、*interMedia* リッチ・コンテンツをフォームに組み込むステップがあります。

- 表またはビューをベースとするフォーム
- マスター・ディテール・フォーム

どちらの方法を使用する場合でも、フォームを介して変更する基礎となるデータベース・オブジェクトは、ORDIMAGE、ORDAUDIO または ORDVIDEO タイプの列に基づいている必要があります。

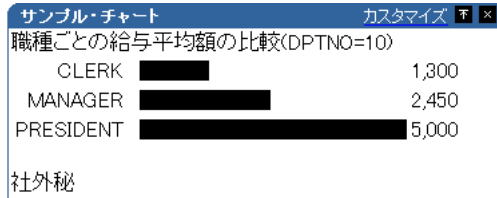
interMedia リッチ・コンテンツをフォームに組み込むには、OracleAS Portal フォーム・ウィザードの「書式および妥当性チェック・オプション」ページで、左側のフレームで列名を選択し、右側のフレームで「アイテム・タイプ」に「ファイルのアップロード (*interMedia*)」を指定します。

注意： 列のタイプの割当ては、OracleAS Portal 外部で実行されます。*interMedia* タイプの列を備えた表を作成する必要がある場合は、データベース管理者に相談してください。

イメージ、オーディオおよびビデオ・クリップを BLOB 列ではなく *interMedia* ベースの列にアップロードする利点の 1 つは、MIME タイプ、長さ、元のメディア・ファイルに挿入されている可能性があるユーザー定義メタデータなど、いくつかの属性を抽出するために、データが自動的に解析されることです。たとえば、QuickTime ファイルには字幕放送機能、RealVideo ファイルには著作権情報が含まれていることがあります。*interMedia* ベースの列では、これらの属性が自動的に抽出され、索引付けや問合せのために、*interMedia* オブジェクトに格納されます。

宣言によるチャートの構築

図 A-46 ポートレット・ビルダーのチャート・ウィザードによるチャートの構築



チャートでは、データベース表またはビューのデータが棒グラフとして表示されます。チャートは、表またはビューの少なくとも 2 つの列がベースとなります。

- 「ラベル」列に値を指定することで、チャート上にバーの表示名が提供されます。
- 「値」列の値を使用して、チャート上で相互に連動するバーのサイズが計算されます。「値」列には、常に数値データを指定する必要があります。

「リンク」列を指定することもできます。この列の値によって、チャートのラベルから他の OracleAS Portal データベース・ポートレットまたは URL へのハイパーテキスト・リンクが作成されます。たとえば、リンクが適切に配置されていると、ユーザーはチャート内でラベルをクリックし、チャートのベースとなるデータを更新できるフォームにジャンプできます。

チャートは、集計データを表示する方法としても効果的です。ウィザードで「**グループ基準**」および「**サマリー・オプション**」を使用すると、たとえば、最小値、最大値、平均値または列内の値の個数など、問合せから戻った列値を集計できます。

OracleAS Portal のポートレット構築ウィザードには、チャートを構築するための 2 種類の構成方式が用意されています。

- 問合せウィザードからのチャート
- SQL 問合せからのチャート

問合せウィザードは、チャートの構成過程を通じて高レベルの支援を提供します。SQL 問合せによる方法では、チャートのデータを取り出すために使用する問合せ全体を高度に制御できます。

この項では、問合せウィザードを使用してチャートを構築する方法について説明します。

問合せウィザードを使用してチャートを作成する手順は、次のとおりです。

1. A-16 ページの「[宣言によるポートレットの構築](#)」に記載されている手順に従います。手順 6 を完了した後、この項に戻ります。
2. 「新規作成 ...」の横にある「[チャート](#)」リンクをクリックします (図 A-47)。

図 A-47 「新規作成 ...」の横にある「チャート」リンク

これらは、使用可能なポートレットおよび操作です。

新規作成... [フォーム](#)、[レポート](#)、[チャート](#)、[カレンダー](#)、[動的ページ](#)、[XMLコンポーネント](#)、[階層メニュー](#)、[URL](#)、[フレームドライバ](#)、[リンク](#)、[値リスト](#)、[データコンポーネント](#)

3. 「チャート」ページで、「問合せウィザードからのチャート」をクリックします (図 A-48)。

図 A-48 「問合せウィザードからのチャート」リンク**チャート**

独自のSQL問合せを構築するウィザード、または独自のSQL問合せの記述を可能にするウィザードを使用して、チャートを作成します。

[問合せウィザードからのチャート](#)

ウィザードを使用して、SQL問合せを構築します。

[SQL問合せからのチャート](#)

独自のSQL問合せを記述します。

4. 表示されたページで、「名前」フィールドにチャートの内部名を入力します (図 A-49)。内部名はユーザーには公開されません。

図 A-49 チャート・ウィザードのステップ 1

5. 「表示名」フィールドに、ユーザーがこのチャートを識別するための名前を入力します (図 A-49)。

表示名は実行時に使用されます。表示名は、チャートのバナーと同様にブラウザ・ウィンドウのタイトルとして表示されます。チャートをポートレットとして公開すると、表示名は、ポートレットのタイトルおよびポートレット・リポジトリの「使用可能なポートレット」リスト内で使用されます。

6. 「説明」フィールドに、このチャートの説明を入力します (図 A-49)。

説明は、ポートレット・リポジトリのポートレットの下に表示されます。説明には、ポートレットの目的の概要、ポートレットのタイプの分類、またはその他の説明的な情報を記載します。開発組織内で、説明に記載する内容について標準的なアプローチを設定すると便利です。

7. 「Portal DB プロバイダ」のリストから、このチャートを所有するプロバイダを選択します (図 A-49)。

このリストには、ポートレットを構築する権限が付与されているプロバイダのみが表示されます。宣言的に構築したポートレットを所有またはホスティングするために作成した専用のプロバイダを使用することをお勧めします。

8. 「次へ」をクリックします。

9. 「表またはビュー」の横にある「リスト」アイコンをクリックし、チャートのベースとする表またはビューを選択します (図 A-50)。

注意: 「表 / ビュー」フィールドでは、OracleAS Portal スキーマおよびアプリケーション・スキーマに定数を使用できません (たとえば、#APP_SCHEMA#EMP を入力できません)。詳細は、A-111 ページの「OracleAS Portal スキーマの参照」を参照してください。

図 A-50 「リスト」アイコンのクリックによる表またはビューの選択



ここで選択できる表およびビューは、前の手順で選択した Portal DB プロバイダからアクセス可能な表およびビューです。また、リストは、権限 (プロバイダ・スキーマ内のすべての表に対する表示権限、パブリックに対する SELECT 権限、プロバイダ・スキーマに対する SELECT 権限) が付与されている表およびビューに制限されます。

表 A-21 チャートのデータ選択オプション

オプション	説明
ラベル	チャート上の各バーのラベルとして使用する値が含まれている列を選択します。たとえば、部内の各職種の給与を比較するチャートを作成する場合は、「ラベル」に JOB 列、「値」に SAL、「集計関数」に AVG を選択します。
リンク	ここで使用できるリンクは、事前に構成し、データベースに格納されているリンクです。このフィールドを使用して、チャートのバー・ラベルから OracleAS Portal の別のポートレットまたは URL にジャンプする構成済のリンクを選択できます。リンク・パラメータを設定するには、「リンク」フィールドの横にある「編集」アイコンをクリックします。リンク・パラメータを設定しない場合は、リンクのデフォルト値が使用されます。
値	<p>チャートのバーの相対的なサイズを計算するために使用する値が含まれている列を選択します。「値」列には、常に数値データを指定する必要があります。</p> <p>「値」のデフォルトは、1 です。1 の値を指定することは、集計関数を選択する場合にも便利です。たとえば、「ラベル」に SCOTT.EMP の JOB 列、「値」に 1、「集計関数」に COUNT を選択できます。この選択によって、各職種の従業員数を表示するチャートが作成されます。</p>
集計関数	<p>集計関数は、「値」列の数値のグループから単一の集計値（%、合計、件数、最大値、最小値、標準偏差または分散）を計算します。OracleAS Portal では、「ラベル」列の値を使用して、これらのグループ化を判断します。</p> <p>たとえば、「ラベル」に SCOTT.EMP の JOB 列、「値」に SAL、「集計関数」に AVG を選択できます。この選択によって、各職種の平均給与を表示するチャートが作成されます。</p>

12. 「次へ」をクリックします。

13. 必要に応じて、チャートに表示されるデータを制限する条件を指定します。

たとえば、部門 10 の全従業員のデータを表示するには、「列名」から DEPTNO を選択し、「条件」から = または LIKE を選択し、「値」フィールドに 10 と入力します (図 A-52)。

表 A-22 「列条件」 ページのオプション

オプション	説明
列名	チャートに表示されるデータを制限するために使用する値の列を選択します。たとえば、SCOTT.EMP 表の EMPNO 列から 3000 を超える値を表示するには、「列名」として EMPNO を選択し、「条件」から > を選択し、「値」フィールドに 3000 と入力します。
条件	<p>「列名」と「値」との関係を定義する条件を選択します。オプションは次のとおりです。</p> <ul style="list-style-type: none"> ■ 等しい (=) ■ より大きい (>) ■ 以上 (>=) ■ より小さい (<) ■ 以下 (<=) ■ LIKE ■ 等しくない (!=) ■ NULL ■ NOT NULL ■ IN ■ NOT IN <p>IN または NOT IN 条件の後に複数の値を指定するには、10:20:30 のように各値の間にコロン (:) を入力します。</p>
値	チャートに表示されるデータを制限するための値を入力します。
条件追加	クリックすると、追加条件を指定するためのフィールドがさらに表示されます。

14. 「次へ」をクリックします。

15. 必要な場合は、条件を指定し、これらの条件を満たす場合に、チャート・データに対して適用する特別な書式を定義します。

たとえば、特定の値を超える列値に関連付けられているバーをチャート上に赤で表示するように指定できます (図 A-53)。

表 A-23 チャート・ウィザードの「書式条件」ページのオプション

オプション	説明
列	条件を適用する列を選択します。
条件	<p>「列」と「値」との関係を定義する条件を選択します。オプションは次のとおりです。</p> <ul style="list-style-type: none"> ■ 等しい (=) ■ より大きい (>) ■ 以上 (>=) ■ より小さい (<) ■ 以下 (<=) ■ LIKE ■ 等しくない (!=) ■ NULL ■ NOT NULL ■ IN ■ NOT IN ■ BETWEEN
値	<p>条件をトリガーする列値を入力します。数値に書式マスクを適用する必要はありません。たとえば、表に EMPNO 列があり、3000 より大きい EMPNO 値に対して特別な書式を適用する場合は、「条件」で > を選択し、このフィールドに 3000 と入力します。日付は NLS 日付書式 (nls_date_format) で入力します。</p> <p>IN または NOT IN 条件の後に複数の値を指定するには、10:20:30 のように各値の間にコロン (:) を入力します。BETWEEN 条件を指定するには、2 つの値をカンマ、空白または AND という語で区切ります。</p>
列 / バー	<p>表示された列名または条件を満たす値に関連付けられている <BAR> に、条件付書式を適用するかどうかを選択します。</p> <p>たとえば、SAL 列と ENAME 列を使用するチャートを考えてみま。ENAME はチャート内の各バーにラベルを付けるために使用し、SAL はバーのサイズを導出する値を指定するために使用します。給与が 40,000 未満の場合にバーを赤く点滅表示するには、<BAR> を選択します。条件を満たす場合に ENAME を赤く点滅表示するには、ENAME を選択します。ラベルとバーに加え、条件を満たす場合にチャート内に含まれている給与を赤く点滅表示するには、SAL を選択します。</p>

表 A-23 チャート・ウィザードの「書式条件」ページのオプション（続き）

オプション	説明
色	指定の条件を満たすデータを表示する色を選択します。色は、「列 / バー」フィールドの選択に従って、テキストまたはバーに適用されます。
フォント	指定の条件を満たすデータを表示するフォントを選択します。
A	指定の条件を満たす結果を太字で表示する場合は、このチェック・ボックスを選択します。
<i>A</i>	指定の条件を満たす結果をイタリックで表示する場合は、このチェック・ボックスを選択します。
A	指定の条件を満たす結果にアンダスコアを表示する場合は、このチェック・ボックスを選択します。
点減	指定の条件を満たす結果を点減させる場合は、このチェック・ボックスを選択します。
順序	連番を入力し、書式設定を実行する優先順位を設定します。たとえば、1つの行に複数の条件を満たすデータが含まれており、それぞれの書式ルールが異なる場合に役立ちます。順序が最も高い（つまり、1）条件が、他のすべてよりも優先されます。

16. 「次へ」をクリックします。

17. チャートに使用できる様々なビューに対する表示オプションを定義します。

- 表 A-24 に、チャートのすべてのビューに共通する表示オプションを示します (図 A-54)。
- 表 A-25 に、チャートのフル・ページ・ビューに共通する表示オプションを示します (図 A-55)。
- 表 A-26 に、ポートレットとして表示するチャートの表示オプションを示します (図 A-56)。
- 表 A-27 に、モバイル・デバイスに表示するチャートの表示オプションを示します (図 A-57)。

図 A-54 すべてのチャート・ビューに共通する表示オプション

SAMPLE_CHART

ステップ 7 / 10

表示オプション

チャートの外観を制御するオプションを選択します。

共通オプション

順序基準 ORDER BY VALUE

NULL値を含む NULL値の取扱い 0

有効期限までの時間(分) 0

表 A-24 すべてのチャート・ビューに共通する表示オプション

オプション	説明
順序基準	<p>チャートのデータを並べる方法を選択します。</p> <ul style="list-style-type: none"> ■ ORDER BY VALUE: チャートのバーは、「値」入力フィールドに指定した表の列値と同順（数値ソート）で表示されます。 ■ ORDER BY VALUE DESC: チャートのバーは、「値」入力フィールドに指定した表の列値の逆順（数値ソート）で表示されます。 ■ ORDER BY LABEL: チャートのバーは、「ラベル」入力フィールドに指定した表の列値と同順（文字列ソート）で表示されます。 ■ ORDER BY LABEL DESC: チャートのバーは、「ラベル」入力フィールドに指定した表の列値の逆順（文字列ソート）で表示されます。
NULL 値を含む	チャートに NULL 値を表示する場合は、このチェック・ボックスを選択します。
NULL 値の取扱い	たとえば、（このフィールドは Null）など、チャート内の NULL 値に対して表示するテキスト文字列を入力します。
有効期限までの時間（分）	最初に表示した後、キャッシュにチャートを保持する時間（分）を入力します。期限が切れると、チャート・データはデータベースからリフレッシュされます。

図 A-55 フル Web ページ・チャート・ビューの表示オプション

フル・ページ・オプション

フォント	Tahoma	フォント色	
フォント・サイズ	12pt	チャート・タイプ	水平
軸	0(標準)	棒グラフ	MULTI
チャート・スケール	200%	棒の幅	15
棒の高さ	30	値の書式マスク	999,999,999,999,99!
最大行数/ページ	20	サマリー・オプション	<input type="checkbox"/> 平均値 <input type="checkbox"/> 軸名 <input type="checkbox"/> 値の件数 <input type="checkbox"/> 最初の値

ログを記録
 実行時間を表示
 問合せ条件の表示

表 A-25 フル Web ページ・チャート・ビューに適用する表示オプション

オプション	説明
フォント	チャート・テキストを表示するフォントを選択します。
フォント色	チャート・テキストを表示するフォント色を選択します。ユーザー定義の色は使用できません。
フォント・サイズ	チャート・テキストを表示するフォント・サイズを選択します。
チャート・タイプ	チャートのバーを表示する方向（水平または垂直）を選択します。
軸	チャートの軸の値に対してチャートのバーを表示する方法を選択します。たとえば、「0（標準）」を選択すると、軸の値は0（ゼロ）に設定されます。「平均値」を選択すると、軸は、チャートの「値」セクションのベースとなっている表またはビューの列のすべての平均値に設定されます。
棒グラフ	チャート上にバーを表示するために使用するイメージを選択します。バーを異なる色で表示するには、「MULTI」を選択します。
チャート・スケール	チャートをホスティングする Web ページに対してチャートのバーのスケールを設定するためのパーセント（%）を選択します。パーセントを高くすると、バーは大きく表示されます。
棒の幅	チャートのバーの幅をピクセルで選択します。このオプションは、水平方向と垂直方向の両方のチャートのバーに適用されます。
棒の高さ	チャートのバーの高さをピクセルで入力します。このオプションは、水平方向と垂直方向の両方のチャートのバーに適用されます。
値の書式マスク	チャートに表示する数値または日付の書式を入力します。
最大行数 / ページ	1 ページに表示するバーの最大数を入力します。

表 A-25 フル Web ページ・チャート・ビューに適用する表示オプション (続き)

オプション	説明
サマリー・オプション	チャートに関するサマリー情報を表示するための 1 つ以上のオプションを選択します。選択した各オプションは、チャート下部のサマリー情報ボックスに入ります。Windows ユーザーは、[Ctrl] キーを押しながら各オプションをクリックすることで、複数のオプションを選択できます。
ログを記録	OracleAS Portal アクティビティ・ログに情報を入力する場合は、このチェック・ボックスを選択します。情報には、パフォーマンス統計およびチャートをリクエストするユーザー名が含まれます。
実行時間を表示	チャートに対して HTML を生成するリクエストをサーバーが受け取った時間を表示する場合は、このチェック・ボックスを選択します。この情報は、チャートの下部に表示されます。
問合せ条件の表示	チャートを作成した SQL 問合せに渡したユーザー指定の全パラメータとチャートが作成された時間を表示する場合は、このチェック・ボックスを選択します。この情報は、チャートの下部に表示されます。

図 A-56 ポートレットとして表示されるチャートの表示オプション

ポートレット・オプション

ページ・スタイル フォント

フォント色 フォント・サイズ

注意: フォント名、色、およびサイズはページ・スタイルの設定よりも優先されます。

棒の幅 棒の高さ

最大行数/ページ

表 A-26 ポートレットとして表示されるチャートの表示オプション

オプション	説明
ページ・スタイル	チャート・テキストを表示するために使用するページ・スタイル要素を選択します。
フォント	選択したページ・スタイル要素のフォントを上書きするフォントを選択します。
フォント色	選択したページ・スタイル要素のフォント色を上書きするフォント色を選択します。
フォント・サイズ	選択したページ・スタイル要素のフォント・サイズを上書きするフォント・サイズを選択します。
棒の幅	チャートのバーの幅をピクセルで選択します。このオプションは、水平方向と垂直方向の両方のチャートのバーに適用されます。

表 A-26 ポートレットとして表示されるチャートの表示オプション（続き）

オプション	説明
棒の高さ	チャートのバーの高さをピクセルで選択します。このオプションは、水平方向と垂直方向の両方のチャートのバーに適用されます。
最大行数 / ページ	チャートに表示するバーの最大数を入力します。

図 A-57 モバイル・デバイスに表示するチャートの表示オプション

モバイル表示オプション

最大行数/ページ

表 A-27 モバイル・デバイスに表示するチャートの表示オプション

オプション	説明
最大行数 / ページ	チャートに表示するバーの最大数を入力します。

18. 「次へ」をクリックします。

19. チャートのカスタマイズ・フォームに表示するコンテンツを定義します。

このページのオプションを使用すると、ユーザーが、チャートのビューに表示されるデータ全体を制御できるようになります。

たとえば、ここで「列名」に SCOTT.EMP 表の DEPTNO 列を選択すると、DEPTNO の入力フィールドがチャートのカスタマイズ・フォームに追加されます。ユーザーは、フィールドに部門番号を入力することで、その部門に関連するデータのみに表示を制限できます。

必要に応じて、入力ファイルに値リストを追加できます。DEPTNO に数値を入力することをユーザーに要求するのではなく、たとえば、10、20、30 などを選択できる値リストを追加できます。

ポートレットの実装では、ユーザーがポートレットの上部の「**カスタマイズ**」リンクをクリックすると、チャートのカスタマイズ・フォームが表示されます。フル・ページの実装では、ユーザーにチャートを実行するリンクを提供できます。リンクはチャートのカスタマイズ・フォームをターゲットにでき、「**実行**」ボタンも提供できます。カスタマイズ・オプションを設定した後、ユーザーはこのボタンをクリックしてチャートを実行します。

必要な場合は、カスタマイズ・フォームにユーザー独自の値を指定できるようにバインド変数を使用できます。詳細は、A-114 ページの「**バインド変数の使用**」を参照してください。

- 表 A-28 に、チャートのカスタマイズ・フォームの表示オプションを示します (図 A-58)。
- 表 A-29 に、チャートのカスタマイズ・フォームの書式オプションを示します (図 A-59)。
- 表 A-30 に、チャートのカスタマイズ・フォームのパブリック書式オプションを示します (図 A-60)。
- 表 A-31 に、チャートのカスタマイズ・フォームのボタン・オプションを示します (図 A-61)。

図 A-58 チャートのカスタマイズ・フォームの表示オプション

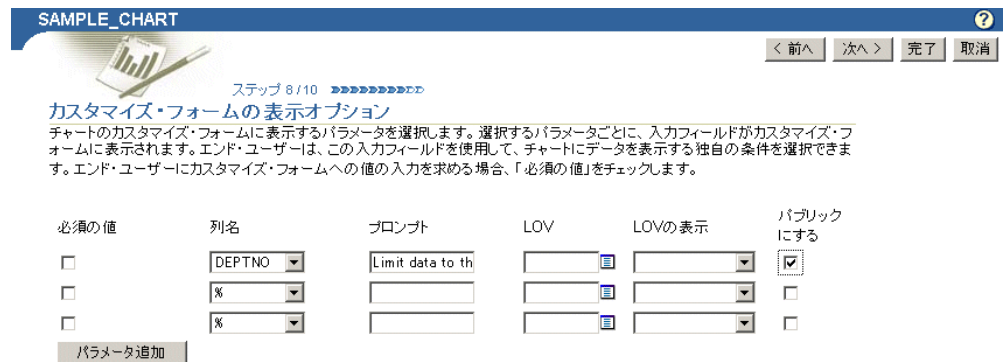


表 A-28 チャートのカスタマイズ・フォームの表示オプション

オプション	説明
必須の値	ユーザーがチャートのカスタマイズ・フォームの列の入力フィールドに値を指定する必要がある場合は、このチェック・ボックスを選択します。
列名	チャートのカスタマイズ・フォームに挿入する表またはビューの列を選択します。入力フィールドは、チャートに表示されるデータを制限するための値をユーザーが指定できるフォームに表示されます。表またはビューの列を選択しないと、カスタマイズ・フォームに入力フィールドは表示されません。
プロンプト	入力フィールドの横に表示するプロンプト・テキストを入力します。プロンプト・テキストを使用して、フィールドに入力する内容をユーザーに伝えます。次に例を示します。表示する部門番号を選択してください

表 A-28 チャートのカスタマイズ・フォームの表示オプション（続き）

オプション	説明
LOV	<p>列の入力フィールドに対する値リスト（LOV）を選択します。カスタマイズ・フォームのユーザーは、このリストから値を選択することで、レポートに表示されるデータを制限できます。LOVを検索するには、このフィールドの横にある「ブラウズ」アイコンをクリックします。LOVは、事前に作成し、Portal リポジトリに格納されている必要があります。詳細は、A-88 ページの「宣言による値リストの構築」を参照してください。</p> <p>「LOV」フィールドには、アプリケーション名の定数（たとえば、enter #APP_NAME#.dept_lov）を使用できます。詳細は、A-111 ページの「OracleAS Portal スキーマの参照」を参照してください。</p>
LOV の表示	<p>値リスト（LOV）を表示する書式を選択します。オプションは次のとおりです。</p> <ul style="list-style-type: none"> ■ <空白>: オプションは選択されていません。 ■ チェック・ボックス: 各選択肢をチェック・ボックスとともに表示します。ユーザーは、希望する値を示すボックスを選択、または選択を解除します。 ■ コンボ・ボックス: ポップアップ・リストと手動による入力フィールドの組合せで表示します。ユーザーは、提供された値を選択するか、または独自に入力できます。 ■ ポップアップ: ユーザーが値を選択するポップアップ・リストを提供します。 ■ ラジオ・グループ: 各選択肢をラジオ・ボタンとともにリストします。ユーザーは、希望する値を示すボタンを選択、または選択を解除します。 ■ 複数選択: 複数の値の選択を許可します。
パブリックにする	<p>この要素をチャートのカスタマイズ・フォームに表示する場合は、このフィールドを選択します。</p> <p>ページ・パラメータに関連付けられているポートレット・パラメータのリストで変数になる要素は必ずパブリックにします。</p>
パラメータ追加	<p>クリックすると、カスタマイズ・フォームに入力フィールドがさらに追加されます。</p>

図 A-59 チャートのカスタマイズ・フォームの書式オプション

書式オプション

チャートのカスタマイズ・フォームに表示する書式オプションを選択します。エンド・ユーザーは、これらのオプションを選択して、チャートの書式を設定できます。

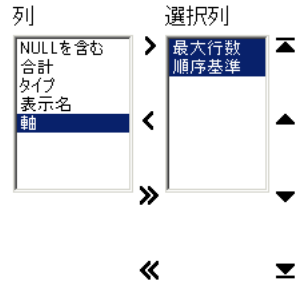


表 A-29 チャートのカスタマイズ・フォームの書式オプション

オプション	説明
軸	選択すると、チャートのカスタマイズ・フォームのユーザーが、チャートの軸の値に対してチャートのバーを表示する方法を選択できるようになります。たとえば、「0 (標準)」を選択すると、軸の値は0 (ゼロ) に設定されます。「平均値」を選択すると、軸は、チャートの「値」セクションのベースとなっている表またはビューの列のすべての平均値に設定されます。
NULL を含む	選択すると、チャートのカスタマイズ・フォームのユーザーが、チャートに NULL 値を表示するかどうかを指定できるようになります。
最大行数	選択すると、チャートのカスタマイズ・フォームのユーザーが、チャートに表示するバーの最大数を指定できるようになります。
合計	選択すると、チャートのカスタマイズ・フォームのユーザーが、チャートに関するサマリー情報を表示する 1 つ以上のオプションを選択できるようになります。選択した各オプションは、チャート下部のサマリー情報ボックスに入ります。
タイプ	選択すると、チャートのカスタマイズ・フォームのユーザーが、チャート・テキストを表示するフォントを選択できるようになります。
表示名	選択すると、チャートのカスタマイズ・フォームのユーザーが、チャートの表示名を編集できるようになります。
順序基準	選択すると、チャートのカスタマイズ・フォームのユーザーが、チャート・データの表示順序を設定できるようになります。

図 A-60 チャートのカスタマイズ・フォームのパブリック書式オプション

パブリック書式オプション

ページのパラメータ・タブに表示する書式オプションを選択します。エンド・ユーザーは、これらのオプションを選択して、チャートの書式を設定できます。

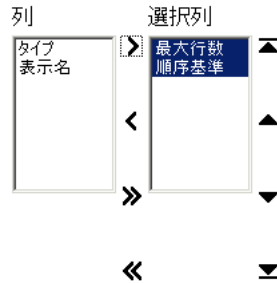


表 A-30 チャートのカスタマイズ・フォームのパブリック書式オプション

オプション	説明
タイプ	選択すると、チャートのカスタマイズ・フォームの未認証ユーザーが、チャート・テキストを表示するフォントを選択できるようになります。
表示名	選択すると、チャートのカスタマイズ・フォームの未認証ユーザーが、チャートの表示名を編集できるようになります。
最大行数	選択すると、チャートのカスタマイズ・フォームの未認証ユーザーが、チャートに表示するバーの最大数を指定できるようになります。
順序基準	選択すると、チャートのカスタマイズ・フォームの未認証ユーザーが、チャート・データの表示順序を設定できるようになります。

図 A-61 チャートのカスタマイズ・フォームのボタン・オプション

ボタン・オプション

チャートのカスタマイズ・フォームに表示するボタンを選択します。エンド・ユーザーは、これらのボタンをクリックして、チャートに対する様々な操作を実行できます。

ボタン	名前	位置	文字位置
<input checked="" type="checkbox"/> 実行	実行	一番上	中央揃え
<input checked="" type="checkbox"/> 保存	保存	一番上	中央揃え
<input type="checkbox"/> バッチ	バッチ	一番上	中央揃え
<input checked="" type="checkbox"/> リセット	リセット	一番上	中央揃え

表 A-31 チャートのカスタマイズ・フォームのボタン・オプション

オプション	説明
ボタン	<p>(フル・ページのみ) チャートのカスタマイズ・フォームに表示する1つ以上のボタンを選択します。</p> <ul style="list-style-type: none"> ■ 実行: ユーザーがチャートのカスタマイズ・フォームに指定したオプションを使用して URL ポートレットを表示します。 ■ 保存: チャートのカスタマイズ・フォームにパラメータを再度入力せずに、同じパラメータを使用してチャートを再実行できるように、ユーザーの現在の選択内容を保存します。このボタンを指定すると、チャートのカスタマイズ・フォームには「デフォルトに戻す」ボタンも表示されます。このボタンを使用すると、チャートのカスタマイズ・フォームのすべての値がデフォルト値にリセットされます。 ■ バッチ: URL ポートレットをバッチ・モードで実行し、結果をデータベースに保存します。 ■ リセット: 最後の「保存」、「リセット」または「デフォルトに戻す」イベント以降、チャートのカスタマイズ・フォームに行われた変更を取り消します。
名前	<p>選択したボタンに表示するラベルを入力します。ボタンが大きく表示されないように、短い名前を指定してください。「デフォルトに戻す」ボタンのラベルは変更できません。</p>
位置	<p>チャートのカスタマイズ・フォームに配置するボタンの位置を指定します。「一番上」、「一番上と一番下」または「一番下」から選択します。</p>
文字位置	<p>チャートのカスタマイズ・フォームに配置するボタンの文字位置を指定します。「左揃え」、「中央揃え」または「右揃え」から選択します。</p>

20. 「次へ」をクリックします。

21. 必要に応じて、チャートをホスティングするページの外観を定義するテンプレートを選択し、チャートおよびチャートのカスタマイズ・フォームの上部または下部（あるいはその両方）に表示するテキストを入力します。

テンプレートは、チャートがフル Web ページに表示される場合のみ適用されます。チャートがポートレットとして表示される場合、ここで選択するテンプレートは無視され、ホスト・ページのスタイルが使用されます。

表 A-32 に、このページで使用可能なオプションをリストし、これらのオプションについて説明します (図 A-62)。

表 A-32 「チャートとカスタマイズ・フォームのテキスト」 ページで使用可能なオプション

オプション	説明
テンプレート	<p>チャートをホスティングするページの外観を設定するためのテンプレートを選択します。テンプレートは、チャートがフル Web ページに表示される場合のみ適用されます。チャートがポートレットとして表示される場合、ここで選択するテンプレートは無視され、ホスト・ページのスタイルが使用されます。</p> <p>ウィザードの前のページで指定した表示オプションは、チャート自体の外観を制御します。</p>
テンプレートのプレビュー	<p>クリックすると、「テンプレート」リストで現在選択されているテンプレートの外観が表示されます。</p>
説明	<p>(チャートのみ) このチャートの目的について説明を記載します。これは、チャートまたはチャート・ポートレットが配置される領域に表示対象として選択した属性に従って、表示または非表示となる属性です。</p>
表示名	<p>チャートの表示名またはチャートのカスタマイズ・フォームのタイトルを編集します。</p> <p>このフィールドには HTML を指定できます。</p>
ヘッダー・テキスト	<p>チャートまたはチャートのカスタマイズ・フォームの上部に表示する紹介テキストを入力します。このテキストは、表示名またはタイトルの下に表示されます。</p> <p>このフィールドには HTML を指定できます。</p>
フッター・テキスト	<p>チャートまたはチャートのカスタマイズ・フォームの下部に表示するテキストを入力します。</p> <p>このフィールドには HTML を指定できます。</p>
ヘルプ・テキスト	<p>チャートまたはチャートのカスタマイズ・フォームのヘルプ・ページに表示するテキストを入力します。チャートおよびチャートのカスタマイズ・フォームがフル・ページに表示される場合は、「ヘルプ」ボタンが自動的に挿入されます。ポートレットとして表示される場合、「ヘルプ」ボタンを表示するかどうかは、ホスト・ポートレット領域の構成方法によって制御されます。ユーザーが「ヘルプ」ボタンをクリックすると、ここで入力したテキストが表示されます。</p> <p>このフィールドには HTML を指定できます。</p>

22. 「次へ」をクリックします。

23. 必要に応じて、チャートまたはチャートのカスタマイズ・フォームのランタイム作成中に、様々な時点で実行する PL/SQL コードを入力します。

表 A-33 に、このページで使用可能なオプションをリストし、これらのオプションについて説明します (図 A-63)。

図 A-63 チャートの追加 PL/SQL コードのオプション

SAMPLE_CHART

ステップ 10 / 10

追加PL/SQLコード

チャートおよびカスタマイズ・フォームを作成するHTMLロード実行時に、様々な箇所で行われるPL/SQLコードを入力します。

	チャート	カスタマイズ・フォーム
...ページ表示前		
...ヘッダー表示後		
...フッター表示前		
...ページ表示後		

表 A-33 「追加 PL/SQL コード」 ページで使用可能なオプション

オプション	説明
... ページ表示前	チャートまたはチャートのカスタマイズ・フォームを含むページが表示される前に実行する PL/SQL プロシージャを入力します。
... ヘッダー表示後	チャートまたはチャートのカスタマイズ・フォームのヘッダーが表示された後に実行する PL/SQL プロシージャを入力します。 PL/SQL プロシージャからの出力をチャートに表示する場合は、「... ヘッダー表示後」 フィールドに PL/SQL を入力する必要があります。
... フッター表示前	チャートまたはチャートのカスタマイズ・フォームのフッターが表示される前に実行する PL/SQL プロシージャを入力します。
... ページ表示後	チャートまたはチャートのカスタマイズ・フォームを含むページが表示された後に実行する PL/SQL プロシージャを入力します。

24. 「完了」をクリックします。

「完了」をクリックすると、結果をテストできるサマリー・ページが表示されます (図 A-64)。結果をテストする方法については、A-105 ページの「ポートレットに対するテスト・ランの実施」を参照してください。

図 A-64 チャートのサマリー・ページ

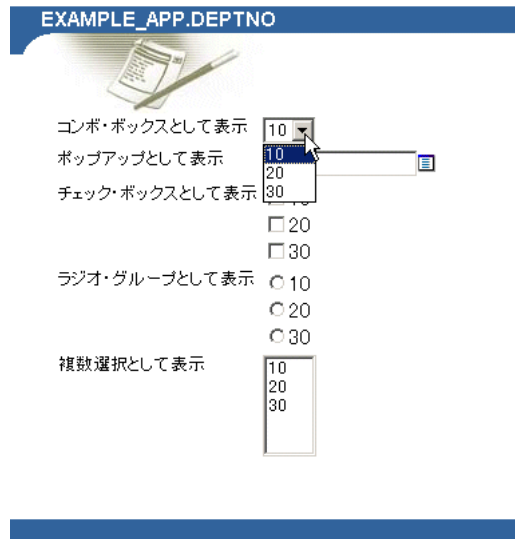
The screenshot shows the Oracle Portal interface for a chart named 'SAMPLE_CHART'. At the top, there are tabs for '開発' (Development), '管理' (Management), and 'アクセス' (Access). Below the tabs, the breadcrumb path is '開発: EXAMPLE_APP.SAMPLE_CHART'. The main content area displays the following information:

- チャート: SAMPLE_CHART
- Portal DBプロバイダ: EXAMPLE_APP
- アーカイブ・バージョン
- 本稼働バージョン状態: 1 (PRODUCTION with VALID Package)
- 最終更新: 金曜日 2004,06/25 14:14:39 by ORCLADMIN
- 実行リンク: PORTAL_DEMO.SAMPLE_CHART.show
PORTAL_DEMO.SAMPLE_CHART.show_parms
- PL/SQLソース: [パッケージ仕様](#), [パッケージ本体](#)
- コール・インタフェース: [表示](#)

At the bottom of the page, there is a row of icons and links: '編集' (Edit), '新規として編集' (Edit as New), '実行' (Execute), 'ポートレットとして実行' (Execute as Portlet), 'カスタマイズ' (Customize), 'お気に入り追加' (Add to Favorites), '情報' (Info), and '削除' (Delete). A '開じる' (Close) button is located in the bottom right corner.

宣言による値リストの構築

図 A-65 動的値リストの異なる表示タイプ



OracleAS Portal の値リスト・ウィザードを使用して値リストを構築する場合、2 種類のリストを作成するオプションがあります。

- 動的値リスト
- 静的値リスト

動的値リストは、ウィザードを介して構築する SELECT 文からリストの内容を取得します。リストの内容は、ソース・データベースから選択可能なデータによって異なります。静的値リストは、作成者がウィザードに入力した値からリストの内容を取得します。リストを編集して変更しないかぎり、この値は変わりません。

この項では、動的値リストの作成方法について説明します。

動的値リストを作成する手順は、次のとおりです。

1. A-16 ページの「宣言によるポートレットの構築」に記載されている手順に従います。手順 6 を完了した後、この項に戻ります。
2. 「新規作成 ...」の横にある「値リスト」リンクをクリックします (図 A-66)。

図 A-66 「新規作成 ...」の横にある「値リスト」リンク

これらは、使用可能なポートレットおよび操作です。

新規作成... [フォーム](#), [レポート](#), [チャート](#), [カレンダー](#), [動的ページ](#), [XMLコンポーネント](#), [階層](#), [メニュー](#), [URL](#), [フレーム・ドライバ](#), [リンク](#), [値リスト](#), [データ・コンポーネント](#)

パス: [プロバイダ](#) > [ローカルで構築されたプロバイダ](#) > **Example Application**

- 「値リスト」 ページで、「動的値リスト」 をクリックします (図 A-67)。

図 A-67 「動的値リスト」リンク

値リスト

値リストを構築して、Portal DBプロバイダまたはコンポーネントの入力フィールドの選択肢をエンド・ユーザーに提供します。値リストは、静的リストを基にすることも、動的SQL問合せから生成することもできます。

[動的値リスト](#)

リストを定義するSQL問合せを使用して、動的値リストを作成します。

[静的値リスト](#)

静的リストに値を入力して、静的値リストを作成します。

- 表示されたページで、「所有者」 ドロップダウン・リストから、このリストを所有する Portal DB プロバイダを選択します (図 A-68)。

図 A-68 値リスト・オプションの定義

EXAMPLE_APP.LOV_0714193552 ?

[適用] [OK] [取消]

データベースからリストへの移入にSQL問合せを使用して、動的な値リストを作成します。リストを所有する Portal DBプロバイダの名前を選択し、値リストを識別する名前を入力します。形式と、NULL値を表示するかどうかを選択してから、表またはビューの2つの列を選択するSQL問合せを入力します。

所有者 ▼

名前

デフォルト書式 ▼

NULL値の表示 ▼

SQL問合せ。バインド変数を入力するには、先頭にコロン(:)を付けます。

```
select distinct deptno, deptno
from PORTAL_DEMO.EMP|
```

構文:
 select [display_column], [return_column] from [table]
 ここで
 display_columnは値リストに表示される選択可能な値を識別します。
 return_columnは値リストを使用するコンポーネントまたはカスタマイズ・フォームに渡される実際の値を識別します。 SQL問合せ。バインド変数を入力するには、先頭にコロン(:)を付けます。

5. 「名前」フィールドに、この値リストの名前を入力します (図 A-68)。
この名前は、値リストを挿入する際に選択候補リストに表示されます。
6. 「デフォルト書式」ドロップダウン・リストから、値リストのデフォルト書式を選択します (図 A-68)。

オプションは次のとおりです。

- **チェック・ボックス**: チェック・ボックスをマークして選択したことを表します。
- **コンボ・ボックス**: ドロップダウン・リストから選択するか、または値を手動で入力します。
- **ポップアップ**: 選択した値に関連するフィールドが自動的に設定された補助ウィンドウから値を選択します。
- **ラジオ・グループ**: 選択する対象の○ (ラジオ・ボタン) をマークします。
- **複数選択**: リストから1つ以上の値を選択します。

値リストをポートレットに追加するポートレット開発者は、このデフォルトを上書きして異なる書式でリストを表示できます。

7. 「NULL 値の表示」ドロップダウン・リストから「はい」、「いいえ」または「%」を選択することで、NULL 値を表示するかどうかを指定します (図 A-68)。
- 「はい」を選択すると、NULL 値は値リストに Null で表示されます。
 - 「いいえ」を選択すると、NULL 値は値リストに空白で表示されます。
 - 「%」を選択しても空白で表示されます。「%」と「いいえ」の違いは、% では値リストをフォームに配置する際、動的なトランザクションの設定に NULL 値 (% ワイルドカード) を使用できることです。
8. 「SQL 問合せ」テキスト・ボックスに、SQL の SELECT 文を入力します (図 A-68)。
- 表またはビューの 2 つの列から 2 つの値を選択します (同じ列から 2 つの値を選択することも可能です)。
- 第 1 の列は値リストに表示される値を指定します。
 - 第 2 の列は、ポートレットに渡される実際の値を指定します。
- たとえば、`select ename, empno from scott.emp` という問合せでは、ENAME 列から従業員名を表示する値リストが作成されます。ユーザーがこの値リストから名前を選択すると、従業員の対応する ID 番号 (EMPNO) がポートレットに渡されます。
- 問合せの最後にはセミコロンを付けません。

注意： 値リストをフォーム・ポートレットで使用する場合のみ、コロン (:) を前に付けてバインド変数を入力できます。バインド変数を含む値リストは、他の種類の OracleAS Portal のポートレットでは機能しません。

SQL 問合せでは、OracleAS Portal スキーマおよびアプリケーション・スキーマに定数を使用できます。詳細は、A-111 ページの「[OracleAS Portal スキーマの参照](#)」を参照してください。

9. 「OK」をクリックして変更内容を保存します。
- 「OK」をクリックすると、値リストを管理できるサマリー・ページが表示されます (図 A-69)。たとえば、このページからリストを実行して、リストを表示できる様々な方法を確認できます。

図 A-69 値リストのサマリー・ページ



完成した値リストはすぐに使用できます。値リストを追加できる位置では、作成した値リストが選択候補リストに自動的に表示されます。たとえば、ポートレット・ビルダー・ウィザードで構成したフォームとレポートのカスタマイズ・フォームの両方で、値リストを追加できるようになります。これらのウィザードを完了すると、ここで作成したリストがすぐに選択オプションとして表示されます。

ポートレット・ビルダーのコンポーネントの編集

ウィザードを使用してポートレットを構築後、ナビゲータのオプションを使用してポートレットを編集できます。ポートレットは、編集して変更内容を現在のバージョンに上書きできます。あるいは、ポートレットを新しいポートレットとして編集して新しいバージョンのポートレットを作成できます（したがって、元の古いバージョンのポートレットは未変更のまま残ります）。

ウィザードではポートレットの作成を通じて各ステップが表示されましたが、エディタでは、ウィザードで使用可能であったオプションが複数のタブとして提供されます。各タブは、ポートレットを作成したウィザードの各ステップに対応しています。各タブの入力フィールドには、ポートレットの作成で指定した値、またはポートレットを最後に編集したユーザーが指定した値が挿入されます。

編集中は、他のユーザーがポートレットを変更しないように、ポートレットがロックされます。ポートレットは、「編集」ページで「OK」をクリックするまでロックされたままです。詳細は、A-98 ページの「[ポートレットに対するロックの管理](#)」を参照してください。

注意： ポートレットを編集するには、ポートレットまたはポートレットを所有するプロバイダに対して少なくとも編集権限が付与されている必要があります。

ポートレットを編集する手順は、次のとおりです。

1. 「Portal ビルダー」 ページ上部の「**ナビゲータ**」リンクをクリックします。
2. 「**プロバイダ**」タブをクリックして前面に表示します。
3. 「**ローカルで構築されたプロバイダ**」リンクをクリックします。
権限が付与されているすべてのプロバイダ名が「**名前**」列に表示されます。
4. 編集するポートレットが含まれている OracleAS Portal データベース・プロバイダの名前をクリックします。
5. ポートレットの名前をクリックします。
6. 表示されたページで、「**開発**」タブを展開します。タブの下部にある「**編集**」リンクをクリックします。

元のバージョンのポートレットを変更せずに残すには、「**編集**」ではなく「**新規として編集**」リンクをクリックします。

7. 1つ以上のタブをクリックして先に進み、対応するオプションを編集します。

タブには、ウィザードで使用可能な同じオプションが含まれています。これらのオプションの詳細は、ポートレットの構築に関する前述の項を参照してください。

- [宣言によるフォームの構築](#)
- [宣言によるレポートの構築](#)
- [宣言によるチャートの構築](#)
- [宣言による値リストの構築](#)

タブを切り替えるたびに、OracleAS Portal によって変更内容が記録されます。各タブで必要なすべての変更が終了した場合のみ、「**OK**」をクリックします。編集内容を保存しない場合は、「**取消**」をクリックします。

「**OK**」または「**取消**」のいずれかをクリックすることが重要です。単純にウィンドウを閉じると、コンポーネントがロックされたままになります。不適切にロックされたポートレットのロックを解除するには、特別な対策が必要です。詳細は、A-98 ページの「[ポートレットに対するロックの管理](#)」を参照してください。

ポートレットの管理

ローカルで構築された各ポートレットについては、OracleAS Portal により、管理タスク（管理ページ）を実行できる中央の位置が提供されます。この項では、ローカルで構築されたポートレットの管理ページを検索し、様々な管理タスクを実行する方法について説明します。この項の内容は、次のとおりです。

- [コンポーネント管理ページへのナビゲート](#)
- [ポートレット名の変更](#)
- [ポートレットの削除](#)
- [ポートレットのコピー](#)
- [ポートレットに対する PL/SQL パッケージの生成](#)

注意： ポートレットの名前を変更、ポートレットを削除またはコピーするには、ポートレットを所有するプロバイダに対して少なくとも編集権限が付与されている必要があります。

ポートレットを生成するには、ポートレットまたはポートレットを所有するプロバイダに対して少なくともソースの表示権限が付与されている必要があります。

コンポーネント管理ページへのナビゲート

コンポーネント管理ページにナビゲートする手順は、次のとおりです。

1. 「Portal ビルダー」 ページ上部の「**ナビゲータ**」リンクをクリックします。
2. 「**プロバイダ**」 タブをクリックして前面に表示します。
3. 「プロバイダ」 タブで、「**ローカルで構築されたプロバイダ**」リンクをクリックします。
4. 管理するポートレットを所有しているデータベース・プロバイダの名前をクリックし、管理するコンポーネントの横にある「**管理**」リンクをクリックします。

このページには3つのタブがあります。

- **開発：**コンポーネントの様々なバージョンを実行、テスト、改訂、削除および管理します。
- **管理：**コンポーネントをエクスポート、コピー、名前変更（内部名と表示名）および生成し、コンポーネントのパフォーマンスを監視します。
- **アクセス：**OracleAS Portal フレームワーク内のコンポーネントのセキュリティを定義します。このタブには、ポートレットとしてのコンポーネントの公開、プロバイダからの権限の継承、およびキャッシュの失効化も含まれています。

ポートレット名の変更

ポートレット名を変更する手順は、次のとおりです。

1. 名前を変更するポートレットのコンポーネント管理ページにナビゲートします。
このページへのナビゲート方法については、A-94 ページの「[コンポーネント管理ページへのナビゲート](#)」を参照してください。
2. 「**管理**」タブをクリックして前面に表示します。
3. 「**名前の変更**」アイコンをクリックします。
4. ポートレットの内部名または表示名（あるいはその両方）を変更します。
 - a. 内部名の場合は、「**新規コンポーネント名**」フィールドにポートレットの新しい名前を入力します。
 - b. 表示名の場合は、「**新規コンポーネントの表示名**」フィールドにポートレットの新しい表示名を入力します。
5. 「**OK**」をクリックします。
ポートレット名が、指定した名前に変更されます。

ポートレットの削除

データベースからポートレットを削除する手順は、次のとおりです。

1. 削除するポートレットのコンポーネント管理ページにナビゲートします。
このページへのナビゲート方法については、A-94 ページの「[コンポーネント管理ページへのナビゲート](#)」を参照してください。
2. 「**開発**」タブをクリックして前面に表示します。
3. 「**削除**」アイコンをクリックします。
「ポートレットの削除」ページに、削除する権限を付与されているポートレットのバージョンが表示されます。
4. 削除するポートレットの各バージョンの横にあるチェック・ボックスを選択します。
5. 「**はい**」をクリックします。

ポートレットのコピー

ポートレットをコピーする手順は、次のとおりです。

1. コピーするポートレットのコンポーネント管理ページにナビゲートします。
このページへのナビゲート方法については、A-94 ページの「[コンポーネント管理ページへのナビゲート](#)」を参照してください。
2. 「管理」タブをクリックして前面に表示します。
3. 「コピー」アイコンをクリックします。
ポートレットのコピー・ページが表示されます。
4. 「新規所有者」リストから、新しくコピーしたポートレットを所有するプロバイダを選択します。
元のポートレットと同じプロバイダを選択してもかまいません。
5. ポートレットに新しい内部名および表示名を指定します。
 - a. 内部名の場合は、「新規コンポーネント名」フィールドにポートレットの新しい名前を入力します。
 - b. 表示名の場合は、「新規コンポーネントの表示名」フィールドにポートレットの新しい表示名を入力します。
6. 「OK」をクリックします。

ポートレットに対する PL/SQL パッケージの生成

ポートレット・コードがエラーなしでコンパイルされ、適切な時間内で実行されることを確認するには、この機能を使用します。この機能は、ポートレット・ビルダー・ウィザードで現在サポートしている範囲を超える複雑なコードを記述している場合は特に有用です。たとえば、パッケージを作成し、そのパッケージをフォームまたはレポートのベースとし、この機能を使用してフォームまたはレポートがコンパイルされることを確認できます。

ポートレットに対して PL/SQL パッケージを生成する手順は、次のとおりです。

1. 該当するポートレットのコンポーネント管理ページにナビゲートします。
このページへのナビゲート方法については、A-94 ページの「[コンポーネント管理ページへのナビゲート](#)」を参照してください。
2. 「管理」タブをクリックして前面に表示します。
3. 「生成」アイコンをクリックします。
最新の OracleAS Portal ルーチンを使用して PL/SQL パッケージが生成されます。

ソース・コードの表示

OracleAS Portal では、ポートレットが PL/SQL パッケージ・プロシージャとしてデータベースに格納されます。ポートレットのパッケージ仕様、本体およびコール・インタフェースを表示できます。

ポートレットのコール・インタフェースには、実行時に設定できる引数が表示されます。たとえば、レポートを作成し、レポート構築ウィザードの「表示オプション」ステップで、「デフォルト書式」に「HTML」を選択した場合、コール・インタフェースには、`_format_out` 引数のデフォルト値として HTML が表示されます。

ポートレットが含まれているパッケージを PL/SQL で実行、または URL からコールして実行する場合は、異なる引数を受け入れるようにコール・インタフェースを編集できます。`_format_out` 引数を、ASCII など、別のレポート出力書式に変更できます。

この項では、ポートレットのパッケージ仕様、本体およびコール・インタフェースを表示する方法について説明します。この項の内容は、次のとおりです。

- [ポートレットのパッケージ仕様および本体の表示](#)
- [ポートレットのコール・インタフェースの表示](#)

注意： ポートレットのソース・コードを表示するには、ポートレットまたはポートレットを所有するプロバイダに対して少なくともソースの表示権限が付与されている必要があります。

ポートレットのパッケージ仕様および本体の表示

ポートレットのパッケージ仕様および本体を表示する手順は、次のとおりです。

1. 該当するポートレットのコンポーネント管理ページにナビゲートします。
このページへのナビゲート方法については、A-94 ページの「[コンポーネント管理ページへのナビゲート](#)」を参照してください。
2. 「**開発**」タブをクリックして前面に表示します。
3. ポートレットの仕様を表示するには、「**PL/SQL ソース**」の横にある「**パッケージ仕様**」リンクをクリックします。本体を表示するには、「**PL/SQL ソース**」の横にある「**パッケージ本体**」リンクをクリックします。

ポートレットのコール・インタフェースの表示

ポートレットのコール・インタフェースを表示する手順は、次のとおりです。

1. 該当するポートレットのコンポーネント管理ページにナビゲートします。
このページへのナビゲート方法については、A-94 ページの「[コンポーネント管理ページへのナビゲート](#)」を参照してください。
2. 「**開発**」タブをクリックして前面に表示します。
3. 「**コール・インタフェース**」の横にある「**表示**」リンクをクリックします。

ポートレットに対するロックの管理

ポートレットをロックすることで、ポートレットの編集中に他の開発者が変更内容を上書きできないようにします。ポートレットがロックされている間、他の開発者はそのポートレットを編集できません。編集中はポートレットが自動的にロックされます。この項に概説されている手順を実行することで、編集している担当者を判断できます。

注意： ポートレットのロックを確認、またはポートレットのロックを解除するには、ポートレットまたはポートレットを所有するプロバイダに対して少なくとも編集権限が付与されている必要があります。

ポートレットをロックしたユーザー名を表示する手順は、次のとおりです。

1. 該当するポートレットのコンポーネント管理ページにナビゲートします。
このページへのナビゲート方法については、A-94 ページの「[コンポーネント管理ページへのナビゲート](#)」を参照してください。
2. 「**管理**」タブをクリックして前面に表示します。
3. 「**このコンポーネントのロックを表示**」リンクをクリックします。

表示されたページで、ロックされているポートレットは「**状態**」が EDIT と表示されます。このページには、状態に加え、ロックの責任を負うユーザー、そのバージョンがロックされている時間および他の情報も表示されます。「**ロック解除**」リンクも用意されています。クリックするとポートレットのロックを解除できます。

「**ロック解除**」リンクは、開発者がコンポーネントを編集し、「**OK**」または「**取消**」をクリックせずにブラウザ・ウィンドウを閉じた場合に役立ちます。このような場合は、コンポーネントがロックされたままになります。「**ロック解除**」リンクを使用して、コンポーネントのロックを解除します。

バージョンの管理

OracleAS Portal では、同じポートレットの複数のバージョンをデータベースに格納できます。「開発」タブの「新規として編集」リンク（A-94 ページの「コンポーネント管理ページへのナビゲート」を参照）を使用し、現在の本稼働バージョンに基づいてポートレットの新規バージョンを作成します。編集内容を保存する場合は、既存のバージョンを上書きするのではなく、新しいバージョンに保存します。次回ポートレットを編集する際は、最新のバージョンまたは以前のバージョンのポートレットを開くことができます。この項では、OracleAS Portal コンポーネントの様々なバージョンへのアクセス方法について説明します。

OracleAS Portal コンポーネントの様々なバージョンにアクセスする手順は、次のとおりです。

1. 該当するポートレットのコンポーネント管理ページにナビゲートします。
このページへのナビゲート方法については、A-94 ページの「コンポーネント管理ページへのナビゲート」を参照してください。
2. 「開発」タブをクリックして前面に表示します。
3. 本稼働バージョンにアクセスするには、次のようにします。
 - 既存の本稼働バージョンを変更する場合は、「編集」リンクをクリックします。
 - 本稼働バージョンに基づく新しいバージョンを変更する場合は、「新規として編集」リンクをクリックします。

「開発」タブの「本稼働バージョン状態」の横に、本稼働バージョンがリストされます。このフィールドには、本稼働バージョンおよびこのバージョンのコード・パッケージが有効（つまり、動作する）かどうかが表示されます。
4. コンポーネントのアーカイブ・バージョンにアクセスするには、次の手順を実行します。
 - a. 「アーカイブ・バージョン」の横の該当する x （アーカイブ）バージョンをクリックします。
変数値 (x) はアーカイブ番号を表します。
 - b. 表示されたページで、「はい」をクリックし、選択したバージョンを本稼働バージョンにします。
「本稼働バージョン状態」の横のバージョン番号が、選択したバージョンに変更されます。
 - c. 「開発」タブで「編集」または「新規として編集」リンクをクリックし、新しく選択した本稼働バージョンのコンポーネントを編集します。

ポートレットの各バージョンは、データベースからいつでも削除できます。A-95 ページの「[ポートレットの削除](#)」の手順に従って、削除するバージョンの横のチェック・ボックスを選択し、「はい」をクリックします。本稼働バージョンを削除し、アーカイブ・バージョンはそのままにした場合は、「開発」タブでアーカイブ・バージョンを選択し、それを本稼働バージョンにすることが必要になる場合があります。

ポートレットの最新バージョンは、本稼働バージョンと呼ばれます。ポートレットが含まれているデータベース・パッケージがエラーなしで実行されるかどうかに従って、本稼働バージョンは有効または無効になります。ポートレットのバージョン状態が **(PRODUCTION with INVALID package)** の場合は、ポートレットを生成 (A-96 ページの「[ポートレットに対する PL/SQL パッケージの生成](#)」を参照) またはポートレットを編集 (A-92 ページの「[ポートレット・ビルダーのコンポーネントの編集](#)」を参照) し、エラーを修正してから実行する必要があります。

ポートレットのセキュリティ管理

ナビゲータの「操作」列 (図 A-70) には、ポートレットに対して実行できる操作が表示されます。ここにリストされる操作は、ポートレットに対して付与されているアクセス権限によって異なります。

図 A-70 ポートレットの操作

タイプ ▲▼	名前 ▲▼	操作	作成者 ▲▼	最終変更 ▲▼ ?
SQL問合せからのレポート	Detail Report	管理 , 実行 , 編集 , 削除 , アクセス権限付与 , エクスポート	PEOPLE_APP	2004-03-26
階層	Organization Chart	管理 , 実行 , 編集 , 削除 , アクセス権限付与 , エクスポート	PEOPLE_APP	2004-03-26
動的ページ	People Search	管理 , 実行 , 編集 , 削除 , アクセス権限付与 , エクスポート	PEOPLE_APP	2004-03-26
SQL問合せからのレポート	People Search Report	管理 , 実行 , 編集 , 削除 , アクセス権限付与 , エクスポート	PEOPLE_APP	2004-03-26
SQL問合せからのレポート	Person Details	管理 , 実行 , 編集 , 削除 , アクセス権限付与 , エクスポート	PEOPLE_APP	2004-03-26
ウィザードからのリンク	ORG_LINK	管理 , 実行 , 編集 , 削除 , アクセス権限付与 , エクスポート	PEOPLE_APP	2004-03-26

デフォルトでは、ポートレットの作成者にはポートレットに対する管理権限が付与されます。管理権限では、最高レベルのアクセスが可能です。これは、ポートレットを実行、編集、削除またはエクスポートできることを意味します。作成したポートレットに対するアクセス権限を別のユーザーに付与することもできます。

OracleAS Portal でポートレットを構築するには、完成したポートレットを所有するプロバイダに対して少なくとも編集権限が付与されている必要があります。デフォルトでは、ポートレットは、それを所有するプロバイダに割り当てられている権限を継承します。たとえば、コンポーネントが MY_APP プロバイダで作成され、そのコンポーネントがポートレットとして公開されると、MY_APP に対して少なくとも実行権限が付与されているすべての開発者はそのコンポーネントを実行できます。ポートレットの所有者は、これらの権限を無視し、プロバイダ・レベルではなく、各ユーザー・レベルでアクセス権限を設定できます。

表 A-34 に、ポートレットに対して実行できる操作をリストし、それらの操作を実行できる権限を示します。

表 A-34 ポートレットの操作と関連付けられているポートレットの権限レベル

操作	管理	編集	ソースの表示	カスタマイズ	実行
他のユーザーへのポートレット 権限の付与	X				
ポートレットのロック管理	X	X			
ポートレット・バージョンの 編集	X	X			
ポートレットのデータベース からの削除	X	X			
ポートレット名の変更	X	X			
別のデータベースへの ポートレットのエクスポート	X	X			
ポートレットのコピー	X	X			
ポートレット PL/SQL パッケージの生成	X	X	X		
ポートレット使用状況の監視	X	X	X		
ポートレットのコール・ インタフェース、パッケージ 仕様および本体の表示	X	X	X		
ポートレットのカスタマイズ	X	X	X	X	
ポートレットの実行	X	X	X	X	X
お気に入りリストへの ポートレットの追加	X	X	X	X	X

ポートレットに対するアクセス権限の付与

注意： ポートレットに対するアクセス権限を他のユーザーまたはグループに付与するには、ポートレットまたはポートレットを所有するプロバイダに対して少なくとも管理権限が付与されている必要があります。

ポートレットに対するアクセス権限は、プロバイダ・レベルおよびポートレット・レベルで定義できます。この項では、プロバイダに対するアクセス権限を継承、またはポートレット・レベルで権限を設定してプロバイダのアクセス権限を無視する方法について説明します。この項の内容は、次のとおりです。

- [ポートレットに対するアクセス権限のプロバイダからの継承](#)
- [個々のユーザーに対するアクセス権限の付与](#)

ポートレットに対するアクセス権限のプロバイダからの継承

デフォルトでは、ポートレットに対するアクセス権限は、そのポートレットを所有するプロバイダから継承されます。たとえば、所有しているプロバイダに対する実行権限が付与されているすべてのユーザーは、自動的にポートレットを実行できます。プロバイダの編集権限が付与されているユーザーは、プロバイダが所有するすべてのポートレットを編集できます。プロバイダに対するアクセス権限は、「[権限付与](#)」タブを介して設定できます（詳細は、A-5 ページの「[データベース・オブジェクトに対する権限の付与および取消し](#)」を参照）。「権限付与」タブで、異なるユーザーのグループに異なる権限を付与できます。プロバイダに対する権限を設定した後、そのプロバイダの下で作成されるポートレットが必ず同じ権限のセットを継承するように指定できます。この項では、ポートレットがそのホスト・プロバイダに対して付与されている権限を継承するように指定する方法を説明します。

注意： ポートレット・ビルダーのコンポーネントをポートレットとして公開するには、その前にホスト・プロバイダを **OracleAS Portal** にプロバイダとして公開する必要があります。これを設定する方法については、A-11 ページの「[プロバイダの公開](#)」を参照してください。

ポートレットに対するアクセス権限をプロバイダから継承する手順は、次のとおりです。

1. 「Portal ビルダー」 ページ上部の「[ナビゲータ](#)」リンクをクリックします。
2. 「[プロバイダ](#)」タブをクリックして前面に表示します。
3. 「[ローカルで構築されたプロバイダ](#)」リンクをクリックします。
4. アクセス権限を付与するポートレットが含まれているデータベース・プロバイダの名前をクリックします。

5. 目的のポータルレットの横にある「**管理**」リンクをクリックします。
「管理」ページが表示されます。
6. 「**アクセス**」タブをクリックします。
7. 「**プロバイダから権限を継承**」チェック・ボックスを選択します。
ポータルレットに対するアクセス権限が、ポータルレットのホスト・プロバイダによって定義されます。

個々のユーザーに対するアクセス権限の付与

ポータルレット・レベルでアクセス権限を設定することで、デフォルトのアクセス権限（つまり、プロバイダを介して付与されたアクセス権限）を無視できます。この項では、個々のユーザーに対して権限を明確に指定できるように、ポータルレット・レベルでアクセス権限を設定する方法について説明します。

アクセス権限を個々のユーザーに付与する手順は、次のとおりです。

1. 「Portal ビルダー」 ページ上部の「**ナビゲータ**」リンクをクリックします。
2. 「**プロバイダ**」タブをクリックして前面に表示します。
3. 「**ローカルで構築されたプロバイダ**」リンクをクリックします。
4. アクセス権限を付与するポータルレットが含まれているデータベース・プロバイダの名前をクリックします。
5. 目的のポータルレットの横にある「**管理**」リンクをクリックします。
「管理」ページが表示されます。
6. 「**アクセス**」タブをクリックします。
7. 「**プロバイダから権限を継承**」チェック・ボックスの選択を解除し、「**適用**」をクリックします。
「アクセス」ページに、「**アクセス権限の付与**」および「**アクセス権限の変更**」の2つのセクションが表示されます。
8. 「**ユーザー/ロール**」フィールドに、権限を付与するユーザーまたはグループの名前を入力します。
ユーザーまたはグループの正確な名前が不明の場合は、「**ユーザーをブラウズ**」または「**グループをブラウズ**」アイコンをクリックし、表示されたリストから選択します。
9. 権限のリストから、ユーザーまたはグループに付与するアクセス権限を選択します。
異なるアクセス権限で許可される操作の詳細は、[表 A-34](#) を参照してください。

10. 「追加」をクリックします。

「ユーザー/ロール」フィールドに入力したユーザーまたはグループが、付与したアクセス権限とともに、ページ下部の「**アクセス権限の変更**」セクションに表示されます。

注意： OracleAS Portal では、ユーザーおよびグループのリポジトリとして機能する Oracle Internet Directory (OID) が識別管理に使用されます。OID では、グループが識別名 (DN) によって一意に識別されます。各グループには一意の DN が割り当てられます。ただし、完全に系統が異なる 2 人の個人 (つまり、John Smith と John Doe) が一般名を共有できるのと同じ方法で、複数のグループも一般名を共有できます。ポータル内で使用する場合、そのポータル内部で作成されたグループは、単に一般名で表示されます。ただし、ポータルが OID で他の場所のグループ (同じ Identity Management Infrastructure に関連付けられている他のポータルのグループなど) を参照する場合は、ローカルに定義されているポータルのグループと区別するために、グループの DN が表示されます。

11. (オプション) ユーザーまたはグループのアクセス権限を変更する場合は、「**アクセス権限の変更**」セクションのユーザーまたはグループの横で、新しい権限を選択します。
12. (オプション) すべての権限を削除する場合は、「**アクセス権限の変更**」セクションのユーザーまたはグループの横にある「**削除**」アイコンをクリックします。
13. 「適用」をクリックします。
14. 「閉じる」をクリックして「管理」ページを終了し、Portal ナビゲータに戻ります。

ポータルレットに対するテスト・ランの実施

Portal DB プロバイダでコンポーネントを作成した後は、コンポーネントを実行し、正しく作成されているかどうかを確認し、必要な場合は不適切な点を編集する必要があります。この機能のすべてを「管理」ページの「開発」タブから利用できます。このタブから、フル・ページとして、ポータルレットとして、およびコンポーネントのカスタマイズ・フォームを介してコンポーネントを実行できます。

また、ポータルレットのカスタマイズ・フォームを介してコンポーネントをポータルレットとして実行するオプションもあります。ただし、このオプションは「開発」タブからは使用できません。これを実行するには、最初にコンポーネントをポータルレットとして公開し、そのポータルレットをページ上に配置してから、ポータルレット・ヘッダーの「カスタマイズ」リンクをクリックする必要があります。

この項では、様々な方法を使用してコンポーネントを実行する方法について説明します。この項の内容は、次のとおりです。

- [フル・ページとしてのコンポーネントの実行](#)
- [ポータルレットとしてのコンポーネントの実行](#)
- [カスタマイズ・フォームを介したコンポーネントの実行](#)
- [ポータルレットのカスタマイズ・フォームを介したポータルレットとしてのコンポーネントの実行](#)

注意： ポータルレットを実行するには、ポータルレットまたはポータルレットを所有するプロバイダに対して少なくとも実行権限が付与されている必要があります。「管理」ページの「アクセス」タブで、「ポータルレットとして公開」チェック・ボックスが選択されていることも確認する必要があります。

有効なポータルレット・バージョンの場合のみ、ポータルレットを実行できません。詳細は、A-99 ページの「バージョンの管理」を参照してください。

フル・ページとしてのコンポーネントの実行

コンポーネントをフル・ページとして実行する手順は、次のとおりです。

1. 「Portal ビルダー」 ページ上部の「ナビゲータ」リンクをクリックします。
2. 「プロバイダ」 タブをクリックして前面に表示します。
3. 「ローカルで構築されたプロバイダ」リンクをクリックします。
4. ポートレットを所有するデータベース・プロバイダの名前をクリックします。
5. 実行するコンポーネントの横にある「管理」リンクをクリックします。
「管理」 ページの「開発」 タブが表示されます。
6. 「実行」 をクリックします。
別のブラウザ・ウィンドウに、コンポーネントがフル Web ページで表示されます。

ポートレットとしてのコンポーネントの実行

コンポーネントをポートレットとして実行する手順は、次のとおりです。

1. 「Portal ビルダー」 ページ上部の「ナビゲータ」リンクをクリックします。
2. 「プロバイダ」 タブをクリックして前面に表示します。
3. 「ローカルで構築されたプロバイダ」リンクをクリックします。
4. ポートレットを所有するデータベース・プロバイダの名前をクリックします。
5. 実行するコンポーネントの横にある「管理」リンクをクリックします。
「管理」 ページの「開発」 タブが表示されます。
6. 「ポートレットとして実行」 をクリックします。
別のブラウザ・ウィンドウに、ポートレットがフル・ページよりも小さい形式で表示されます。

カスタマイズ・フォームを介したコンポーネントの実行

カスタマイズ・フォームを介してコンポーネントを実行する手順は、次のとおりです。

1. 「Portal ビルダー」 ページ上部の「ナビゲータ」リンクをクリックします。
2. 「プロバイダ」タブをクリックして前面に表示します。
3. 「ローカルで構築されたプロバイダ」リンクをクリックします。
4. ポートレットを所有するデータベース・プロバイダの名前をクリックします。
5. 実行するコンポーネントの横にある「管理」リンクをクリックします。
「管理」ページの「開発」タブが表示されます。
6. 「カスタマイズ」をクリックします。
カスタマイズ・フォームが表示されます。
7. 設定をカスタマイズし、「実行」ボタンをクリックします。
別のブラウザ・ウィンドウに、コンポーネントがフル Web ページで表示されます。

ポートレットのカスタマイズ・フォームを介したポートレットとしてのコンポーネントの実行

ポートレットのカスタマイズ・フォームを介してポートレットとしてコンポーネントを実行する手順は、次のとおりです。

1. コンポーネントをポートレットとして公開します。
 - a. 「Portal ビルダー」 ページ上部の「ナビゲータ」リンクをクリックします。
 - b. 「プロバイダ」タブをクリックして前面に表示します。
 - c. 「ローカルで構築されたプロバイダ」リンクをクリックします。
 - d. 該当するコンポーネントを所有しているデータベース・プロバイダをクリックします。
 - e. 該当するポートレットの横にある「管理」リンクをクリックします。
 - f. 「管理」ページで、「アクセス」タブをクリックして前面に表示します。

- g. 「Portal アクセス」ヘッダーの下の「ポートレットとして公開」チェック・ボックスが選択されていることを確認します。

注意：「ポートレットとして公開」チェック・ボックスは、プロバイダが適切に構成されている場合のみ使用可能になります。詳細は、A-11 ページの「プロバイダの公開」を参照してください。

- h. 「閉じる」をクリックして変更内容を保存し、Portal ナビゲータに戻ります。
2. ポートレットをページに追加します。
- ポートレットを配置するページに移動します。
 - ページ上部の「編集」リンクをクリックします。
 - ページのポートレット領域に移動します。
 - 「ポートレットの追加」アイコンをクリックします。
 - ポートレット・リポジトリの検索フィールドにポートレットの表示名を入力し、「Go」をクリックします。

正確な名前が不明な場合は、ポートレット・リポジトリ内でポートレットの位置にドリルできます。通常、新しく作成されたポートレットは「Portlet ステージング領域」ノードの下にあります。ノードをクリックし、次にポートレットのホスト・プロバイダ名をクリックします。
 - ポートレットをクリックして、「選択したポートレット」リストに追加します。
 - 「OK」をクリックしてポートレットを追加し、ポートレットを配置したページに戻ります。
3. ポートレット・ヘッダーの「カスタマイズ」リンクをクリックします。

このリンクが表示されない場合は、このリンクが表示されるようにホスト領域のプロパティを編集します（ページ編集モードで領域上部の「領域の編集」アイコンをクリックします）。
4. 設定をカスタマイズし、「OK」をクリックします。

カスタマイズされたポートレットが表示されます。

バッチ・モードでの実行

OracleAS Portal の開発者は、エンド・ユーザーがバッチ・モードでコンポーネントを実行できるように、カスタマイズ・フォームに「バッチ」ボタンを追加できます。バッチ・モードは非同期のため、ユーザーはトランザクションをバックグラウンドで実行し、システムを他のタスクに解放できます。「バッチ」ボタンは、チャートとレポートのカスタマイズ・フォームに配置できます。バッチ処理は、コンポーネントが大量のデータに基づいている場合や、ポートレットに多数の行のデータが表示されることが予想される場合に便利です。

ジョブをバッチ・モードで実行すると、ジョブがバッチ・キューに送信されたことを示すページが表示され、ジョブを識別する番号が提示されます。

この項では、「バッチ」ボタンを既存のコンポーネントに追加する方法、およびバッチ処理を使用可能にするためにデータベース管理者が事前に `init.ora` に設定する必要があるパラメータに関する情報を提供します。この項の内容は、次のとおりです。

- [バッチ・ジョブのための `init.ora` パラメータの設定](#)
- [既存のコンポーネントへの「バッチ」ボタンの追加](#)

バッチ・ジョブのための `init.ora` パラメータの設定

Oracle Portal のエンド・ユーザーがバッチ・モードでジョブを実行できるように、データベース管理者は、Oracle Portal がインストールされている Oracle データベースの `init.ora` ファイルのパラメータを確認する必要があります。パラメータが正しく設定されていないと、バッチ・ジョブがバッチ・キューに送信されても、バッチ・ジョブが実行されない可能性があります。

[表 A-35](#) に、推奨設定をいくつか示します。

表 A-35 バッチ処理に関する `init.ora` ファイルの推奨設定

<code>init.ora</code> のパラメータおよび設定	指定内容
<code>job_queue_processes=2</code>	2つのバックグラウンド・プロセス
<code>job_queue_intervals=60</code>	60秒ごとにプロセスをウェイクアップ
<code>job_queue_keep_connections=TRUE</code>	切断せずにスリープ

既存のコンポーネントへの「バッチ」ボタンの追加

既存のコンポーネントに「バッチ」ボタンを追加する手順は、次のとおりです。

1. 該当するポートレットのコンポーネント管理ページにナビゲートします。
このページへのナビゲート方法については、A-94 ページの「[コンポーネント管理ページへのナビゲート](#)」を参照してください。
2. 必要な場合は、「**開発**」タブをクリックして前面に表示します。
3. タブの下部にある「**編集**」リンクをクリックします。
4. 「**カスタマイズ・フォームの表示オプション**」タブをクリックして前面に表示します。
このタブは右から3番目にあります。
5. タブの下部にある「**ボタン・オプション**」セクションまでスクロールします。
6. 「**バッチ**」の横のチェック・ボックスを選択します。
7. 必要に応じて、「**バッチ**」ボタンの表示オプションを設定します。
 - a. 「**バッチ**」ボタンの表示名を入力します。
デフォルトは「バッチ」です。
 - b. ボタンの位置を指定します。
「**一番上**」、「**一番上と一番下**」または「**一番下**」から選択します。デフォルトは「一番上」です。
 - c. ボタンの文字位置を指定します。
「**中央揃え**」、「**左揃え**」または「**右揃え**」から選択します。デフォルトは「中央揃え」です。
8. 「**OK**」をクリックして変更内容を保存し、「**管理**」ページに戻ります。
9. 「**閉じる**」をクリックし、Portal ナビゲータに戻ります。

OracleAS Portal スキーマの参照

PL/SQL コードを使用する場合は、OracleAS Portal スキーマ名、アプリケーション・スキーマ名およびアプリケーション名のかわりに定数を使用すると、アプリケーションのインポートおよびエクスポートをより確実に実行できます。表 A-36 に、これらの定数を使用できるコンポーネント、およびポートレット・ビルダー内で使用できる位置を示します。

表 A-36 ポートレット・ビルダーのコンポーネントでのアプリケーションおよびスキーマの定数

コンポーネント	定数	ポートレット・ビルダーでの使用位置
レポート	#APP_SCHEMA#	■ SQL 問合せ
	#PORTAL_SCHEMA#	■ 表名 (#APP_SCHEMA#、#PORTAL_SCHEMA#)
	#APP_NAME#	■ 拡張 PL/SQL セクション
		■ LOV 名 (#APP_NAME#)
チャート	#APP_SCHEMA#	■ SQL 問合せ
	#PORTAL_SCHEMA#	■ 表名 (#APP_SCHEMA#、#PORTAL_SCHEMA#)
	#APP_NAME#	■ LOV 名 (#APP_NAME#)
カレンダー	#APP_SCHEMA#	■ SQL 問合せ
	#PORTAL_SCHEMA#	■ 拡張 PL/SQL セクション
	#APP_NAME#	■ LOV 名 (#APP_NAME#)
フォーム	#APP_SCHEMA#	■ 表名 (#APP_SCHEMA#、#PORTAL_SCHEMA#)
	#PORTAL_SCHEMA#	■ 拡張 PL/SQL セクション
	#APP_NAME#	■ LOV 名 (#APP_NAME#)
動的ページ	#APP_SCHEMA#	■ SQL 問合せ
	#PORTAL_SCHEMA#	■ 拡張 PL/SQL セクション
	#APP_NAME#	■ LOV 名 (#APP_NAME#)
XML	#APP_SCHEMA#	■ SQL 問合せ
	#PORTAL_SCHEMA#	■ 拡張 PL/SQL セクション
	#APP_NAME#	■ LOV 名 (#APP_NAME#)
階層	#APP_SCHEMA#	■ 表名 (#APP_SCHEMA#、#PORTAL_SCHEMA#)
	#PORTAL_SCHEMA#	■ 拡張 PL/SQL セクション
	#APP_NAME#	■ LOV 名 (#APP_NAME#)

表 A-36 ポートレット・ビルダーのコンポーネントでのアプリケーションおよびスキーマの定数 (続き)

コンポーネント	定数	ポートレット・ビルダーでの使用位置
フレーム・ ドライバ	#APP_SCHEMA#	■ SQL 問合せ
	#PORTAL_SCHEMA#	■ 拡張 PL/SQL セクション
	#APP_NAME#	■ LOV 名 (#APP_NAME#)
リンク	#APP_SCHEMA# #PORTAL_SCHEMA#	コンポーネント名
LOV	#APP_SCHEMA# #PORTAL_SCHEMA#	SQL 問合せ

これらの定数は、実行時に置換されます。つまり、コンポーネントの実行時にパラメータとしてコンポーネントに渡されます。ポートレット・ビルダーでコンポーネントを編集する際、使用した定数はそのまま表示されます。つまり、#APP_SCHEMA# のように表示されます。

表 A-37 に、これらの定数の使用例を示します。

表 A-37 定数の使用例

ポートレット・ビルダーでの 位置	例
SQL 問合せ	<pre>Select * from #app_schema#.emp; Select * from #portal_schema#.emp; Select #app_name# from #portal_schema#.emp;</pre>
拡張 PL/SQL セクション	<pre>Htp.p('#APP_NAME#'); Htp.p('#APP_SCHEMA#'); Htp.p('#PORTAL_SCHEMA#'); Select * from #app_schema#.emp; Select * from #portal_schema#.emp; Select #app_name# from #portal_schema#.emp;</pre>
LOV 名	#APP_NAME#.dept_lov
表名	#APP_SCHEMA#.EMP #PORTAL_SCHEMA#.DEPT

追加機能のコーディング

OracleAS Portal のコンポーネントのデータは、SQL 問合せを使用してデータベースから取り出します。チャートおよびレポートを構築する場合は、独自の SQL 問合せを記述するか、または構築ウィザードを使用して生成できます。カレンダーまたはフレーム・ドライバを構築する場合は、独自の SQL 問合せを記述する必要があります。

チャート、カレンダーおよび階層はすべて、これらのポートレットを作成する SQL 文に OracleAS Portal 固有の構文が必要です。たとえば、チャートの SQL 文で、チャートにラベルを提供する表またはビューの列を識別する必要があります。また、チャートのバー・サイズを判断するために、数値データが含まれている列も識別します。チャートおよび他のタイプのコンポーネントを構築するウィザードにより、特別な構文仕様に対するガイドが提供されます。

また、コンポーネントに表示される列値から他のコンポーネントまたは Web ページにジャンプするハイパーテキスト・リンクを SQL 問合せで指定できます。たとえば、部門チャートの従業員名を、チャート上の各従業員に関する情報が含まれている個々のレポートにリンクできます。

この項では、ローカルで構築されたコンポーネントに追加機能を構築するいくつかの方法を紹介します。この項の内容は、次のとおりです。

- [バインド変数の使用](#)
- [フォームのアイテムに対するイベント・ハンドラの記述](#)
- [PL/SQL によるフォームの値の取得および設定](#)
- [PL/SQL によるフォームまたはレポートの Cookie の取得または設定](#)
- [ページ・パラメータを介した値の定義](#)

バインド変数の使用

ローカルで構築されたコンポーネントがカスタマイズ・フォームからのユーザー入力を受け取れるように、SQL 問合せにバインド変数を追加できます。各バインド変数は、コンポーネントのベースとなっている表またはビューの列に対応します。各バインド変数によって、ポートレットのカスタマイズ・フォームに入力フィールドが作成されます。その結果、ユーザーは、コンポーネントに表示するデータを選択できます。

注意： ウィザードを介して宣言的にバインド変数を設定することもできます。この設定方法および OracleAS Portal でページ・パラメータに変数をマップする方法については、A-120 ページの「[ページ・パラメータを介した値の定義](#)」を参照してください。

SQL 問合せに使用するバインド変数は、コロンが前に付く英数文字列 (:var1、:var2、:var3 など) として表されます。たとえば、次の SQL 問合せでは、SCOTT.EMP 表の SALARY 列と DEPT 列に対して入力フィールドが作成されます。

```
select ename, sal, dept
from scott.emp
where sal = :salary and deptno like :dept
```

この SQL 問合せを入力すると、2つのテキスト入力フィールドを備えたカスタマイズ・フォームが作成されます。第1のフィールドでは給与を選択できます。第2のフィールドでは部門番号を選択します。OracleAS Portal は、ユーザーがカスタマイズ・フォームの入力フィールドに入力した値を使用して出力を作成します。

バインド変数を指定するための SQL を開発者が把握する必要はありません。各種ポートレット構築ウィザードの「**カスタマイズ・フォームの表示オプション**」ステップの「**列**」フィールドで、パラメータを受け入れる列を識別できます。

フォームのアイテムに対するイベント・ハンドラの記述

JavaScript および PL/SQL イベント・ハンドラを記述して、フォーム上のアイテムのレイアウトおよび操作をカスタマイズできます。イベント・ハンドラをフォームに正しく組み込むには、JavaScript および PL/SQL に関する詳細な知識が必要です。

- JavaScript イベント・ハンドラを使用して、ボタン、チェック・ボックス、リスト、テキスト・ボックスなど、フォーム・アイテムの動作をカスタマイズできます。使用する JavaScript のバージョンがブラウザでサポートされている必要があります。
- PL/SQL イベント・ハンドラを使用して、ボタンの動作をカスタマイズできます。

同じボタンに対して JavaScript と PL/SQL の両方のイベント・ハンドラを記述する場合は、2つのスクリプトの実行間の潜在的な競合を回避するように注意してください。

OracleAS Portal で使用できるイベントについては、JavaScript または PL/SQL のドキュメントを参照してください。

この項では、ポートレット構築ウィザードを使用してイベント・ハンドラを追加する方法について説明します。この項の内容は、次のとおりです。

- [フォームのアイテムに対する JavaScript イベント・ハンドラの記述](#)
- [フォームのボタンに対する PL/SQL イベント・ハンドラの記述](#)

フォームのアイテムに対する JavaScript イベント・ハンドラの記述

フォーム上のアイテムに対して JavaScript イベント・ハンドラを記述する手順は、次のとおりです。

1. イベント・ハンドラを記述するアイテムが含まれているフォームを編集します。
 - a. 目的のフォーム・コンポーネントの「管理」ページに移動します。

このページへのナビゲート方法については、A-94 ページの「[コンポーネント管理ページへのナビゲート](#)」を参照してください。
 - b. 必要な場合は、「開発」タブをクリックして前面に表示します。
 - c. タブの下部にある「編集」リンクをクリックします。
2. 「書式および妥当性チェック・オプション」タブをクリックして前面に表示します。
3. 左側のフレームで、JavaScript イベント・ハンドラを記述するアイテムの名前をクリックします。
4. 右側のフレームで、「JavaScript イベント・ハンドラ」セクションまでスクロールします。
5. 「JavaScript イベント・ハンドラ」リストから必要なイベントを選択し、イベントが発生したときに実行する JavaScript を入力します。

例 A-1 値を大文字に変換

テキスト・ボックスまたはテキスト領域入力フィールドがフォーカスを失ったときに、フィールドの値を大文字に変換するには、onBlur イベントを選択し、次の JavaScript を入力します。

```
this.value = this.value.toUpperCase();
```

例 A-2 フォームの送信確認

フォームを送信する前に、エンド・ユーザーがボタンをクリックした時点で確認メッセージを表示する処理を追加するには、onClick イベントを選択し、次の JavaScript を入力します。

```
return confirm('Are you sure you want to submit the form?');
```

この結果、エンド・ユーザーが「OK」または「取消」をクリックできるプロンプト・ダイアログが表示され、「OK」をクリックするとフォームが送信されます。

例 A-3 値のデータ型のチェック

エンド・ユーザーによって入力された値が数値のみであることを確認するには、onChange イベントを選択し、次の JavaScript を入力します。

```
if (isNaN(this.value)) {  
  alert('Please enter a valid number.');}
```

フォームのボタンに対する PL/SQL イベント・ハンドラの記述

フォーム上のボタンに対して PL/SQL イベント・ハンドラを記述する手順は、次のとおりです。

1. イベント・ハンドラを記述するアイテムが含まれているフォームを編集します。
 - a. 目的のフォーム・コンポーネントの「管理」ページに移動します。

このページへのナビゲート方法については、A-94 ページの「[コンポーネント管理ページへのナビゲート](#)」を参照してください。
 - b. 必要な場合は、「開発」タブをクリックして前面に表示します。
 - c. タブの下部にある「編集」リンクをクリックします。
2. 「書式および妥当性チェック・オプション」タブをクリックして前面に表示します。
3. 左側のフレームで、PL/SQL イベント・ハンドラを記述するボタンの名前をクリックします。
4. 右側のフレームで、「PL/SQL ボタン・イベント・ハンドラ」セクションまでスクロールします。

5. 次のいずれかを定義します。
 - 事前定義イベント: 「PL/SQL ボタン・イベント・ハンドラ」リストからボタンのイベントを選択します。
 - カスタム・イベント: 「PL/SQL ボタン・イベント・ハンドラ」リストから「カスタム」を選択し、対応する PL/SQL コードを入力します。

PL/SQL によるフォームの値の取得および設定

変数 `p_session` に含まれているフォームのセッション記憶域オブジェクトのメソッドを使用して、フォームのフィールド値をアクセスおよび変更できます。値を取得するには、`p_session` のそれぞれの型に対応する `get` メソッドを使用する必要があります。この `p_session` には、データ型 `NUMBER`、`VARCHAR2` および `DATE` について、次の `get` ファンクションがあります。

- `get_value_as_NUMBER`
- `get_value_as_VARCHAR2`
- `get_value_as_DATE`

これらのファンクションの引数として、ブロック名と属性名を指定する必要があります。単一ブロックのフォームの場合、ブロック名は `DEFAULT` です。マスター・ディテール・フォームの場合、ブロック名は `MASTER_BLOCK` または `DETAIL_BLOCK` です。

ディテール・ブロックの場合は行索引も指定する必要があります。属性名は、列名の前に `A_` が付いています。

単一ブロックのフォームで `DEPTNO` の値を取得するには、次のコードを記述します。

```
declare
    my_deptno number;
begin
    my_deptno := p_session.get_value_as_NUMBER(
        p_block_name => 'DEFAULT',
        p_attribute_name => 'A_DEPTNO');
end;
```

たとえば、ディテール・ブロックの3番目のレコードでこれを実行するには、次のコードを使用します。

```
my_deptno := p_session.get_value_as_NUMBER(
    p_block_name => 'DETAIL_BLOCK',
    p_attribute_name => 'A_DEPTNO',
    p_index => 3);
```

フィールド値を設定するには、`p_session.set_value` を使用します。この `p_session.set_value` ファンクションは、`get_value` ファンクションと同じ引数と、値を指定するための `p_value` を使用します (`p_value` はあらゆるデータ型にオーバーロードされます)。`set_value` はプロシージャのため、値を戻しません。たとえば、`DEPTNO` の値を設定するには、次のコードを使用します。

```
p_session.set_value(  
    p_block_name => 'DEFAULT',  
    p_attribute_name => 'A_DEPTNO',  
    p_value => '20');
```

任意のボタン・イベント・ハンドラおよびフォーム・ウィザードの「**拡張 PL/SQL**」ステップの次の位置で、セッション記憶域オブジェクト (`p_session`) を使用できます (表 A-6 も参照)。

- ページの前
- ページの後
- 処理前
- 処理後

フォーム前およびフォーム後ハンドラは、セッション記憶域オブジェクトにアクセスできないことに注意してください。

`Insert`、`Update` または `Delete` イベントにマップされるボタンのカスタム・コードを定義する場合は、ボタンのデフォルト機能を実行するために、コードの適切な位置でプロシージャ `doInsert`、`doUpdate` または `doDelete` をコールする必要があります。たとえば、「挿入」ボタンのフィールド値をプログラムで設定する場合は、`p_session.set_value` の後に `doInsert` をコールして、表に値を挿入します。

PL/SQL によるフォームまたはレポートの Cookie の取得または設定

OracleAS Portal で PL/SQL を使用する場合は、PL/SQL Gateway の一部である様々なパッケージに対してアクセス権限が付与されます。Gateway は、OracleAS Portal の一部として付属しています。これらのパッケージの 1 つがパッケージ `owa_cookie` です。

このパッケージには、クライアントのブラウザとの間で HTTP Cookie を送信および取得できるデータ型、プロシージャおよびファンクションが格納されています。HTTP Cookie とは、HTTP コール間の状態を維持するために、ブラウザに送信される内部的な文字列です。クライアントのセッション全体にわたって状態を維持できます。有効期限が含まれている場合はさらに長く維持されます。

状態を維持することは、セッション全体にわたってクライアントを識別でき、トランザクションの実行が許可されることを意味します。したがって、たとえば、ショッピング・アプリケーションでは現在選択されている商品の情報を格納できます。料金サービスでは、登録情報を返信して次回接続時にクライアントによるユーザー ID の再入力を免除できます。サイトでは、ユーザーごとのプリファレンスをクライアント上に格納できます。その結果、そのサイトに再接続するたび、クライアントによってプリファレンスが提供されます。

`owa_cookie` パッケージには、Cookie の値を設定および取得するために使用できるサブプログラムおよびデータ型が格納されています。表 A-38 に、`owa_cookie` のサブプログラムおよびデータ型をいくつかリストし、各使用方法について説明します。

表 A-38 `owa_cookie` のサブプログラムおよびデータ型

サブプログラムまたはデータ型	使用目的
<code>owa_cookie.cookie</code> データ型	Cookie の名前 / 値ペアが含まれます。
<code>owa_cookie.get</code> ファンクション	指定された Cookie の値を取得します。
<code>owa_cookie.get_all</code> プロシージャ	すべての Cookie の名前 / 値ペアを取得します。
<code>owa_cookie.remove</code> プロシージャ	指定された Cookie を削除します。
<code>owa_cookie.send</code> プロシージャ	HTTP ヘッダーに Set-Cookie 行を生成します。

セッションは、Web Request Broker によって、一定期間にわたる複数回のアクセスを通してゲートウェイ内で永続的な状態を保持するために使用されます。PL/SQL Gateway は、データベースへの接続ではすべての状態がデータベース内で保持されるという特徴があるため、セッションの概念は PL/SQL Gateway には適用されません。かわりに、Cookie を使用して、クライアント・ブラウザからの永続的な状態変数を保持できます。

ページ・パラメータを介した値の定義

多くのコンポーネントは、ページ・パラメータを使用して列の値を設定できます。次に例を示します。

- コンポーネントの「カスタマイズ・フォームの表示オプション」ページで、列をカスタマイズ可能に指定します。
- コンポーネントをホスティングするページ・グループに対してパラメータを使用可能にします。
- コンポーネントをホスティングするページにページ・パラメータを追加します。
- ページ・パラメータをポートレット・パラメータにマップします。

この項では、ページ・パラメータを使用して値を設定する手順について簡単に説明します。ここでは、カスタマイズ可能な値（パラメータまたはバインド変数）をレポートに追加し、ポートレットのコンポーネントをホスティングするページにそのパラメータをマップします。

パラメータを設定してポータル・ページで使用する手順は、次のとおりです。

1. A-36 ページの「宣言によるレポートの構築」の説明に従って、レポートを作成します。
2. 「カスタマイズ・フォームの表示オプション」に関するステップで、「必須の値」チェック・ボックスを選択し、列を選択し、そのプロンプトを割り当て、列をパブリックにします（[図 A-71](#)）。

図 A-71 レポートのカスタマイズ・フォームの表示オプション

カスタマイズ・フォームの表示オプション

レポートのカスタマイズ・フォームに表示するパラメータを選択します。選択するパラメータごとに、入力フィールドがカスタマイズ・フォームに表示されます。エンド・ユーザーは、この入力フィールドを使用して、レポートにデータを表示する独自の条件を選択できます。エンド・ユーザーにパラメータ入力フィールドへの値の入力を求める場合、「必須の値」をチェックします。

必須の値	列名	プロンプト	LOV	LOVの表示	パブリックにする
<input checked="" type="checkbox"/>	EMP_DEPTNO	DeptNo	<input type="text"/>	<input type="text"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>			<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
<input type="checkbox"/>			<input type="text"/>	<input type="text"/>	<input type="checkbox"/>

パラメータ追加

3. レポートの作成を完了します。
4. ページ・グループのポートレットに移動します（通常は、Portal ビルダの「構築」タブにあります）。

5. 「ページ・グループ」フィールドで、レポートをホスティングするページ・グループを選択し、「編集」ボタンをクリックしてページ・グループのプロパティを編集します。
6. 「構成」タブをクリックして先に進み、タブの「パラメータとイベント」セクションまで下にスクロールします。
7. 「編集」リンクをクリックします。
8. 「パラメータおよびイベントを有効にする」チェック・ボックスを選択します。
9. 「OK」をクリックし、次に「閉じる」をクリックします。
10. ページ・グループのポートレットで、「ページの編集」の下の「名前」フィールドの横にある「ブラウズ」アイコンをクリックします。
11. 新しく表示されたウィンドウで、ポートレットをホスティングするページにドリルし、その横の「オブジェクトを戻す」リンクをクリックします。
12. 新しく作成された「名前」フィールドの横にある「編集」ボタンをクリックします。
13. 新しいレポート・ポートレットをページに追加します。
 - a. 特定の領域で、「ポートレットの追加」アイコンをクリックします。
 - b. ポートレット・リポジトリの「Portlet ステージング領域」ノードをクリックします。
 - c. ポートレットを所有するプロバイダの名前をクリックします。
 - d. ポートレットをクリックして、「選択したポートレット」リストに追加します。
 - e. 「OK」をクリックして、ホスト・ページに戻ります。

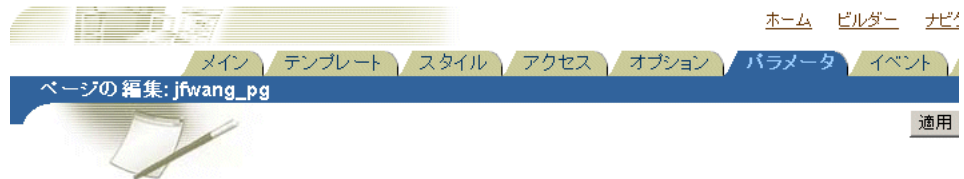
ポートレットには、「エラー: 必須フィールドが未設定です - emp.deptno (WWV-14900)」というエラー・メッセージが表示されます。ページ・パラメータを定義して、値を指定できるポートレット・パラメータにそのページ・パラメータを割り当てると、このメッセージは表示されなくなります。

14. ページ上部のページ・ツールバーで、「ページ: プロパティ」リンクをクリックします。
15. 表示されたページで、「パラメータ」タブをクリックして前面に表示します。

このタブは、ページの親ページ・グループに対してパラメータおよびイベントが使用可能な場合にのみ表示されます。

16. 「パラメータ名」フィールドに、コンポーネントに対して作成したパラメータ（バインド変数）に割り当てるページ・パラメータの名前を入力します（図 A-72）。

図 A-72 「パラメータ」タブの「パラメータ名」フィールド



新規ページ・パラメータ

パラメータの名前を入力して、「追加」をクリックしてください。

パラメータ名

17. 「追加」をクリックします。
18. 必要に応じて「ページ・パラメータのプロパティ」セクションに移動し、パラメータを構成します（図 A-73）。
- 他のユーザーがパラメータを識別するための表示名を入力します。
 - デフォルト値を入力します。
 - ユーザーがページをカスタマイズする際に、パラメータの値を変更可能にするかどうかを選択します。
 - パラメータの説明を入力します。

図 A-73 ページ・パラメータ・プロパティの定義

ページ・パラメータのプロパティ

他のユーザーがパラメータの表示名を識別できるように、表示名を入力します。必要に応じて、デフォルト値および説明を入力します。ユーザーがページをカスタマイズするときにパラメータの値を変更できるようにするかどうかを選択します。ページをカスタマイズする際、ページからパラメータを削除する場合は削除アイコン、パラメータの表示順序を変更する場合は移動アイコンをクリックします。

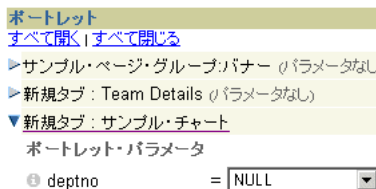
名前	表示名	デフォルト値	カスタマイズ可能	説明
✘ department_no	<input type="text" value="部署No."/>	<input type="text" value="20"/>	<input checked="" type="checkbox"/>	<input type="text"/>

19. 「ポートレット・パラメータ」セクションに移動し、手順 13 でページに追加したポートレットの横のノードを開きます (図 A-74)。

図 A-74 ポートレット・パラメータ値の下のポートレット・ノードを開く

ポートレット・パラメータ値

ポートレットを開いてパラメータを表示し、パラメータ値の設定方法を指定します。ポートレット・パラメータをページ・パラメータ、システム変数、または定数値にマップできます。

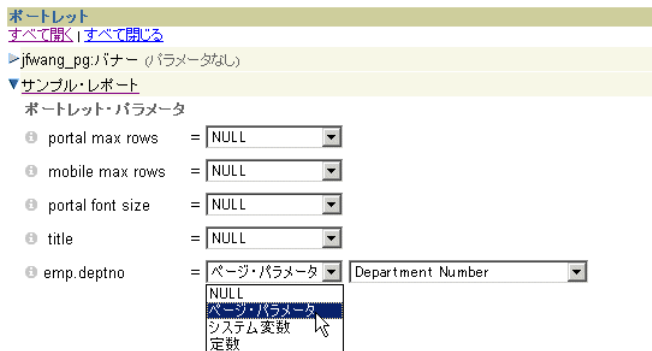


20. ポートレット・パラメータの横にあるドロップダウン・リストから、「ページ・パラメータ」を選択します (図 A-75)。

図 A-75 ポートレット・パラメータとともに使用するマッピングのタイプの選択

ポートレット・パラメータ値

ポートレットを開いてパラメータを表示し、パラメータ値の設定方法を指定します。ポートレット・パラメータをページ・パラメータ、システム変数、または定数値にマップできます。





21. デフォルトでは、横のドロップダウン・リストには関連するページ・パラメータの表示名が表示されます。表示されない場合は、関連する名前をリストから選択します (例では、関連する表示名は「**Department Number**」です)。

この結果、「(カスタマイズ・フォームの表示オプション) ページ」でポートレットに定義したパラメータが、「(パラメータ) タブ」でページに定義したパラメータにマップされます。

22. 「OK」をクリックします。
23. 手順 18b でページ・パラメータにデフォルト値を指定した場合は、レポート・ポートレットが、パラメータに関連する結果（たとえば、指定した部門番号の結果）とともに表示されます（[図 A-76](#)）。


図 A-76 部門 20 に対するレポート結果

サンプル・レポート		カスタマイズ  			
Empno	Ename Job	Manager	Hiredate	Sal Comm	Deptno
7369	SMITH CLERK		7902 80-12-17	800 (NULL)	20
7566	JONES MANAGER		7839 81-04-02	2975 (NULL)	20
7788	SCOTT ANALYST		7566 82-12-09	3000 (NULL)	20
7876	ADAMS CLERK		7788 83-01-12	1100 (NULL)	20
7902	FORD ANALYST		7566 81-12-03	3000 (NULL)	20

デフォルト値を指定しなかった場合は、ポートレット・ヘッダーの「カスタマイズ」リンクをクリックし（ただし、ステップ 18b でパラメータをカスタマイズ可能にしていた場合）、「問合せオプション」の下で値を設定することで値を指定できます（[図 A-77](#)）。

図 A-77 ページおよびポートレットのパラメータ値のカスタマイズ

パラメータのカスタマイズ



パラメータのカスタマイズ
カスタマイズするパラメータの値を指定してください

タイトル _____

表示名

問合せオプション _____

部署No. [=

共有コンポーネントによる外観の作成

各ポートレット構築ウィザードには、ポートレットのコンテンツの外観を定義できるオプションが含まれています。たとえば、レポートに表示される行の色、チャート・ラベルのフォントのタイプ、色、サイズ、およびフォームの入力フィールドの高さ、幅、順序を選択できます。外観に関する組込みの機能に加えて、OracleAS Portal には、共有コンポーネント・プロバイダによる拡張機能が用意されており、外観をさらにカスタマイズできます。

共有コンポーネント・プロバイダによって、ウィザードで構築されるフォームで使用する JavaScript の作成、カスタムの色、フォント、イメージの定義、およびページやポートレットへのユーザー・インタフェース・テンプレートの適用ができます。

共有コンポーネント・プロバイダは、Portal ナビゲータ内の「プロバイダ」タブにある「ローカルで構築されたプロバイダ」にあります。

OracleAS Portal には、複雑な設定が不要なデフォルトの共有コンポーネントのセットが含まれています。これらは、システム・タイプのコンポーネントです。ユーザー自身が共有コンポーネント・プロバイダの下に開発するコンポーネントは、ユーザー・タイプのコンポーネントです。ユーザー・タイプの共有コンポーネントに対しては編集、エクスポートおよび削除ができますが、システム・タイプの共有コンポーネントに対してはこれらの操作ができません。システム・タイプの共有コンポーネントを編集するには、そのコンポーネントをコピーし、コピーを編集します。

この項では、共有コンポーネント・プロバイダの機能を紹介し、その使用方法について説明します。この項の内容は、次のとおりです。

- [共有コンポーネントに対するアクセス権限の付与](#)
- [JavaScript によるフィールド・レベルおよびフォーム・レベルの妥当性チェック](#)
- [色定義の作成](#)
- [イメージ定義の作成](#)
- [フォント定義の作成](#)
- [ユーザー・インタフェース・テンプレートの使用方法](#)

共有コンポーネントに対するアクセス権限の付与

共有コンポーネント・プロバイダに対するアクセス権限は、共有コンポーネントに対して実行できる操作を定義します。表 A-39 に、共有コンポーネント・プロバイダに関連するアクセス権限をリストし、これらの権限について説明します。

表 A-39 共有コンポーネント・プロバイダに関連するアクセス権限

アクセス権限	実行できる操作
管理	<ul style="list-style-type: none"> ■ 他のユーザーまたはグループへの共有コンポーネント・アクセス権限の付与 ■ 新規共有コンポーネントの作成 ■ システム・タイプの共有コンポーネントをコピーして新しいユーザー・タイプの共有コンポーネントを作成 ■ ユーザー・タイプの共有コンポーネントの編集 ■ ユーザー・タイプの共有コンポーネントの削除 ■ 別のスキーマまたはデータベースへのユーザー・タイプの共有コンポーネントのエクスポート
新規作成	<ul style="list-style-type: none"> ■ 新規共有コンポーネントの作成 ■ システム・タイプの共有コンポーネントをコピーして新しいユーザー・タイプの共有コンポーネントを作成

デフォルトでは、DBA グループまたは PORTAL_DEVELOPERS グループのメンバーであるすべてのユーザーに、「管理」共有コンポーネント・アクセス権限が付与されます。

アクセス権限は、すべての共有コンポーネントについて付与されます。共有コンポーネントのタイプ（すべての JavaScript など）に対して付与したり、個々の共有コンポーネント（特定の JavaScript など）に対して付与することはできません。アクセス権限は、すべての共有コンポーネントに対して付与するか、または付与しないかのいずれかにする必要があります。

共有コンポーネント・プロバイダに対するアクセス権限をユーザーまたはグループに付与する手順は、次のとおりです。

1. ナビゲータで、「**プロバイダ**」タブをクリックして前面に表示します。
2. 「プロバイダ」タブのルート・レベルで、「**ローカルで構築されたプロバイダ**」リンクをクリックします。
3. 共有コンポーネント・プロバイダの「**操作**」列で、「**アクセス権限付与**」リンクをクリックします。

4. 「ユーザー/ロール」フィールドに、共有コンポーネントに対するアクセスを許可するユーザーまたはグループの名前を入力します。
「ユーザーをブラウズ」または「グループをブラウズ」アイコンをクリックし、表示されたリストからユーザーまたはグループを選択する方法もあります。
5. ユーザーまたはグループに付与するアクセス権限のレベルを、使用可能な権限のリストから選択します。
関連する権限のリストは、表 A-39 を参照してください。
6. 「追加」をクリックします。
指定したユーザーまたはグループが、ページ下部の「アクセス権限の変更」セクションに表示されます。
7. (オプション) アクセス権限を変更する場合は、「アクセス権限の変更」セクション内のユーザーまたはグループの横で、新しい権限を選択します。
8. (オプション) 権限を削除する場合は、「アクセス権限の変更」セクションのユーザーまたはグループの横にある「削除」アイコンをクリックします。
9. 「OK」をクリックします。

JavaScript によるフィールド・レベルおよびフォーム・レベルの妥当性チェック

OracleAS Portal には、フォームの入力フィールドに対してフィールド・レベルおよびフォーム・レベルの妥当性チェックを実行する JavaScript を作成するためのツールが用意されています。フィールド・レベルの妥当性チェックは、ユーザーが入力フィールドに値を入力した後、タブで別の入力フィールドに移動したときなど、onBlur 条件が発生した場合に実行されます。フォーム・レベルの妥当性チェックは、ユーザーが入力フィールドに値を入力して、「OK」ボタンをクリックしたときなど、ページのすべての値を送信した後に発生します。

この項では、JavaScript を使用してフォームのフィールド・レベルまたはフォーム・レベルの妥当性チェックを構築するためのガイドラインをいくつか示し、JavaScript を作成してフォームに追加する方法について説明します。この項の内容は、次のとおりです。

- フィールド・レベルまたはフォーム・レベルの妥当性チェックを実行する JavaScript の作成に関するガイドライン
- 共有コンポーネント・プロバイダの下に JavaScript を作成する方法
- JavaScript のフォームへの追加

フィールド・レベルまたはフォーム・レベルの妥当性チェックを実行する JavaScript の作成に関するガイドライン

フィールド・レベルまたはフォーム・レベルの妥当性チェックを実行する JavaScript を作成するときは、次のガイドラインに従ってください。

注意： JavaScript を作成するには、少なくとも「新規作成」共有コンポーネント・アクセス権限が必要です。

JavaScript を編集するには、「管理」共有コンポーネント・アクセス権限が必要です。

- 妥当性チェックのルーチンはすべてファンクションとして作成され、TRUE または FALSE のいずれかの値を戻す必要があります。
- ルーチンは、チェック対象の要素（入力フィールド）に無効な値が含まれている場合は、ユーザーにアラート・メッセージを表示する必要があります。
- ルーチンは、JavaScript でフラグを設定した誤った値が入力された入力フィールドに、フォーカスをあわせる（カーソルを置く）必要があります。

例 A-4 に、フィールド・レベルの妥当性チェックを実行する JavaScript の使用例を示します。

例 A-4 JavaScript の例

```
1-> function isNumber(theElement)
    {
2->   if (isNaN(Math.abs(theElement.value)))
    {
3->     alert("Value must be a number.");
4->     theElement.focus();
        return false;
    }
    return true;
  }
```

例 A-4 の JavaScript では、フィールド・レベルの妥当性チェックのタスクが次の順序で実行されます。

1. ファンクション名およびチェック対象の入力フィールドを識別します。
2. 入力フィールドの絶対値が数値であるかどうかをチェックします。値が数値でないことを通知する関連 JavaScript ファンクションは、`isNaN` です。
3. 入力フィールドの値が数値以外の場合、ユーザーは「値は数値である必要があります。」というアラート・メッセージを受信します。
4. ルーチンは、入力フィールドにフォーカスをあわせません。

共有コンポーネント・プロバイダの下に JavaScript を作成する方法

共有コンポーネント・プロバイダの下に JavaScript を作成する手順は、次のとおりです。

1. ナビゲータで、「**プロバイダ**」タブをクリックして前面に表示します。
2. 「プロバイダ」タブのルート・レベルで、「**ローカルで構築されたプロバイダ**」リンクをクリックします。
3. 「**共有コンポーネント**」プロバイダのリンクをクリックします。
4. 「**新規作成 ...**」の横にある「**Javascript**」リンクをクリックします。
5. 「**JavaScript 名**」フィールドに、JavaScript の説明的な名前を入力します。
たとえば、入力フィールドに NULL 値がないことを確認する JavaScript の場合は、NotNull と入力します。
6. 「**言語**」フィールドに、JavaScript を作成する言語を入力します。
たとえば、JavaScript1.1 または JavaScript1.2 と入力します。
7. 「**次へ**」をクリックします。
8. 表示されるフィールドに JavaScript を入力またはコピーします。
9. 「**完了**」をクリックします。

作成した JavaScript を編集するには、目的の JavaScript にドリル (Portal ナビゲータ→「プロバイダ」タブ→「ローカルで構築されたプロバイダ」リンク→「共有コンポーネント」→「JavaScript」→作成した JavaScript) し、その JavaScript の横にある「**編集**」リンクをクリックします。

JavaScript のフォームへの追加

共有コンポーネント・プロバイダの下に JavaScript を作成すると、その JavaScript は、フォーム構築ウィザードで使用可能な選択候補リストに自動的に追加されます。たとえば、フォームを作成するときに、「書式および妥当性チェック・オプション」ページが表示された場合、作成した JavaScript を選択できます。使用可能な JavaScript のリストは、このページの「**妥当性チェック・オプション**」セクションの下にあります (詳細は A-19 ページの「**宣言によるフォームの構築**」を参照)。

色定義の作成

色定義の作成では、ポータルで使用する色に、わかりやすい名前を指定できます。色定義は、色名とその 16 進値の関連付けです。たとえば、標準の赤を企業の色として使用する場合があります。色を定義する機能を使用すると、その赤を選択して、`standard_red`、`company_red` または `<your company name>_red` など、OracleAS Portal 内でわかりやすい名前を付けることができます。

#XXXXXX (X は 0 ~ 9 または A ~ F の値) という形式の色の値を、選択した名前に関連付けます。定義した色名は、OracleAS Portal ポートレットのフォント、ページの背景などの要素で使用されます。

注意： 色定義を作成するには、少なくとも「新規作成」共有コンポーネント・アクセス権限が必要です。

色定義を編集するには、少なくとも「管理」共有コンポーネント・アクセス権限が必要です。

色定義を作成する手順は、次のとおりです。

1. ナビゲータで、「**プロバイダ**」タブをクリックして前面に表示します。
2. 「プロバイダ」タブのルート・レベルで、「**ローカルで構築されたプロバイダ**」リンクをクリックします。
3. 「**共有コンポーネント**」プロバイダのリンクをクリックします。
4. 「**新規作成 ...**」の横にある「**色**」リンクをクリックします。
5. 「**色名**」フィールドに、色に指定する名前を入力します。
My_Blue_Color など、どのような名前でも指定できます。
6. 「**色の値**」フィールドに、色の 16 進値を入力します。たとえば、赤の場合は #FF0000 です。
16 進の値の前に # を付ける必要があります。パレット内の色をクリックすると、その 16 進値が「**色の値**」フィールドに自動的に入力されます。
7. (オプション) 色の値をプレビューするには、「**プレビュー**」をクリックします。
8. 色定義の内容を確認し、「**新規作成**」をクリックします。

ページが更新され、リンクが表示されます。このリンクをクリックすると色定義を編集できます。この時点で色定義を編集しない場合は、「**閉じる**」をクリックします。

イメージ定義の作成

イメージ定義の作成では、ポータルで使用するイメージに、わかりやすい名前とタイプを指定できます。イメージ定義は、イメージ名とイメージが含まれるファイル名の関連付けです。どのような名前でも指定できます。

注意： イメージ定義を作成するには、少なくとも「新規作成」共有コンポーネント・アクセス権限が必要です。

イメージ定義を編集するには、少なくとも「管理」共有コンポーネント・アクセス権限が必要です。

イメージ定義を作成する手順は、次のとおりです。

1. ナビゲータで、「**プロバイダ**」タブをクリックして前面に表示します。
2. 「**プロバイダ**」タブのルート・レベルで、「**ローカルで構築されたプロバイダ**」リンクをクリックします。
3. 「**共有コンポーネント**」プロバイダのリンクをクリックします。
4. 「**新規作成 ...**」の横にある「**イメージ**」リンクをクリックします。
5. 「**イメージ名**」フィールドに、イメージに指定する名前を入力します。
SiteLogo など、どのような名前でも指定できます。
6. 「**イメージ・ファイル名**」フィールドに、イメージが含まれているファイルの名前と拡張子を入力します。

たとえば、logo.gif と入力します。イメージは、OracleAS Portal の仮想ディレクトリである /images/ にマップされたディレクトリに格納されている必要があります。

注意： /images/ 仮想ディレクトリのパスは、Oracle HTTP Listener の plsql.conf ファイルで設定されます。

7. 「**イメージ・タイプ**」リストで、イメージ・タイプ（「アイコン 24x24」など）を選択します。
選択したタイプは、Portal ナビゲータの「タイプ」列にあるイメージの横に表示されます。
8. 「**新規作成**」をクリックします。

ページが更新され、リンクが表示されます。このリンクをクリックするとイメージ定義を編集できます。この時点でイメージ定義を編集しない場合は、「**閉じる**」をクリックします。

フォント定義の作成

フォント定義の作成では、ポータルで使用するフォントに、わかりやすい名前を指定できます。フォント定義は、フォント名とそのフォントに指定する説明的な名前の関連付けです。たとえば、すべてのパブリック・ドキュメントで使用されるフォントを指定する場合があります。フォント定義を作成する機能を使用すると、OracleAS Portal 内で、`<your company name>_font` などのカスタム名でそのフォントを識別できます。定義したフォントは、OracleAS Portal ポートレットに表示されるテキストに使用されます。

注意： フォント定義を作成するには、少なくとも「新規作成」共有コンポーネント・アクセス権限が必要です。

フォント定義を編集するには、少なくとも「管理」共有コンポーネント・アクセス権限が必要です。

フォント定義を作成する手順は、次のとおりです。

1. ナビゲータで、「**プロバイダ**」タブをクリックして前面に表示します。
2. 「プロバイダ」タブのルート・レベルで、「**ローカルで構築されたプロバイダ**」リンクをクリックします。
3. 「**共有コンポーネント**」プロバイダのリンクをクリックします。
4. 「**新規作成 ...**」の横にある「**フォント**」リンクをクリックします。
5. 「**フォント名**」フィールドに、フォントに指定する名前を入力します。

`web_banner_font` など、どのような名前でも指定できます。

6. 「**フォント値**」フィールドに、「Arial」などのフォント名を入力します。

Web ブラウザでサポートされているすべてのフォントの名前を指定できます。サポートされていないフォントを指定すると、Web ブラウザではそのデフォルト・フォントが使用されます。

「**フォント値**」フィールドに、複数のフォントをカンマで区切って入力すると、代替フォントを指定できます。たとえば、「Times New Roman, Times」のように指定します。この例では、ユーザーの Web ブラウザで Times New Roman フォントがサポートされていない場合は、かわりに Times フォントが使用されます。

7. 「**新規作成**」をクリックします。

ページが更新され、リンクが表示されます。このリンクをクリックするとフォント定義を編集できます。この時点でフォント定義を編集しない場合は、「**閉じる**」をクリックします。

ユーザー・インタフェース・テンプレートの使用方法

ユーザー・インタフェース (UI) テンプレートを使用すると、Web ページまたはポートレットにヘッダーおよびフッターを提供できます。UI テンプレートは、ページまたは OracleAS Portal ポートレットに適用できます。テンプレートを適用することによって、ページ・タイトル、タイトルの背景、ホーム・ページやヘルプ・ページへのリンク、および背景色や背景イメージを自動的に指定できます。

UI テンプレートは、多数のページの全体的な外観の標準化、または OracleAS Portal データベース・プロバイダ内のポートレット・グループの標準化に役立ちます。たとえば、あるプロバイダ用に、ヘッダーに会社のロゴ、タイトルに会社名、および共通の背景イメージを組み込んだ UI テンプレートを設計できます。プロバイダ内の各ポートレットで必ず同じ UI テンプレートを使用することで、すべて標準の外観に統一できます。

非構成済 UI テンプレートを使用すると、テンプレートを使用するページに、JavaScript を介して値を渡すこともできます。

この項では、構成済と非構成済の 2 種類のユーザー・インタフェース・テンプレートに関する情報を提供し、これらのテンプレートを使用して外観を作成する方法について簡単に説明します。また、UI テンプレートを使用できるようにページ・グループを構成する方法、および UI テンプレートをページに適用する方法についても説明します。この項の内容は、次のとおりです。

- [構成済ユーザー・インタフェース・テンプレートの構築](#)
- [非構成済ユーザー・インタフェース・テンプレートの構築](#)
- [UI テンプレートの使用を可能にするためのページ・グループの構成](#)
- [UI テンプレートのページへの適用](#)

構成済ユーザー・インタフェース・テンプレートの構築

構成済ユーザー・インタフェース・テンプレートは、ポートレットにのみ適用されます。このテンプレートはウィザードで作成されます。ウィザードで、テンプレートを使用する各ポートレットに適用するイメージ、テキストおよびレイアウトの各要素を指定します。

独自の HTML コードを使用して、事前に指定されている以外の属性でテンプレートを拡張することはできません。柔軟性をさらに高める、またはポートレットのみでなくページにも適用できるユーザー・インタフェース・テンプレートを構築するには、非構成済ユーザー・インタフェース・テンプレートを構築してください (A-135 ページの「[非構成済ユーザー・インタフェース・テンプレートの構築](#)」を参照)。

構成済ユーザー・インタフェース・テンプレートを構築する手順は、次のとおりです。

1. ナビゲータで、「**プロバイダ**」タブをクリックして前面に表示します。
2. 「プロバイダ」タブのルート・レベルで、「**ローカルで構築されたプロバイダ**」リンクをクリックします。
3. 「**共有コンポーネント**」プロバイダのリンクをクリックします。

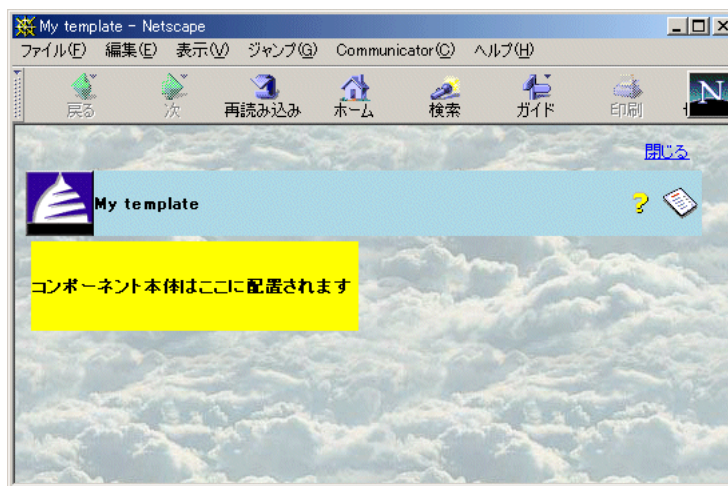
4. 「新規作成 ...」の横にある「ユーザー・インタフェース・テンプレート」リンクをクリックします。
5. 「構成済み UI テンプレート」リンクをクリックします。
6. 「テンプレート名」フィールドに、テンプレートの名前を入力します。

テンプレート名は、できるだけわかりやすい名前にしてください。この名前は、構築プロセス時に、ユーザーが UI テンプレートをポートレットに適用するときに表示されません。たとえば、そのテンプレートがスケジューリング・プロバイダ用に作成されるすべてのポートレットに適用される場合は、Schedule_Template のような名前を設定します。

7. その他のオプションを選択して、テンプレートの外観を整えます。

たとえば、[図 A-78](#) に示すように、テンプレートの左上隅に表示されるイメージ、およびポートレットの後ろの背景イメージを選択できます。

図 A-78 雲のイメージを選択した構成済み UI テンプレート



オプションに関して不明な点がある場合は、ヘルプ・アイコンをクリックしてください。テンプレートに組み込まないオプションは空白のままにします。

8. (オプション)「プレビュー」をクリックします。UI テンプレートを表示する新しいブラウザ・ウィンドウが開きます。

オプションを再選択して「プレビュー」をもう一度クリックすると、オプションの変更によってテンプレートの外観がどのように変わるかを確認できます。

9. テンプレートの内容を確認し、「**新規作成**」をクリックします。

ページが更新され、リンクが表示されます。このリンクをクリックするとテンプレートを編集できます。この時点でテンプレートを編集しない場合は、「**閉じる**」をクリックします。

非構成済ユーザー・インタフェース・テンプレートの構築

非構成済ユーザー・インタフェース・テンプレートは、提供した HTML コードに基づいて作成されます。独自の HTML コードを記述するため、構成済 UI テンプレートより複雑で洗練された非構成済 UI テンプレートを作成できます。

非構成済ユーザー・インタフェース・テンプレートを作成するには、最初に HTML コードを記述して Web ページを作成します。このコードは、Web ページ・エディタなどの別のソースから OracleAS Portal にコピーすることもできます。OracleAS Portal に入力した後、その HTML コードを編集して代替タグを追加します。HTML コードが実行されると、代替タグによって、ポートレット、タイトルなどの要素が Web ページに埋め込まれます。たとえば、チャートまたはレポートなどのポートレットを追加する `#BODY#` タグを、元の Web ページの背景に追加できます。HTML コードに組み込むことができるすべての代替タグのリストは、[表 A-40](#) を参照してください。

非構成済テンプレートでは、`<ORACLE></ORACLE>` タグを使用して、SQL 文または PL/SQL ブロックを組み込むことができます。また、カスタム JavaScript、カスケード・スタイル・シート参照および META タグなどの HEAD 要素も組み込むことができます。ページ・メタデータを統合するための特別な代替タグを使用すると、PL/SQL スクリプトを埋め込むことができます。

非構成済 UI テンプレートは、ページにも適用できます。ユーザー・インタフェース・テンプレートでは、ページ・コンテンツの**周囲**に表示される装飾が制御されます。**実際の**ページ・コンテンツの外観を制御するには、ページ・テンプレートを使用する必要があります。UI テンプレートおよびページ・テンプレートの使用方法の詳細は、『Oracle9iAS Portal ユーザーズ・ガイド』を参照してください。

非構成済ユーザー・インタフェース・テンプレートを構築する手順は、次のとおりです。

1. ナビゲータで、「**プロバイダ**」タブをクリックして前面に表示します。
2. 「プロバイダ」タブのルート・レベルで、「**ローカルで構築されたプロバイダ**」リンクをクリックします。
3. 「共有コンポーネント」プロバイダのリンクをクリックします。
4. 「**新規作成...**」の横にある「**ユーザー・インタフェース・テンプレート**」リンクをクリックします。
5. 「**非構成済み UI テンプレート**」リンクをクリックします。

6. 「**テンプレート名**」フィールドに、テンプレートの名前を入力します。

この名前は、構築プロセス時に、ユーザーが UI テンプレートをページまたはポートレットに適用するときに表示されるため、できるだけわかりやすい名前にしてください。たとえば、そのテンプレートがカレンダー用に作成されるすべてのポートレットに適用される場合は、`Calendar_Template` のような名前を設定します。

7. 「**テンプレート定義**」フィールドに、非構成済ユーザー・インタフェース・テンプレートのベースとして使用する HTML コードを入力するかまたは貼り付けます。

提供した HTML コードで Web ページが作成される必要があります。

8. HTML コード内に代替タグを埋め込みます。タグを埋め込む場所は、完成したテンプレート内で、タグと関連付けられたアイテムを表示させる場所です。

表 A-40 に、使用できるタグをリストし、これらのタグについて説明します。たとえば、コードに `#BODY#` 代替タグを埋め込むとします。HTML コードが実行されると、`#BODY#` によって、ページ・コンテンツまたは OracleAS Portal ポートレットなど、ページの本体が Web ページに追加されます。HTML ソース・コードでページが 2 つのフレームに分割されている場合は、コード内の別々の場所に `#BODY#` タグを埋め込むことができ、ポートレットを左側のフレームまたは右側のフレームに表示できます。

表 A-40 非構成済 UI テンプレートの代替タグ

タグ	設定される値
<code>#BODY#</code>	ポートレット自体またはページのポートレット。
<code>#IMAGE_PREFIX#</code>	<code>plssql.conf</code> 構成ファイルで指定された OracleAS Portal イメージ・ディレクトリ。
<code>#USER#</code>	現在ログインしているユーザーのユーザー名。
<code>#USER.FULLNAME#</code>	現在ログインしているユーザーのフル・ネーム。フル・ネームには、ユーザーの名、ミドル・ネームおよび姓が含まれます。
<code>#VERSION#</code>	現在使用している OracleAS Portal のバージョン。
<code>#HELPSCRIPT#</code>	オンライン・ヘルプを表示するウィンドウを開くために使用される JavaScript ファンクション。テンプレートに <code>#HELPLINK#</code> を含める場合は、このタグも指定する必要があります。
<code>#DIRECTION#</code>	文字レイアウトの方向（左から右または右から左）。この値は言語に基づいて設定されます。
<code>#ALIGN_LEFT#</code>	テキストを左揃えにします。
<code>#ALIGN_RIGHT#</code>	テキストを右揃えにします。

表 A-40 非構成済 UI テンプレートの代替タグ (続き)

タグ	設定される値
#PAGE.STYLE#	ページのカスケード・スタイル・シートを参照する HTML LINK 要素。 このタグは、ドキュメントの HEAD でのみ使用できます。
#PAGE.STYLE.URL#	ページのカスケード・スタイル・シートの URL。
#PAGE.BASE#	ドキュメントのベース URL の HTML 基本要素。 このタグは、ドキュメントの HEAD でのみ使用できます。
#PAGE.BASE.URL#	ベース URL。
#PAGE.BGIMAGE#	ページ全体の背景イメージの HTML ソース。
#PAGE.BGCOLOR#	ページ全体の背景色の HTML ソース。
#PAGE.SUBPAGELINK#	サブページ・リンクの URL。
#PORTAL.HOME#	ポータル・ホーム・ページへの HTML イメージ・ハイパーリンク。
#PORTAL.HOME.URL#	ポータル・ホーム・ページの URL。
#PORTAL.HOME.IMAGE#	ポータル・ホーム・ページ・リンクに使用されるイメージ。
#PORTAL.HOME.LABEL#	ポータル・ホーム・ページ・リンクに使用されるテキスト。
#PORTAL.NAVIGATOR#	ナビゲータへの HTML イメージ・ハイパーリンク。
#PORTAL.NAVIGATOR.URL#	ナビゲータの URL。
#PORTAL.NAVIGATOR.IMAGE#	ナビゲータ・リンクに使用されるイメージ。
#PORTAL.NAVIGATOR.LABEL#	ナビゲータ・リンクに使用されるテキスト。
#PORTAL.HELP#	オンライン・ヘルプへの HTML イメージ・ハイパーリンク。
#PORTAL.HELP.URL#	オンライン・ヘルプの URL。
#PORTAL.HELP.IMAGE#	オンライン・ヘルプ・リンクに使用されるイメージ。
#PORTAL.HELP.LABEL#	オンライン・ヘルプ・リンクに使用されるテキスト。
#PORTAL.LOGOUT#	ログアウト URL への HTML テキスト・ハイパーリンク。
#PORTAL.LOGOUT.URL#	ログアウト URL。
#PORTAL.LOGOUT.LABEL#	ログアウト・リンクに使用されるテキスト。

表 A-40 非構成済 UI テンプレートの代替タグ (続き)

タグ	設定される値
#PORTAL.ACCOUNTINFO#	アカウント情報ダイアログへの HTML テキスト・ハイパーリンク。
#PORTAL.ACCOUNTINFO.URL#	アカウント情報ページの URL。
#PORTAL.ACCOUNTINFO.LABEL#	アカウント情報リンクに使用されるテキスト。
#PORTAL.COMMUNITY#	OracleAS Portal コミュニティの Web サイトの名前。
#PORTAL.COMMUNITY.URL#	OracleAS Portal コミュニティの Web サイトの URL。
#PORTAL.COMMUNITY.IMAGE#	OracleAS Portal コミュニティの Web サイトのイメージ。
#PORTAL.COMMUNITY.LABEL#	OracleAS Portal コミュニティの Web サイトのラベル。
#PAGE.CUSTOMIZEPAGE#	カスタマイズ・ページへの HTML テキスト・ハイパーリンク。
#PAGE.CUSTOMIZEPAGE.URL#	カスタマイズ・ページの URL。
#PAGE.CUSTOMIZEPAGE.LABEL#	カスタマイズ・ページ・リンクに使用されるテキスト。
#PAGE.EDITPAGE#	ページの編集を可能にする HTML テキスト。
#PAGE.EDITPAGE.URL#	ページの URL の編集を可能にする HTML テキスト。
#PAGE.EDITPAGE.LABEL#	ページ・ラベルの編集を可能にする HTML テキスト。
#PAGE.REFRESH#	ページの更新に使用される HTML テキスト・ハイパーリンク。
#PAGE.REFRESH.URL#	更新 URL。
#PAGE.REFRESH.LABEL#	更新リンクに使用されるテキスト。

9. (オプション) 「**プレビュー**」をクリックします。UI テンプレートを表示する新しいブラウザ・ウィンドウが開きます。

コードを更新して「**プレビュー**」をもう一度クリックすると、コードの変更によってテンプレートの外観がどのように変わるかを確認できます。

10. テンプレートの内容を確認し、「**新規作成**」をクリックします。

ページが更新され、リンクが表示されます。このリンクをクリックするとテンプレートを編集できます。この時点でテンプレートを編集しない場合は、「**閉じる**」をクリックします。

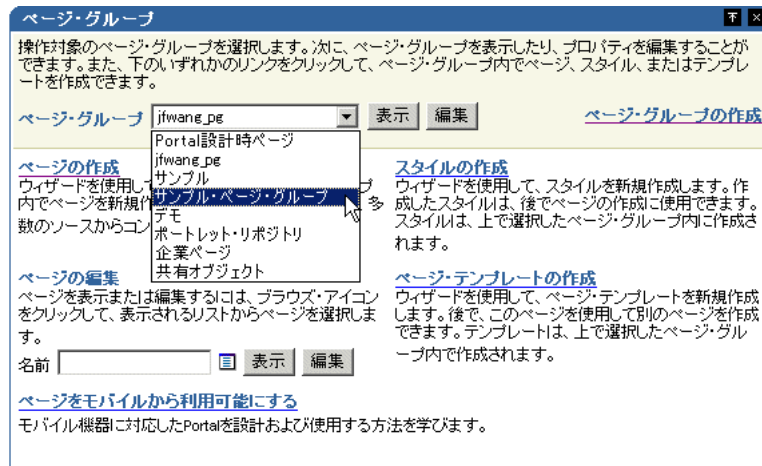
UI テンプレートの使用を可能にするためのページ・グループの構成

UI テンプレートをページに適用するには、テンプレートを使用できるようにホスト・ページ・グループを構成する必要があります。UI テンプレートを使用できるようにページ・グループを構成する手順は、次のとおりです。

1. Portal ビルダー内の「ページ・グループ」ポートレットに移動し、「ページ・グループ」ドロップダウン・リストから目的のページ・グループを選択します (図 A-79)。

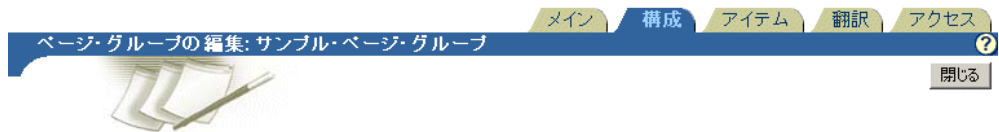
標準のインストールでは、「ページ・グループ」ポートレットは、Portal ビルダーの「構築」タブにあります。

図 A-79 「ページ・グループ」ポートレットでのページ・グループの選択



2. 選択したページ・グループの横にある「編集」ボタンをクリックします。
3. 「ページ・グループ・プロパティ」ページで、「構成」タブをクリックして前面に表示します。
4. 「構成」タブで、「ページ・タイプとテンプレート」の下にある「編集」リンクをクリックします (図 A-80)。

図 A-80 「ページ・タイプとテンプレート」の下にある「編集」リンク



ページ・タイプとテンプレート

「編集」リンクをクリックして、ユーザーがこのページ・グループ内に作成できるページのタイプを選択し、このページ・グループ内のページでデフォルトで使用するページ・テンプレートおよびナビゲーション・ページを選択します。また、ユーザー・インタフェース・テンプレートをページで使用するかどうかを選択できます。

現在の選択 **編集**

ページ・タイプ:	標準, URL
デフォルト・スタイル:	<なし>
デフォルト・ナビゲーション・ページ:	サンプル・ページ・グループ・バナー
デフォルト・ページ・テンプレート:	<なし>
UIテンプレート:	不許可

- 表示された「ページのデフォルト」ページで、「ユーザー・インタフェース・テンプレート」セクションまでスクロール・ダウンし、「ページで UI テンプレートを使用可能にする」チェック・ボックスを選択します (図 A-81)。

図 A-81 「ページのデフォルト」ページの「ユーザー・インタフェース・テンプレート」セクション

ユーザー・インタフェース・テンプレート

このページ・グループ内のページで、ユーザー・インタフェース・テンプレートを使用可能にするかどうかを選択します。UIテンプレートには、ページの周囲に表示する要素を指定します。ページ・テンプレートには、ページ自身の外観およびコンテンツを指定します。

ページでUIテンプレートを使用可能にする

- 「OK」をクリックして変更内容を保存し、「ページのデフォルト」ページを終了します。
- 「閉じる」をクリックして「ページ・グループ・プロパティ」を終了します。

UI テンプレートの作成方法の詳細は、A-133 ページの「ユーザー・インタフェース・テンプレートの使用方法」を参照してください。UI テンプレートをページに適用する方法の詳細は、A-141 ページの「UI テンプレートのページへの適用」を参照してください。

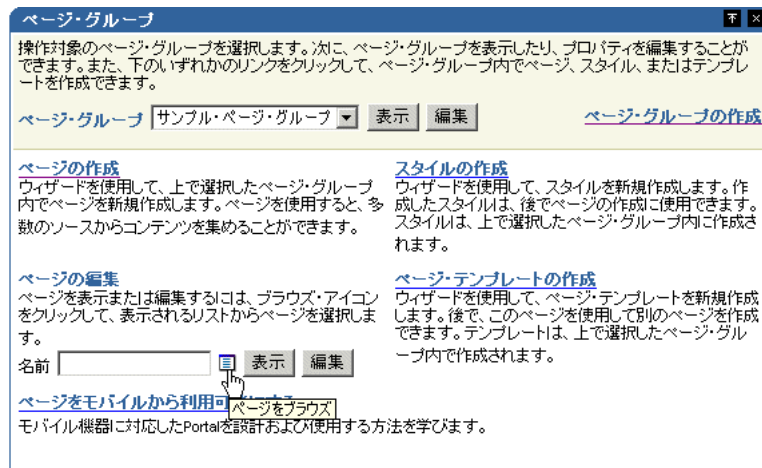
UI テンプレートのページへの適用

UI テンプレートを使用できるようにページ・グループが構成されると (A-139 ページ「[UI テンプレートの使用を可能にするためのページ・グループの構成](#)」)、作成した UI テンプレートを、そのページ・グループ内のページに適用できます (A-133 ページ「[ユーザー・インタフェース・テンプレートの使用方法](#)」)。UI テンプレートを適用する手順は、次のとおりです。

1. 「ページ・グループ」ポートレットに移動し、「ページの編集」の下の「名前」フィールドの横にある「ページをブラウズ」アイコンをクリックします (図 A-82)。

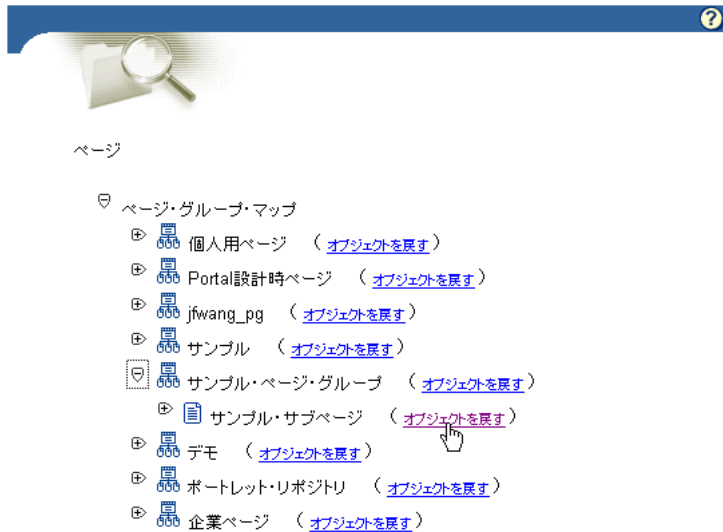
標準のインストールでは、「ページ・グループ」ポートレットは、Portal ビルダークの「構築」タブにあります。

図 A-82 「ページの編集」の下にある「ページをブラウズ」アイコン



2. UI テンプレートを適用するページまでドリルダウンします。
3. 目的のページの横にある「オブジェクトを戻す」リンクをクリックします (図 A-83)。

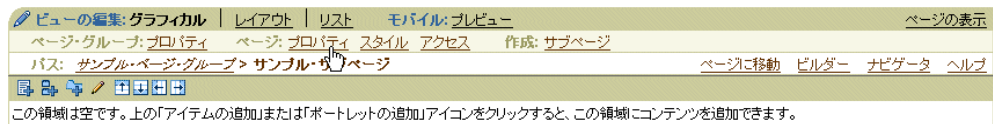
図 A-83 ページの横にある「オブジェクトを戻す」リンク



4. 「ページ・グループ」ポートレットで、「名前」フィールドの横にある「編集」フィールドをクリックします。
5. 表示されたページで、ページ上部のツールバーにある「ページ:プロパティ」リンクをクリックします (図 A-84)。

「ページ・グループ:プロパティ」リンクではなく、必ず「ページ:プロパティ」リンクをクリックしてください。

図 A-84 ページのツールバーの「ページ:プロパティ」リンク



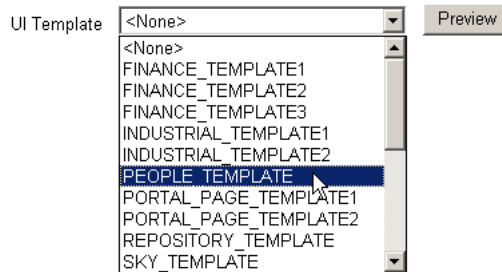
6. 「ページのプロパティ」ページで、「オプション」タブをクリックして前面に表示します。

- 「オプション」タブの「ユーザー・インターフェース・テンプレート」セクションに移動し、「UI テンプレート」ドロップダウン・リストから UI テンプレートを選択します (図 A-85)。

図 A-85 「オプション」タブでの UI テンプレートの選択

User Interface Templates

Choose the user interface template to use for the page. User interface templates govern what is displayed around the page, whereas page templates govern the appearance and contents of the page itself.



- 「OK」をクリックして変更内容を保存し、ページに戻ります。

例：チャートおよびレポートの構築

この例では、SCOTT スキーマ内の myCompany_DB_Provider という OracleAS Portal データベース・プロバイダに対するアクセス権限があると仮定しています。適切な権限がある場合、このプロバイダは自分で作成できます（作成方法は、A-10 ページの「[ローカルで構築されたポートレットに対するプロバイダの作成](#)」を参照）。OracleAS Portal データベース・プロバイダを作成するための適切な権限がない場合は、使用するプロバイダの作成をポータル管理者に依頼してください。

この例の内容は、次のとおりです。

- [演習：チーム詳細レポートの構築](#)
- [演習：平均給与チャートの構築](#)
- [演習：チーム賞与レポートの構築](#)

演習：チーム詳細レポートの構築

チーム詳細レポート（[図 A-86](#)）には、営業部門（部門 30）の従業員のリストが表示されます。

図 A-86 チーム詳細レポート

Name	Employee No	Job
ALLEN	7499	SALESMAN
WARD	7521	SALESMAN
MARTIN	7654	SALESMAN
BLAKE	7698	MANAGER
TURNER	7844	SALESMAN
JAMES	7900	CLERK

チーム詳細レポートを構築する手順は、次のとおりです。

1. ナビゲータで、「[プロバイダ](#)」タブをクリックして前面に表示します。
2. 「[プロバイダ](#)」タブのルート・レベルで、「[ローカルで構築されたプロバイダ](#)」リンクをクリックします。
3. プロバイダ「[myCompany_DB_Provider](#)」のリンクをクリックします。
このプロバイダが表示されない場合は、作成できます。詳細は、A-10 ページの「[ローカルで構築されたポートレットに対するプロバイダの作成](#)」を参照してください。
4. 「[新規作成 ...](#)」の横にある「[レポート](#)」リンクをクリックします。

5. 「問合せウィザードからのレポート」リンクをクリックします。
6. 「名前」フィールドに、<YourName>_team_details と入力します。
7. 「表示名」フィールドに、Team Details と入力します。
8. 「説明」フィールドに、team details exercise と入力します。
9. 「Portal DB プロバイダ」リストで、「MYCOMPANY_DB_PROVIDER」が選択されていることを確認します。
10. 「次へ」をクリックします。
11. 「表およびビュー」フィールドに、SCOTT.EMP と入力します（必要な場合）。
12. 「追加」をクリックします。
13. 「次へ」をクリックします。
14. 「列」リストから、次の列を選択します。
 - EMP.ENAME
 - EMP.EMPNO
 - EMP.JOB各列の選択後、右矢印ボタンをクリックして「列」から「選択列」に移動します。
15. 必要に応じて、「選択列」の右側にある上矢印と下矢印を使用して、前の手順で指定されている順に列を並べます。
16. 「次へ」をクリックします。
17. 「列名」リストから、「EMP.DEPTNO」を選択します。
18. 「条件」リストから、「=」を選択します。
19. 「値」フィールドに、30 と入力します。
20. 「次へ」をクリックします。
21. 「表」を選択します。
22. 「次へ」をクリックします。
23. 各列の横にある「列ヘッダー・テキスト」に、表 A-41 に示されているテキストを入力します。

表 A-41 チーム詳細レポートの列ヘッダー・テキスト

列	列ヘッダー・テキスト
ENAME	Name
EMPNO	Employee No
JOB	Job

この結果、レポートに表示される列の上に説明ラベルが追加されます。

24. 「次へ」を2回クリックするか、ウィザードの「**表示オプション**」ステップが表示されるまでクリックします。
25. 「**フル・ページ・オプション**」セクションと「**ポートレット・オプション**」セクションの両方で、表 A-42 に示されている値を選択します。

表 A-42 チーム詳細レポートのフル・ページ・オプションとポートレット表示オプション

オプション	値
ヘッダーのフォント	Arial
ヘッダーのフォント色	白
ヘッダーのサイズ	10pt
行テキストのフォント	Arial
行テキストのフォント色	黒
行テキストのサイズ	10pt
ヘッダーの背景色	スレート・グレー
境界線	細い罫線

26. 「完了」をクリックします。
27. 「**ポートレットとして実行**」をクリックして、レポートの表示内容を確認します。
28. レポートが表示されているブラウザ・ウィンドウを閉じます。「**閉じる**」をクリックしてコンポーネント・マネージャを閉じて、Portal ナビゲータに戻ります。

演習：平均給与とチャートの構築

平均給与チャート (図 A-87) には、各役職名の平均給与が表示されます。

図 A-87 平均給与とチャート

ANALYST		3,000
CLERK		1,187
MANAGER		3,133
PRESIDENT		7,295
SALESMAN		1,400

平均給与チャートを構築する手順は、次のとおりです。

1. ナビゲータで、「**プロバイダ**」タブをクリックして前面に表示します。
2. 「プロバイダ」タブのルート・レベルで、「**ローカルで構築されたプロバイダ**」リンクをクリックします。
3. プロバイダ「**myCompany_DB_Provider**」のリンクをクリックします。
4. 「新規作成...」の横にある「**チャート**」リンクをクリックします。
5. 「**SQL 問合せからのチャート**」リンクをクリックします。
6. 「名前」フィールドに、<YourName>_team_average_salary と入力します。
7. 「表示名」フィールドに、Average Salaries と入力します。
8. 「説明」フィールドに、average salaries exercise と入力します。
9. 「Portal DB プロバイダ」リストで、「**MYCOMPANY_DB_PROVIDER**」が選択されていることを確認します。
10. 「次へ」をクリックします。
11. 「SQL 問合せ」フィールドに、次のコードを入力します。

```
select
    null    the_link,
    job     the_name,
    avg(sal) the_data
from emp
group by job
```

この SQL 問合せは、SCOTT スキーマ内の EMP 表に対して SELECT 権限がある場合のみ使用できます。

12. 「次へ」を2回クリックするか、ウィザードの「**表示オプション**」ステップが表示されるまでクリックします。
13. 「**フル・ページ・オプション**」セクションと「**ポートレット・オプション**」セクションの両方で、表 A-43 に示されている値を選択します。

表 A-43 平均給与チャートのフル・ページ・オプションとポートレット表示オプション

オプション	値
フォント	Arial
フォント色	黒
フォント・サイズ	10pt
チャート・タイプ	水平
棒グラフ	赤のバー (red.gif)

14. 「完了」をクリックします。
15. 「ポートレットとして実行」をクリックして、チャートの表示内容を確認します。
16. チャートが表示されているブラウザ・ウィンドウを閉じます。「閉じる」をクリックしてコンポーネント・マネージャを閉じて、Portal ナビゲータに戻ります。

演習：チーム賞与レポートの構築

チーム賞与レポート (図 A-88) には、各部門の従業員に支払われた平均賞与が表示されません。

図 A-88 チーム賞与レポート

Department No	Department	Bonus Paid
10	ACCOUNTING,NEW YORK	(null)
20	RESEARCH,DALLAS	(null)
30	SALES,CHICAGO	2210

チーム賞与レポートを構築する手順は、次のとおりです。

1. ナビゲータで、「**プロバイダ**」タブをクリックして前面に表示します。
2. 「プロバイダ」タブのルート・レベルで、「**ローカルで構築されたプロバイダ**」リンクをクリックします。
3. 「**myCompany_DB_Provider**」プロバイダのリンクをクリックします。

4. 「新規作成 ...」の横にある「レポート」リンクをクリックします。
5. 「SQL 問合せからのレポート」リンクをクリックします。
6. 「名前」フィールドに、<YourName>_team_bonuses と入力します。
7. 「表示名」フィールドに、Team Bonuses と入力します。
8. 「説明」フィールドに、team bonuses example と入力します。
9. 「Portal DB プロバイダ」リストで、「MYCOMPANY_DB_PROVIDER」が選択されていることを確認します。
10. 「次へ」をクリックします。
11. 「SQL 問合せ」フィールドに、次のコードを入力します。

```
select dept.deptno, dept.dname||','||dept.loc, sum(emp.comm)
from dept, emp
where dept.deptno = emp.deptno
group by dept.deptno, dept.dname||','||dept.loc
```

この問合せでは、DEPT 表および EMP 表内の選択された列のデータが集計されます。ここではレポートの構築に SQL 問合せを使用していますが、レポート問合せウィザードを使用した場合も同様に簡単に構築できます。

この SQL 問合せは、SCOTT スキーマ内の DEPT 表および EMP 表に対して SELECT 権限がある場合のみ使用できます。

12. 「次へ」をクリックします。
13. 「表」を選択します。
14. 「次へ」をクリックします。
15. 各列の横にある「列ヘッダー・テキスト」に、表 A-44 に示されているテキストを入力します。

表 A-44 チーム賞与レポートの列ヘッダー・テキスト

列	列ヘッダー・テキスト
DEPTNO	Department No
DEPT.DNAME ',' DEPT.LOC	Department
SUM(EMP.COMM)	Bonus Paid

この結果、レポートに表示される列の上に説明ラベルが追加されます。

16. 「次へ」を2回クリックするか、ウィザードの「表示オプション」ステップが表示されるまでクリックします。

17. 「フル・ページ・オプション」セクションと「ポートレット・オプション」セクションの両方で、表 A-45 に示されている値を選択します。

表 A-45 チーム賞与レポートのフル・ページ・オプションとポートレット表示オプション

オプション	値
ヘッダーのフォント	Arial
ヘッダーのフォント色	白
ヘッダーのサイズ	10pt
行テキストのフォント	Arial
行テキストのフォント色	黒
行テキストのサイズ	10pt
ヘッダーの背景色	スレート・グレー
境界線	細い罫線

18. 「完了」をクリックします。
19. 「ポートレットとして実行」をクリックして、レポートの表示内容を確認します。
20. レポートが表示されているブラウザ・ウィンドウを閉じます。「閉じる」をクリックしてコンポーネント・マネージャを閉じて、Portal ナビゲータに戻ります。

用語集

ACL

アクセス制御リスト。オブジェクトに対する特定のアクセスを許可されたグループおよびユーザーのリスト。

AJP

Apache JServ Protocol。Oracle HTTP Server などの Apache 対応の HTTP サーバーと、Jserv、Tomcat または OC4J などの J2EE サーバー間との通信のためのオープン標準プロトコル。

Apache JServ Protocol

「AJP」を参照。

API

Application Program Interface。アプリケーション側でポートレット、ページまたはページ・グループ上のサービスを起動するときに使用する一連の公開済のデータ構造と関数。OracleAS Portal の API は、Portal Center (<http://portalcenter.oracle.com/>) で入手可能な PDK を使用して公開される。

Application Programming Interface

「API」を参照。

ASP

アプリケーション・サービス・プロバイダ (Application Service Provider)。アプリケーションのリモート・ホスティングであり、アプリケーションの実行に必要なハードウェア、ソフトウェアおよびその他のリソースを管理および運用する。そのよい例が Oracle Portal Online (<http://portal.oracle.com>) であり、これは、ポータルの構築を希望していても、その構築と管理のためのリソースが十分でない比較的小規模の企業や組織に対して、OracleAS Portal の機能を提供するホスティング・サブスクリプション・サービスである。

CHTML

コンパクト HTML (Compact HTML)。小型デバイス用に設計された、[HTML](#) に関する推奨事項のサブセット。

CSS

カスケード・スタイル・シート (Cascading Style Sheet)。

DAD

データベース・アクセス記述子 (Database Access Descriptor)。HTTP リクエストを実行するためにアプリケーションが Oracle データベースに接続する方法を指定する一連の値。DAD に含まれる情報には、ユーザー名 (スキーマおよび権限も指定する)、パスワード、接続文字列、エラー・ログ・ファイル、標準エラー・メッセージ、NLS パラメータ (NLS 言語、NLS 日付書式、NLS 日付言語、NLS 通貨など) がある。

DAV

「[WebDAV](#)」を参照。

DBA

データベース管理者 (Database Administrator)。DBA [グループ](#) に所属するユーザー。デフォルトでは、DBA グループのメンバーはすべての OracleAS Portal 製品ページにアクセスでき、すべての [ページ](#)、[ページ・グループ](#)、[データベース・プロバイダ](#) および管理に対する「管理」権限を持っている。

Developer Services

「[Portal Developer Services](#)」を参照。

DIP

Directory Integration Platform。様々なディレクトリとディレクトリ対応アプリケーションを同期化するために、[OID](#) によって提供されるプロビジョニング・プラットフォーム。

Directory Integration Platform

「[DIP](#)」を参照。

DIT

ディレクトリ情報ツリー (Directory Information Tree)。エントリの [DN](#) で構成される Oracle Internet Directory ([OID](#)) 内のツリー形式の階層構造。

DMZ

非武装地帯 (De-militarized Zone)。企業のプライベート・ネットワークと外部のパブリック・ネットワークとの間に中立地帯として挿入されたコンピュータ・ホストまたは小規模ネットワーク。これによって、企業データが格納されているサーバーに外部のユーザーが直接アクセスするのを防ぐ。DMZはオプションで、**ファイアウォール**に対する安全な手法であり、**プロキシ・サーバー**としても効果的に動作する (この用語は、1950年代初頭に、国連警察軍によって北朝鮮と韓国との間に設定された地理的な緩衝地帯に由来)。

DN

識別名 (Distinguished Name)。Oracle Internet Directory (**OID**) 内のディレクトリ・エントリの一意名。親エントリの個々の名前がすべて、下からルートに向かって順に結合された形で構成されている。したがって、識別名によってディレクトリの階層内のエントリの位置が正確にわかる。この階層は、ディレクトリ情報ツリー (**DIT**) によって表現される。

DR

障害回復 (Disaster Recovery)。

Edge Side Includes

「**ESI**」を参照。

EM

Oracle Enterprise Manager。Oracle Application Server のコンポーネント。管理者が単一の環境を介して Oracle Application Server を管理できるようにする。たとえば、管理者は EM を使用して、**HTTP** サービス、**mod_plsql** サービス、**Oracle Application Server Single Sign-On Server**、**PPE**、Oracle データベース、**プロバイダ**など、OracleAS Portal インスタンスを構成するサービスを監視できる。

ESI

Edge Side Includes。HTML フラグメントによりページの部分的なキャッシュ (**PPC**) を可能にするマークアップ言語。

eXtensible Markup Language

「**XML**」を参照。

FPA

連携された Portal アダプタ (Federated Portal Adapter)。Java および PL/SQL で書かれたポータル・インスタンス内のモジュールであり、**Web プロバイダ**宛の **SOAP** メッセージを受信し、その SOAP を解析し、メッセージを PL/SQL プロシージャ・コールとして**データベース・プロバイダ**に送信する。つまり、連携された Portal アダプタによってデータベース・プロバイダが Web プロバイダとまったく同じように動作するようになり、ユーザーは複数のデータベース・サーバーにデータベース・プロバイダを配布できるようになる。すべてのリモート・プロバイダが Web プロバイダとして扱われ、ユーザーからはその実装 (データベースまたは Web) はわからない。最も一般的な使用法は、1つのポータル・イン

スタンスが所有するデータベース・プロバイダ（ページ・グループも含む）を他のポータル・インスタンス間で共有することである。

HA

高可用性（High Availability）。

Handheld Device Markup Language

「[HDML](#)」を参照。

HDML

Handheld Device Markup Language。ディスプレイの小さい携帯用デバイス向けに、ハイパーテキストに似たマークアップ・コンテンツおよびアプリケーションを定義するための単純な言語。

HTML

Hyper Text Markup Language。ハイパーテキスト・ドキュメントをエンコードするための書式。テキスト、グラフィックス、プログラムへの参照および別のハイパーテキスト・ドキュメントを含めることができる。

HTTP

Hyper Text Transfer Protocol。メッセージの書式設定と送信を行い、各種のコマンドに応じて **Web サーバー** およびブラウザが実行する必要のあるアクションを決定するために、Web で使用される基本的な方式（プロトコル）。HTTP は、通常は **Oracle Application Server** とクライアントとの間で使用されるプロトコルである。

Hyper Text Markup Language

「[HTML](#)」を参照。

Hyper Text Transfer Protocol

「[HTTP](#)」を参照。

IDE

統合開発環境（Integrated Development Environment）。エディタ、デバッガ、スクリーン・ペインタ、オブジェクト・ブラウザなどを含むビジュアル・ツール。

ILS

アイテム・レベルのセキュリティ（Item Level Security）。特定の**ページ**上の**アイテム**へのアクセスを詳細に制御するメカニズム。ILS によって、アイテム管理者は、ユーザーおよび**グループ**に、ページ・レベルの権限に優先する明示的なアイテムのアクセス権限を付与できるようになる。

J2EE

Java 2 Platform Enterprise Edition。このプラットフォームを使用すると、アプリケーション開発者は複数層からなるサーバー中心の企業レベル・アプリケーションを開発、デプロイおよび管理できる。J2EE プラットフォームは、複数層からなる分散アプリケーション・モデル、統合された XML ベースのデータ交換、統合セキュリティ・モデル、および柔軟なトランザクション制御を提供する。独自の J2EE **ポートレット**を作成して、それを **Web プロバイダ**を通じて公開できる。

「**OC4J**」も参照。

J2SE

Java 2 Platform Standard Edition。このプラットフォームを使用すると、アプリケーション開発者は Java のアプレットおよびアプリケーションを、パーソナル・コンピュータやワークステーションなどのデスクトップ・クライアント・プラットフォーム上で開発、デプロイおよび管理できる。J2SE は **API** 標準を定義するのみでなく、企業アプリケーションのデプロイも指定するため、アプリケーション・サーバーの管理者は、J2SE サーバーのベンダーには関係なく、アプリケーションをデプロイできる。

「**OC4J**」も参照。

Java 2 Platform Enterprise Edition

「**J2EE**」を参照。

Java 2 Platform Standard Edition

「**J2SE**」を参照。

JavaScript

Netscape 社が開発したスクリプト言語。静的な **HTML** とは異なる動的な動作を取り入れた **ポートレット**を生成できる。OracleAS Portal では、JavaScript を使用して、**フォーム**および **カスタマイズ・フォーム**内の入力フィールドを検証するルーチンを作成できる。さらに、フォーム上の入力フィールドとボタンに対応する JavaScript のイベント・ハンドラを作成することもできる。

JavaServer Page

「**JSP**」を参照。

JSP

JavaServer Page。Web ページに対して簡単なプログラムによるインタフェースを提供する **サブレット**の拡張機能。JSP は、Web やアプリケーション・サーバー上で実行される専用のタグと埋込み Java コードが含まれた **HTML** ページである。HTML ページに動的機能を与える。JSP は最初にリクエストされたときに実際にサブレットにコンパイルされ、サブレット・コンテナ内で稼働する。

「**JSP タグ (JSP tags)**」も参照。

JSP タグ (JSP tags)

JSP 内に埋込み可能な Java コードを囲むタグ。JSP タグでは `<jsp:` 構文が使用され、JSP 内のアクション要素は **XML** 要素と同様に `begin` タグと `end` タグで囲まれる。

LBR

ロード・バランス・ルーター (load-balancing router)。Web リクエストを多数のサーバーに分散させる非常に高速なネットワーク・デバイス。ポータル・ユーザーに単一の公開アドレスを提供し、ユーザーが各リクエストを特定の**中間層**サーバーに送信せずに済むようにする。

LDAP

Lightweight Directory Access Protocol。ユーザーおよびグループのプロファイル情報を表し、これらにアクセスするための標準。

Lightweight Directory Access Protocol

「**LDAP**」を参照。

LOV

値リスト (list of values)。開発者が**フォーム**内の入力フィールドに選択可能な値を追加できる OracleAS Portal **ポートレット**。単一の値リストは、コンボ・ボックス、ラジオ・ボタン、チェック・ボックスなどの様々な形式で表示できる。

mod_oc4j

Oracle HTTP Server と **OC4J** 間の通信を管理する **Oracle HTTP Server** のモジュール。

mod_plsql

Oracle HTTP Server からのデータベース接続を処理する **Oracle HTTP Server** のモジュール。PL/SQL データベース・プロシージャで、Web ブラウザ内に表示できる書式設定されたデータと **HTML** コードが含まれた **HTTP** レスポンスを生成できるようにする。

OC4J

Oracle Application Server Containers for J2EE。すべてが Java で書かれた **Oracle Application Server** の **J2EE** サーバー・コンポーネント。標準 Java Development Kit (JDK) 仮想マシン (Java VM) 上で実行される。**JSP** トランスレータ、Java **サーブレット**・コンテナ、および Enterprise JavaBeans (JB) コンテナが含まれる。

OID

Oracle Internet Directory。OracleAS Portal ユーザー証明書と**グループ**・メンバーシップを格納するためのリポジトリ。デフォルトでは、分散しているユーザーおよびネットワーク・リソースについての OID 情報と照らしあわせて、**Oracle Application Server Single Sign-On Server** がユーザー資格証明を認証する。OID によって、LDAP バージョン 3 に、Oracle データベースの高いパフォーマンス、スケーラビリティ、堅牢性および可用性が取り込まれる。

OmniPortlet

スプレッドシート、XML および Web サービスのデータを、表、チャート、ニュース、箇条書き、フォーム・レイアウトとして表示できるポートレットを提供する Web プロバイダ。

OPCA

OracleAS Portal Configuration Assistant。OracleAS Portal をインストールして構成するための Java ベースの構成ツール。標準の **Oracle Application Server** インストールでは、OPCA はインストール後のフェーズで Oracle Universal Installer (OUI) によって実行される。OPCA をスタンドアロン・モードで起動することもできる。

OPPI

OracleAS Portal Partner Initiative。このイニシアティブは、OracleAS Portal に関して構築されるソリューションを構築、配信およびサポートする、主要なシステム・インテグレータ、コンテンツ・プロバイダおよびソフトウェア・ベンダーのコミュニティを育成するためのもの。Oracle の企業ポータルビジョンを共有し、企業ポータルの構築と管理のために Oracle のオープン・フレームワークを使用してソリューションを開発するサード・パーティに公開される。OracleAS Portal Partner Initiative 内のメンバーシップにより、範囲が広がり、メンバーが利用できる市場が大きくなる。メンバーは、自分の会社とその製品およびサービスに関する情報を **Portal Catalog** (<http://portalcatalog.oracle.com/>) に入力できる。Oracle との連携により、メンバーは価値あるコンテンツとアプリケーションを OracleAS Portal フレームワークに追加する。同時に、Oracle およびイニシアティブの他のメンバーによって提供されたコンテンツとサービスを活用することにより、パートナーはそれぞれの顧客により多くの利益を提供できる。詳細は、**Portal Center** (<http://portalcenter.oracle.com/>) の OPPI ページを参照。

Oracle Application Server

Oracle の統合アプリケーション・サーバーの特長は次のとおり。

- 標準に準拠している (**J2EE**、Web サービスおよび **XML**)。
- ポータル、**キャッシュ**、ワイヤレス、統合およびパーソナライズなど、包括的な機能のセットを提供する。
- Java と J2EE、Web サービス、XML、SQL および PL/SQL のための単一の統合プラットフォームを提供する。

Oracle Application Server

「**Oracle Application Server**」を参照。

Oracle Application Server Containers for J2EE

「**OC4J**」を参照。

Oracle Application Server Metadata Repository

アプリケーション・サーバー・コンポーネント（OracleAS Portal を含む）およびインフラストラクチャの他の部分で使用されるスキーマとビジネス・ロジックが含まれた Oracle データベース。

OracleAS Portal では、Oracle Application Server Metadata Repository 内のスキーマを使用して、ポータル・インスタンスに関連付けられたコンテンツとメタデータを格納および管理する。

Oracle Application Server Portal

Oracle Application Server のコンポーネント。企業クラスのポータルを開発、デプロイ、管理および構成するために使用する。OracleAS Portal では、ポータル内でアクセスされる情報を作成および管理できるようにするために、ポータル構築フレームワークとセルフサービスの公開機能を組み合わせている。

Oracle Application Server Portal Developer Kit

「PDK」を参照。

Oracle Application Server Single Sign-On Server

Oracle Application Server のコンポーネント。ユーザーが単一のユーザー名とパスワードを使用して Oracle Application Server 製品群のすべての機能、および他の Web アプリケーションにログインできるようにする。OracleAS Portal は、**パートナ・アプリケーション**として Oracle Application Server Single Sign-On に統合されており、認証をこれに委譲する。

Oracle Application Server Web Cache

Oracle Application Server のコンポーネント。頻繁に使用される Web サイトのパフォーマンス、スケーラビリティおよび可用性を向上させる。頻繁にアクセスされる URL をメモリー内に格納することによって、Oracle Application Server Web Cache では、Web サーバー上でこのような URL に対するリクエストを繰り返し処理する必要がなくなる。Oracle Application Server Web Cache は**失効化ベースのキャッシュ**を使用し、パフォーマンスの向上のために OracleAS Portal と統合されている。

「ポータル・キャッシュ (portal cache)」も参照。

Oracle Application Server Wireless

Oracle Application Server のコンポーネント。情報およびアプリケーションをモバイル・デバイスに配信するために使用する。Oracle Application Server Wireless を使用すると、Web ページ、カスタム Java アプリケーション、XML ベースのアプリケーションをはじめとする様々な種類のコンテンツを使用するカスタムのポータル・サイトを作成できる。Oracle Application Server Wireless により作成したサイトでは、これらの多様な情報にモバイル・デバイスからアクセスでき、ターゲットとなる端末のプラットフォームごとにコンテンツを書き直す必要はない。

Oracle Application Server Wireless XML

OracleAS Portal と **Oracle Application Server Wireless** との通信に使用する、デバイスに依存しないマークアップ言語。

Oracle Enterprise Manager

「**EM**」を参照。

Oracle HTTP Server

Oracle Application Server の **Web サーバー**・コンポーネント。Apache Web サーバーのテクノロジーに基づいて構築され、**HTTP** リクエストを処理するために使用される。Web と OracleAS Portal 間のリクエストを処理する **中間層**の一部。Oracle HTTP Server の拡張機能では、Java **サーブレット**、**JSP**、Perl、PL/SQL および CGI アプリケーションがサポートされる。

Oracle Internet Directory

「**OID**」を参照。

Oracle Technology Network

「**OTN-J**」を参照。

Oracle Text

Oracle9i の機能。Oracle リポジトリ内に格納されたコンテンツに対する拡張検索と検索サービスを提供する。OracleAS Portal に完全に統合されており、**Oracle Application Server Metadata Repository** の OracleAS Portal スキーマ内で管理されているコンテンツの全文検索を実行する機能をユーザーに提供する。また、**要旨別**および**テーマ別**に、結果を自動的にグループ化して分類する機能も提供する。

Oracle Ultra Search

Oracle Text に基づくアプリケーションであり、データベース、ファイル・システム、**Web サーバー**、電子メール・リストのアーカイブなど、複数の異種のリポジトリのクロール、索引作成および統合検索をサポートする。

「**検索ポータルレット (search portlet)**」と対比。

OracleAS Portal

「**Oracle Application Server Portal**」を参照。

OracleAS Portal Configuration Assistant

「**OPCA**」を参照。

OracleAS Portal Partner Initiative

「**OPPI**」を参照。

OTN-J

Oracle Technology Network Japan. オンラインの Oracle テクニカル・コミュニティ。Oracle9i に基づくアプリケーションを構築するための多様な技術上のリソースを提供する。OTN-J には、<http://otn.oracle.co.jp/> からアクセスできる。

Parallel Page Engine

「[PPE](#)」を参照。

PDK

OracleAS Portal Developer Kit. Web コンテンツとアプリケーションを構築して OracleAS Portal と統合するために使用する開発用フレームワーク。ポータルの開発を簡略化するのに役立つツールキット、サンプルおよび技術記事が含まれている。既存の Java [サブレット](#)、[JSP](#)、[URL](#) でアクセス可能なコンテンツおよび Web サービスを使用して、それらを [ポートレット](#) にすることができる。通常は、ポートレットとサービスを作成するために外部の開発者とベンダーが使用する。PDK は、開発者に最新のツールとテクニックを提供するために、[Portal Center](#) (<http://portalcenter.oracle.com/>) で定期的に更新される。

「[PDK-Java](#)」、「[PDK-PL/SQL](#)」および「[PDK-URL Services](#)」も参照。

PDK-Java

Java で [ポートレット](#) を実装し、ポータル機能を追加するためのツールキット。既存の Java [サブレット](#)、[JSP](#) および Web サービスを宣言によって [ポートレット](#) にするために使用する。

「[PDK](#)」も参照。「[PDK-PL/SQL](#)」と対比。

PDK-PL/SQL

PL/SQL プログラマが簡単に [ポートレット](#) を作成し、[PL/SQL API](#) を使用してそれらを拡張できるようにする一連の記事、サンプルおよびサービス。

「[PDK](#)」も参照。「[PDK-Java](#)」と対比。

PDK-URL Services

保護されたパブリックな Web コンテンツを宣言によって [ポートレット](#) にするユーティリティ。これらのサービスでは、ターゲット [URL](#) に動的にパラメータを渡すこと、コンテンツのクリッピングとリフォーマットを行うこと、フォーム・ベース認証または Basic 認証を必要とするアプリケーションに [SSO](#) を提供することが可能。また、これらのサービスによって、開発者は任意の言語で作成された任意のアプリケーションを使用して、簡単に統合 [ポートレット](#) を作成できるようになる。PDK-URL Services は、アプリケーションの [URL](#) を使用し、コンテンツを解析し、[PDK-Java](#) フレームワークを使用して [ポートレット](#) を作成する。

「[PDK](#)」も参照。

「PL/SQL」アイテム・タイプ (PL/SQL item type)

コンテンツ作成者がページに追加できるデフォルトのアイテム・タイプの1つ。PL/SQL アイテムには、1ブロックのPL/SQLコードが含まれている。ユーザーがこのアイテムをクリックすると、そのブロックが実行される。結果は、ユーザーのブラウザに表示される。PL/SQL アイテムは、ページに直接表示することもできる。

PL/SQL ファンクション (PL/SQL function)

「ファンクション (function)」を参照。

PL/SQL ページ・タイプ (PL/SQL page type)

OracleAS Portal によってサポートされるページ・タイプの1つ。PL/SQL ページには、そのページがレンダリングされるときに HTML を生成する PL/SQL コードが含まれている。

Portal Catalog

OracleAS Portal に関連する付加価値のある製品とサービスに関する情報源。主に、OracleAS Portal パートナと、これらパートナが提供する製品とサービスに関する情報が含まれている。このカタログは、Portal Center (<http://portalcenter.oracle.com/>) にある。または、<http://portalcatalog.oracle.com/> に直接アクセスすることもできる。ユーザーは、カタログ内で様々なパラメータに基づいてポートレットを検索できる。キーワード以外に、ポートレットのカテゴリ、プロバイダ・タイプ、OracleAS Portal のサポートバージョン、会社名、場所、会社のカテゴリなど。ユーザーはツリー制御を利用してカタログを参照することもできる。興味のあるカテゴリを展開すると、参加しているパートナの会社、製品およびサービスが表示される。このカタログは、パートナにとってもセルフサービスの利点がある。つまり、パートナは自分のエントリを完全にオンラインで送信および管理できる。

Portal Center

OracleAS Portal について知りたいあらゆることを検出できる Web サイト (<http://portalcenter.oracle.com/>)。頻繁に更新され、常に最新の製品情報があり、Portal Developer Services と Portal Catalog にとってのホーム。この Web サイトには、製品に関するすべての情報 (ドキュメントやデモも含む) があり、OracleAS Portal の専門知識へのアクセスを提供する。

Portal DB プロバイダ (Portal DB provider)

「データベース・プロバイダ (database provider)」を参照。

Portal Developer Kit

「PDK」を参照。

Portal Developer Services

OracleAS Portal の専門知識を開発および交換する人々のためのネットワーク。このプログラムは、他の場所と同様に、**Portal Center** (<http://portalcenter.oracle.com/>) を介してオンライン・テスト・ツールを提供する。また、ニュースレター、サーベイおよび **Portal Knowledge Exchange** を通じて制作チームや他の Portal 開発者と対話できる。

「**Portal コミュニティ (Portal Community)**」も参照。

Portal Knowledge Exchange

Portal Developer Services の加入者が、ホワイトペーパー、テクニックおよびポートレットをコミュニティ内の他の人々と共有できる、セルフサービスの Web サイト。

Portal Studio

ポータル開発者のニーズにあわせて特別に調整された、**Portal Center** (<http://portalcenter.oracle.com/>) の主要コンポーネント。機能とサービスには、大規模なテクニカル・ライブラリ、サンプル・コード、開発者専用のディスカッション・フォーラム、およびマルチメディアのトレーニング教材が含まれている。開発者は、自分用の OracleAS Portal のコピーをインストールしなくても、ここにある情報とホスティングされているリソースを使用して、**ポートレット**を登録してテストできる。
<http://portalstudio.oracle.com/> で OracleAS Portal Studio に直接アクセスすることもできる。

Portal コミュニティ (Portal Community)

OracleAS Portal に関する情報を作成および交換する人々のためのネットワーク。このコミュニティには、OracleAS Portal を使用するすべての人々が含まれる。Portal 関連の情報を共有するために **Portal Knowledge Exchange** を活用し、**Portal Developer Services** を利用できる。

「Portal スマート・テキスト」アイテム・タイプ (Portal smart text item type)

自己構成型の**ナビゲーション・アイテム・タイプ**。たとえば、「現在の日付」スマート・テキスト・アイテムを**ナビゲーション・ページ**に追加した場合は、現在の日付がサーバーから自動的に取得されるため、複雑なコーディングを指定する必要はない。

「Portal スマート・リンク」アイテム・タイプ (Portal smart link item type)

自己構成型の**ナビゲーション・アイテム・タイプ**。たとえば、「ホーム」スマート・リンクを**ナビゲーション・ページ**に追加した場合は、ユーザーが「ホーム」リンクをクリックすると、自動的にその**ホーム・ページ**に移動する。

「Portal ビルダー」ページ (Portal Builder page)

開発および管理用の**ポートレット**が含まれた事前定義済みの**ページ**。OracleAS Portal **オブジェクト**とサービスを構築して管理するために使用する。

PPC

ページの部分的なキャッシュ (Partial Page Caching)。Oracle Application Server Web Cache によって HTML ドキュメントのフラグメントを個別にキャッシュおよび管理可能にする機能。テンプレート・ページに Edge Side Includes (ESI) マークアップ・タグを使用することによって HTML フラグメントを取得および挿入するように Oracle Application Server Web Cache に指示する。フラグメント自体は、細分化されたテキストまたはその他のオブジェクトが含まれている HTML ファイル。

PPE

Parallel Page Engine。OC4J コンテナ内で実行され、ページ・リクエストを処理するマルチスレッド・サーブレット・エンジン。PPE は、ページ・メタデータを読み取り、ポートレット・コンテンツを得るためにプロバイダをコールし、プロバイダのレスポンスを受け取って、指定されたページ・レイアウトに従ってリクエストされたページを収集する。Parallel Page Engine は、OracleAS Portal 中間層の一部。

Simple Object Access Protocol

「SOAP」を参照。

Single Sign-On

「Oracle Application Server Single Sign-On Server」を参照。

SOAP

Simple Object Access Protocol。非集中型の分散した環境内で情報を交換するための、軽量な XML ベースのプロトコル。SOAP では、リモート・プロシージャ・コール (RPC) やメッセージ指向型の情報交換をはじめとする様々なスタイルの情報交換がサポートされる。RPC スタイルの情報交換では、リクエスト / レスポンス型の処理が可能であり、エンドポイントでプロシージャ指向のメッセージを受け取り、それに関連するレスポンス・メッセージが返される。メッセージ指向型の情報交換では、メッセージを送っても送信側が迅速なレスポンスを期待したり待機しない可能性がある、ビジネス関連などのドキュメントの交換を必要とする企業やアプリケーションをサポートする。

SSL アクセラレータ・カード (SSL accelerator card)

通常の SSL ソフトウェアよりもかなり高速にトラフィックを処理するハードウェア・デバイス。

「LBR」も参照。

SSO

Single Sign-On。

「Oracle Application Server Single Sign-On Server」を参照。

Ultra Search

「Oracle Ultra Search」を参照。

Uniform Resource Locator

「[URL](#)」を参照。

URL

Uniform Resource Locator。インターネットを通じて使用可能なリソースの場所を示す簡潔な文字列表現。URL は、Web クライアントが [Oracle Application Server](#) に対するリクエストをエンコードするために使用する形式でもある。

「URL」 アイテム・タイプ (URL item type)

[コンテンツ作成者](#)がページに追加できるデフォルトの [アイテム・タイプ](#) の 1 つ。URL アイテムは、クリックされたときに、別の Web ページへの経路を提供する。ユーザーが URL アイテムの [表示名](#) をクリックすると、その [URL](#) が参照している Web ページが表示される。

URL ページ (URL page)

その [URL](#) で識別される、別の Web ページへの経路を提供する [ページ・タイプ](#)。ユーザーが [ページ・リンク](#) をクリックすると、そのリンクが参照している Web ページが表示される。

URL ポートレット (URL portlet)

[URL](#) に指定された Web ページのコンテンツが表示される OracleAS Portal [ポートレット](#)。

VPD

仮想プライベート・データベース (Virtual Private Database)。Oracle データベースを活用して顧客をホスティングする [ASP](#) のための機能。基本的には、すべての顧客に対して 1 つの物理データベース・インスタンスを使用するが、各顧客にとっては、独自のデータベースを所有しているように見える。ユーザーは自分に関係のない情報を見ることができず、顧客は完全に分離される。作業の大部分はデータベース・レベルで行われるため、中核となるアプリケーションで実施する変更はわずかであるか、またはまったく必要ない。VPD の実装には基本的に 2 つの主要手順が必要である。最初にコンテキスト列 (会社名など) をすべてのデータベース表に追加し、次に、ログイン・ユーザーのコンテキストに基づいて各表の間合せを制限するポリシーを実装する。VPD では、この方法を使用して安全性の高い完全なサブスクライバの分離を提供する。

WAP

Wireless Application Protocol。ワイヤレス・ネットワークを使用するアプリケーションとサービスを開発するための、オープンな一連のグローバル・プロトコル。

Web Cache

「[Oracle Application Server Web Cache](#)」を参照。

WebDAV

Web-based Distributed Authoring and Versioning。分散オーサリングおよびバージョン管理をサポートする、[HTTP 1.1](#) のプロトコル拡張。WebDAV を使用すると、インターネットを透過的な読取りおよび書込みメディアとして利用できるようになる。つまり、[URL](#) アドレスに基づいてコンテンツをチェックアウト、編集およびチェックインできる。

Web クリップング (Web clipping)

ページ設計者が Web コンテンツを収集して単一のポータルに集約できるようにする機能。大規模な組織に散在している多数の様々な Web サイトのコンテンツを統合するために使用できる。

Web サーバー (Web server)

Web ページを配信するプログラム。

Web プロバイダ (Web provider)

OracleAS Portal によって [HTTP](#) リクエストを介してコールされ、[HTML](#)、[XML](#) または [WML](#) の [ポータルレット](#)・コンテンツを戻すエンティティ。Web プロバイダは、特定のアプリケーション環境 (Java、ASP、Perl など) 内で定義された 1 つ以上のポータルレットのプロキシとして動作し、OracleAS Portal の外部のアプリケーションとして実行される。Web プロバイダは、Web アクセス可能な情報源に特に適している。

「[プロバイダ \(provider\)](#)」も参照。「[データベース・プロバイダ \(database provider\)](#)」と対比。

Wireless Application Protocol

「[WAP](#)」を参照。

Wireless Markup Language

「[WML](#)」を参照。

WML

Wireless Markup Language。携帯用デバイス向けにハイパーリンクのようなコンテンツとアプリケーションを定義するために使用する [XML](#) ベースのマークアップ言語。

XML

Extensible Markup Language。SGML 構文のサブセットを使用してデータを記述するためのオープン標準。

XML ポータルレット (XML portlet)

[XML](#) コードの実行結果を表示する OracleAS Portal [ポータルレット](#)。このポータルレットを作成するには、XML コードまたは XML コードを指す [URL](#) を指定する。

「Zip ファイル」アイテム・タイプ (zip file item type)

コンテンツ作成者がページに追加できるデフォルトのアイテム・タイプの1つ。Zip ファイル・アイテムによって、1回の操作で多数のファイルをアップロードできるようになる。このアイテムを使用して、ファイル・システムまたは Web サイトのコンテンツを **Oracle Application Server Metadata Repository** の OracleAS Portal スキーマに移行できる。Zip ファイルを OracleAS Portal にアップロードし、その後アップロードしたファイルを解凍すると、各ディレクトリについてページが作成され、各ファイルについてアイテムが作成される。このアイテムは、ターゲット・ページに公開される。

アイテム (item)

ページ上のアイテム領域にあるコンテンツの個々の要素 (テキスト、ハイパーリンク、イメージなど)。適切な権限レベルを持つユーザーが、アイテムをページに追加できる。アイテムのコンテンツとメタデータは、**Oracle Application Server Metadata Repository** の OracleAS Portal スキーマ内に格納される。アイテムは、アイテム領域に定義されたレイアウト、スタイルおよび属性表示に従って、ページ上にレンダリングされる。

「アイテム・タイプ (item type)」も参照。

アイテム ID (item ID)

アイテムのコンテンツへのローカル・データベース参照。アイテム ID の値は、アイテムを PL/SQL プロシージャに渡すためにカスタム・アイテム・タイプ内で使用される。ファンクションはアイテム ID を使用してアイテムのコンテンツにアクセスする。

アイテム・タイプ (item type)

アイテムのコンテンツとアイテムについて格納される属性 (メタデータ) を定義する。アイテム・タイプは、コンテンツ・アイテム・タイプとナビゲーション・アイテム・タイプに分類される。

カスタム・アイテム・タイプは、アイテム・タイプで提供される機能を拡張し、アイテムに関する追加の属性情報を格納するために、ページ・グループ管理者によって作成される。

アイテム・バージョンング (item versioning)

「バージョンング (versioning)」を参照。

アイテム・レベルのセキュリティ (item level security)

「ILS」を参照。

アクセス制御リスト (access control list)

「ACL」を参照。

値リスト (list of values)

「LOV」を参照。

アプリケーション (application)

廃止された用語。「[データベース・プロバイダ \(database provider\)](#)」を参照。

アプリケーション・サービス・プロバイダ (Application Service Provider)

「[ASP](#)」を参照。

一時表領域 (temporary tablespace)

表の行のソートなどの操作で一時的な表セグメントを作成するために使用する、データベース内の領域の割当て。

一括ロード (bulk load)

「[Zip ファイル](#)」 [アイテム・タイプ \(zip file item type\)](#)」を参照。

イベント (event)

「[ページ・イベント \(page event\)](#)」を参照。

イベント・サーブレット (Event servlet)

イベント対応 [ポートレット](#) へのアクセス時に動的なページ・ナビゲーションを可能にするための OracleAS Portal の機能を実装する。[PPE](#) と同じコンテナ内で実行される。

「イメージ」アイテム・タイプ (image item type)

[コンテンツ作成者](#)が [ページ](#) に追加できるデフォルトの [アイテム・タイプ](#) の1つ。JPEG、GIF または PNG 形式のイメージを追加できる。

「イメージ・マップ」アイテム・タイプ (image map item type)

[コンテンツ作成者](#)が [ページ](#) に追加できるデフォルトの [アイテム・タイプ](#) の1つ。イメージ・マップは、クリックすると他の [URL](#) にリンクするホットスポットが付いた単一のイメージ。たとえば、各大陸がその大陸に関する詳細情報へのハイパーリンクとなっている世界地図のイメージ・マップを作成できる。

インポート (import)

コンテンツと [オブジェクト](#) ([ページ・グループ](#)、[ページ](#)、[ポートレット](#)など) を OracleAS Portal インスタンスにトランスポートする方法。たとえば、ある OracleAS Portal インスタンスの [ページ](#)、関連付けられた [スタイル](#)、およびそのコンテンツを別のインスタンスにインポートできる。

インポートの置換モード (Replace on Import mode)

インポートのモード。このオプションを選択すると、オブジェクトがターゲットに存在する場合は、そのオブジェクトが置き換えられる。オブジェクトが存在しない場合は、作成される。このオプションを選択しないと、オブジェクトがターゲットに存在する場合は、そのオブジェクトが参照される。オブジェクトがターゲットに存在しない場合は、作成される。

ウィザード (wizard)

プロセス全体を順を追ってユーザーに示すグラフィカル・インタフェース。OracleAS Portal では、[データベース・プロバイダ](#)、[ポートレット](#)、データベース・[オブジェクト](#)、[ページ](#)、[ページ・グループ](#)および[アイテム](#)を作成するためにウィザードが使用される。

上書きモード (Overwrite mode)

「[インポートの置換モード \(Replace on Import mode\)](#)」を参照。

永続的なリンク (durable link)

アイテムを一意に識別するためにアイテムの GUID (グローバル一意識別子) を使用する、アイテムへのリンク。アイテムの GUID は変わらないため、アイテムの編集、名前の変更、移動、別のポータル・インスタンスへのインポートが行われても永続的なリンクは有効である。

「[ダイレクト・アクセス URL \(direct access URL\)](#)」と対比。

エクスポート (export)

単一の OracleAS Portal インスタンスの[ページ・グループ](#)、[ページ](#)、[ポートレット](#)などのコンテンツを含む一連のファイル ([トランスポート・セット](#)) を作成する方法。作成後、このファイルのセットを別の Oracle Application Server インスタンスに[インポート](#)できる。

お気に入り (favorite)

頻繁に参照される [URL](#) にすばやくアクセスできるようにするお気に入り [ポートレット](#) 内のハイパーリンク。会社のファイアウォールの内側でも外側でもかまわない。[認証されたユーザー](#)は、自分が頻繁にアクセスする [URL](#) のセットでお気に入り [ポートレット](#) をカスタマイズできる。

お気に入りグループ (favorite group)

通常は論理的に関連付けられた[お気に入り](#) (およびお気に入りグループ) の集合。

オブジェクト (object)

1. OracleAS Portal オブジェクト。[ページ・グループ](#)、[ポートレット](#)、[ページ](#)、[スタイル](#)などの構造。
2. データベース・オブジェクト。表、プロシージャ、トリガーなどの Oracle データベース構造。これらのオブジェクトは、OracleAS Portal ウィザードまたは Oracle データベース・コマンドを使用して作成できる。

「オブジェクトのリスト」アイテム・タイプ (list of objects item type)

ユーザーが[ページ](#)に追加できる[ナビゲーション・アイテム・タイプ](#)。このアイテム・タイプはオブジェクト (ページや[パースペクティブ](#)など) のリストであり、ドロップダウン・リストとして表示するか、リンク (関連イメージを付けて、または付けずに) として表示するかをユーザーが選択できる。

「オブジェクト・マップ・リンク」アイテム・タイプ (object map link item type)

ユーザーがページに追加できる **ナビゲーション・アイテム・タイプ**。ポータル内で使用可能な **オブジェクト** のマップを表示する。

階層 (hierarchy)

自己参照型の表またはビューからのデータを表示する OracleAS Portal **ポートレット**。表内では少なくとも 2 つの列が再帰的な関係を共有している必要がある。階層には最大 3 レベルを含めることができ、組織チャート内の従業員などのデータや、Web サイト内のメニュー間の階層関係を表示できる。

開発者 (developer)

他のユーザーが自分のページに含めるためのポートレットを構築する。OracleAS Portal の機能を拡張するための API を重点的に利用するため、アドバイスまたはヒントを求めて **Portal Knowledge Exchange** またはフォーラムを頻繁に参照する。

外部アプリケーション (external application)

通常は外部アプリケーション・**ポートレット** から起動される、OracleAS Portal の外部のアプリケーション。各外部アプリケーションは **ポータル管理者** によって構成されるため、ユーザーはユーザー名とパスワード情報を指定するのみでよい。**Oracle Application Server Single Sign-On Server** では、将来の認証要求に対応してこれらの資格証明を提示する予定。

外部オブジェクト (external object)

エクスポート・オブジェクトのカテゴリ。**明示オブジェクト** の外部依存性であるオブジェクト。外部オブジェクトによって、明示オブジェクトはターゲットのポータル上で機能するようになる。

「**マニフェスト (manifest)**」も参照。

拡張検索 (advanced search)

ユーザーが次のことを実行できるようにする検索エンジン。

- 検索文字列内にある任意の文字列またはすべての文字列が含まれるコンテンツを検索する。
- 選択した **ページ・グループ**、またはすべての **ページ・グループ** を検索する。
- 検索の範囲を、特定の **ページ**、**カテゴリ**、**パースペクティブ**、**アイテム・タイプ** または **属性** に制限する。

Oracle Text がインストールされて有効になっている場合は、拡張検索を使用して、近接、同音語およびファジー検索も実行できる。

「**検索ポートレット (search portlet)**」も参照。「**基本検索 (basic search)**」および「**カスタム検索 (custom search)**」と対比。

カスタマイズ・フォーム (customization form)

OracleAS Portal [ポートレット](#)に渡す値の入力をエンド・ユーザーに求めるページ。エンド・ユーザーがポートレットの「[カスタマイズ](#)」リンクをクリックすると、ポートレットのカスタマイズ・フォームが表示される（作成されている場合）。

カスタム・アイテム・タイプ (custom item type)

「[アイテム・タイプ \(item type\)](#)」を参照。

カスタム検索 (custom search)

[Oracle Application Server Metadata Repository](#) の OracleAS Portal スキーマに格納された情報に対する様々な検索をユーザーが定義できるようにする機能。カスタム検索ポートレットのデフォルトのカスタマイズを編集すると、特定の検索要件を満たす一意の検索送信フォームおよび結果ページを定義したり、事前定義済の検索基準に基づいて検索を実行して結果を戻すポートレットを構成できる。

「[検索ポートレット \(search portlet\)](#)」も参照。「[拡張検索 \(advanced search\)](#)」および「[基本検索 \(basic search\)](#)」と対比。

カスタム属性 (custom attribute)

「[属性 \(attribute\)](#)」を参照。

カスタム・プロバイダ (custom provider)

顧客固有のコンテンツまたはアプリケーションにアクセスする [ポートレット](#) を作成および管理できるようにする [プロバイダ](#) のタイプ。宣言またはプログラムを使用して、OracleAS Portal にカスタム・ポートレットを作成できる。

カスタム・ページ・タイプ (custom page type)

「[ページ・タイプ \(page type\)](#)」を参照。

仮想プライベート・データベース (virtual private database)

「[VPD](#)」を参照。

仮想プライベート・ポータル (virtual private portal)

同じポータル・インスタンス内で複数の会社または組織を安全にホスティングするための OracleAS Portal 機能。

カテゴリ (category)

[ページ・グループ](#) 内の [ページ](#)、[アイテム](#) および [ポートレット](#) をグループ化または分類するために使用する事前定義済の [属性](#)。ユーザーがアイテムまたはページの用途を理解するのに役立つ。たとえば、旅行に関する [ページ・グループ](#) には、地図、スナップ写真、ホテル・ガイドなどのカテゴリを設定できる。個々のアイテムまたはページに割り当てることができるカテゴリは1つのみである。

「[パースペクティブ \(perspective\)](#)」と対比。

カレンダー (calendar)

SQL [問合せ](#)の結果をカレンダー形式で表示する OracleAS Portal [ポートレット](#)。

簡潔な URL (pretty URL)

「[ダイレクト・アクセス URL \(direct access URL\)](#)」を参照。

キーワード (keyword)

ユーザーが検索中に簡単に[ページ](#)または[アイテム](#)を発見できるように、ページまたはアイテムに関する追加情報を提供するために使用する[属性](#)。

企業ポータル (enterprise portal)

企業ポータルは、関連する企業情報源へのパーソナライズされたアクセスを提供する共通の総合的な入り口。企業ポータルでは、パブリックなインターネットで利用できるリソースのビューをサイト訪問者がカスタマイズできる。

基本アイテム・タイプ (base item type)

「[アイテム・タイプ \(item type\)](#)」を参照。

基本検索 (basic search)

ユーザーが特定の検索文字列を含むコンテンツを検索できるようにする機能。

「[検索ポートレット \(search portlet\)](#)」も参照。「[拡張検索 \(advanced search\)](#)」および「[カスタム検索 \(custom search\)](#)」と対比。

「基本検索ボックス」アイテム・タイプ (basic search box item type)

他のユーザーが検索できるように、ユーザーが[ページ](#)に追加できる[ナビゲーション・アイテム・タイプ](#)。検索ボックスのユーザーがすべての[ページ・グループ](#)を検索できるか、選択した[ページ・グループ](#)のみを検索できるかを指定できる。

基本属性 (base attribute)

「[属性 \(attribute\)](#)」を参照。

キャッシュ (caching)

頻繁にアクセスされる情報（通常は Web ページや OracleAS Portal 内の[ポートレット](#)など）をすばやくアクセスできる場所に格納し、コンテンツが頻繁に生成されるのを回避するための操作。たとえば、[Oracle Application Server Web Cache](#) は、動的に生成されたポートレットをメモリー内に格納し、指定のポートレットがリクエストされたときに [PPE](#) に提供する。この記憶域を使用すると、バックエンド・データベースや他の Web サイトへの接続が行われないため、リクエストの処理にかかる合計時間が短縮される。

「[有効期限ベースのキャッシュ \(expiry-based caching\)](#)」、「[失効化ベースのキャッシュ \(invalidation-based caching\)](#)」、「[システム・レベルのキャッシュ \(system level caching\)](#)」および「[妥当性チェックベースのキャッシュ \(validation-based caching\)](#)」も参照。

行 (row)

表内の値のセット。たとえば、SCOTT.EMP 表内の 1 人の従業員を表す値のセットなど。

共有オブジェクト (shared object)

ページ・グループ間で共有できる、個人用ページ、ナビゲーション・ページ、スタイル、ページ・テンプレート、パースペクティブ、カテゴリなどの OracleAS Portal オブジェクト。

共有画面モード (Shared Screen mode)

ポートレットの本体がレンダリングされるポートレット表示モード。各ポートレットには、少なくとも共有画面モードが必要である。

共有ポートレット (shared portlet)

OracleAS Portal のデータベース・プロバイダ内の他のポートレット間で共有される OracleAS Portal ポートレット。各共有ポートレットは、同じようにパーソナライズして、複数のページに表示できる。

クラスタ (cluster)

相互に関係があり、通常ディスク上の同じ領域で結合される表を格納するために使用するデータベース・オブジェクト。

グラフィカル・ビュー (graphical view)

ページ上の所定の位置にページ・コンテンツをレンダリングするページ編集ビュー。グラフィカル・ビューを使用すると、編集時に最終的なページと同じようにページとアイテムを表示できる。

「レイアウト・ビュー (layout view)」および「リスト・ビュー (list view)」と対比。

グループ (group)

たとえば、人事や会計などのように、通常、共通のニーズまたは利害を共有する OracleAS Portal ユーザーの集合。グループを使用すると、オブジェクト (ページやポートレットなど) へのアクセス権を一度に複数のユーザーに与えることが容易になる。また、ロールに関する権限をグループに割り当て、そのロール内のユーザーを追加することによって、ユーザー・ロールを実装できる。OID によって、OracleAS Portal グループのメンバーシップが追跡される。

グループの所有者 (group owner)

グループに対してメンバーの追加や削除を行う権限、またはグループ自体を削除する権限を持ったユーザー。グループには複数の所有者を設定できる。

グローバル権限 (global privilege)

特定のタイプのすべてのオブジェクトに対する一定レベルのアクセス権をユーザーまたはグループに付与する権限。たとえば、Web 設計者グループにすべてのスタイルに対する「管理」権限を付与できる。

権限 (privilege)

OracleAS Portal で、アクションを実行する権利。権限は、グローバルなもの（ユーザーまたはグループ・プロファイルで設定）あるいは特定の**オブジェクト**に限られるもの（通常はオブジェクトの「アクセス」タブで設定）のいずれかになる。アプリケーションを構築するときに、データベース・オブジェクト、共有ポートレット、ポートレットおよびアプリケーションに対してもアクセス権が付与される。

権限を持ったユーザー (authorized user)

「**認証されたユーザー (authenticated user)**」を参照。

現行バージョン (current version)

ページに表示されている**アイテム**のバージョン。現行バージョンは必ずしもアイテムの最新バージョンではない。

「**バージョンング (versioning)**」も参照。

検索ポートレット (search portlet)

Oracle Application Server Metadata Repository の OracleAS Portal スキーマ内の**ページ**とコンテンツをユーザーが検索できるようにするポートレット。ユーザーは、CONTAINS、GREATER THAN、LESS THAN、EQUAL TO などの演算子を使用して、テキスト文字列、カテゴリ、パースペクティブおよび属性に基づいた検索を実行することもできる。

「**Oracle Ultra Search**」と対比。

コール・インタフェース (call interface)

OracleAS Portal **ポートレット**が最初に作成されたとき、または最後に編集されたときに選択された引数を表示する。

構成済 UI テンプレート (structured UI template)

OracleAS Portal **ポートレット**の外観を制御し、スタンドアロン・モードで実行される**共有ポートレット**。構成済 UI テンプレートによって、このテンプレートを使用する各ポートレットに関して同じ位置に同じイメージとテキストが表示される。

「**ユーザー・インタフェース (UI) テンプレート (user interface (UI) template)**」を参照。「**非構成済 UI テンプレート (unstructured UI template)**」と対比。

子オブジェクト (child object)

階層の一部であるオブジェクト。たとえば、サブページ、サブカテゴリ、サブパースペクティブは、それぞれ**ページ**、**カテゴリ**および**パースペクティブ**の子オブジェクトである。

「**マニフェスト (manifest)**」も参照。

個人用ページ (personal page)

認証されたユーザーが個人的なコンテンツを格納し、それを他のユーザーと共有できる OracleAS Portal 内の領域。ポータル管理者は、ユーザー・アカウントを作成するときに、そのユーザーのための個人用ページの作成を選択できる。

コミュニティ (community)

「Portal コミュニティ (Portal Community)」を参照。

コンテンツ・アイテム・タイプ (content item type)

ページにアップロードされるドキュメント、テキスト、イメージなどのアイテムの実際のコンテンツを識別する方法。

次の組込みコンテンツ・アイテム・タイプがある。

- 「ファイル」アイテム・タイプ および 「シンプル・ファイル」アイテム・タイプ
- 「テキスト」アイテム・タイプ および 「シンプル・テキスト」アイテム・タイプ
- 「URL」アイテム・タイプ および 「シンプル URL」アイテム・タイプ
- 「イメージ」アイテム・タイプ および 「シンプル・イメージ」アイテム・タイプ
- 「イメージ・マップ」アイテム・タイプ
- 「PL/SQL」アイテム・タイプ および 「シンプル PL/SQL」アイテム・タイプ
- 「ページ・リンク」アイテム・タイプ および 「シンプル・ページ・リンク」アイテム・タイプ
- 「Zip ファイル」アイテム・タイプ

「アイテム・タイプ (item type)」も参照。「ナビゲーション・アイテム・タイプ (navigation item type)」と対比。

コンテンツ作成者 (content contributor)

ページにアイテムを追加するための適切な権限を持つユーザー。適切なページ権限には、「コンテンツの管理」および「承認付きアイテムの管理」が含まれる。適切なアイテム権限には、「管理」、「編集」および「表示」が含まれる。

コンテンツ領域 (content area)

廃止された用語。「ページ・グループ (page group)」を参照。

コンパクト HTML (compact HTML)

「CHTML」を参照。

コンポーネント (component)

廃止された用語。「ポートレット (portlet)」を参照。

サーブレット (servlet)

通常は **Web サーバー** で実行される Java プログラムで、Web サーバーの機能を拡張する。**HTTP** サーブレットは、クライアント HTTP リクエストを受け取り、動的コンテンツを作成（データベースに問い合わせるなどにより）して、HTTP レスポンスを返す。

サーベイ (survey)

ユーザーから情報を引き出すために使用する一連の質問。サーベイでは、特定の質問に対する回答に応じてユーザーをサーベイの様々なセクションに振り分けることができる。

「**投票 (poll)**」および「**テスト (test)**」と対比。

最近使用したオブジェクト (recent object)

最近表示または編集された、**ページ**や**ポートレット**などの OracleAS Portal **オブジェクト**。各**認証されたユーザー**には、最近アクセスした n 個のオブジェクトへのリンクを提供する独自の最近使用したオブジェクト・ポートレットがある。

再使用モード (Reuse mode)

「**インポートの置換モード (Replace on Import mode)**」を参照。

索引 (index)

表に関連付けられたオプションの体系。表の行を迅速に検索し、(オプションで) 個々の行が一意であることを保証するために使用される。

サブアイテム (sub-item)

別の (親) アイテムの下の階層に存在する**アイテム**。これにより、関連性の強いアイテムをグループ化できる。たとえば、特定の HTML ファイル・アイテムで使用するスプレッドシートを、その HTML ファイル・アイテムのサブアイテムとして追加できる。

サブカテゴリ (sub-category)

別の (親) カテゴリの下の階層に存在する**カテゴリ**。これにより、関連性の強いカテゴリをグループ化できる。

サブスクライバ (subscriber)

ASP にサインアップし、ホスティングされた OracleAS Portal 上で**ストライブ**を受け取る会社。

サブパースペクティブ (sub-perspective)

別の (親) パースペクティブの下の階層に存在する**パースペクティブ**。これにより、関連性の強いパースペクティブをグループ化できる。

サブページ (sub-page)

別の (親) ページの下の階層に存在する**ページ**。ページ・グループ内の各ページは、ルート・ページを除きすべてサブページ。

参照しているオブジェクト (referenced object)

エクスポート・オブジェクトのカテゴリ。明示オブジェクトが直接または間接的に参照しているオブジェクト。

「マニフェスト (manifest)」も参照。

参照パス (reference path)

ページ上でポートレット・インスタンスを一意に識別するパス。参照パスは、特定のページの個別のポートレットに対するパラメータを対象として設定するために使用できる。

識別名 (Distinguished Name)

「DN」を参照。

自己登録 (self registration)

ログイン・ポートレット内のリンクを通じて、ユーザーが自分用の新しいアカウントを作成できるようにする機能。

システム・パージ (system purge)

Oracle Application Server Metadata Repository の OracleAS Portal スキーマから、ページ・グループ内で削除または失効とマークされたすべてのアイテムを削除する機能。システム・ページは、ページ・グループ管理者またはポータル管理者によって実行される。

システム・レベルのキャッシュ (system level caching)

システム・レベルのキャッシュでは、すべてのユーザーについて、オブジェクトの単一コピーがシステム・キャッシュ (中間層上) に置かれる。その結果、個別のユーザーが、そのオブジェクトの独自のインスタンスに設定したすべてのカスタマイズ・オプションは無効になる。また、オブジェクトのアクセス権限は実施されない。システム・レベルのキャッシュに適したコンテンツの例として、ページ・バナーとニュース・ポートレットがある。

失効化ベースのキャッシュ (invalidation-based caching)

Oracle Application Server Web Cache で使用されるキャッシュ方式。アイテムの更新が必要となるイベントが発生するまで、アイテムはキャッシュ内に残る。たとえば、ユーザーがアイテムを更新すると、キャッシュの更新が必要になる。イベントにตอบสนองして、ポータルまたはプロバイダは失効メッセージを Oracle Application Server Web Cache に送信する。失効化されたアイテムに対するリクエストが次に発生したときに、そのアイテムはキャッシュ内で更新される。

Oracle Application Server Web Cache 内に保存されたデータは、失効化されるか期限切れになるまで、有効であるとみなされる。Oracle Application Server Web Cache 内にキャッシュされた情報が不正確になったときは、失効化する必要がある。たとえば、Oracle Application Server Web Cache 内に保存されたページ・メタデータは、ページ設計者がページの構造を変更したとき、またはユーザーの権限が変わったときに失効化される。ポートレットは、エンド・ユーザーがカスタマイズすると失効化される可能性がある。

「有効期限ベースのキャッシュ (expiry-based caching)」および「妥当性チェックベースのキャッシュ (validation-based caching)」も参照。

失効通知 (expiry notification)

ユーザーまたはグループに自動的に送信され、ページのアイテムの有効期限が近いことを示すメッセージ。この通知は、ページ・グループ管理者によって設定される。

シノニム (synonym)

表またはビューに割り当てられた追加の名前。後で参照のために使用できる。

主キー (primary key)

表中の行を識別するために使用できる一意の値で構成されるデータベース表の列。

順序 (sequence)

表の行に対して自動的に番号を生成するために使用するデータベース・オブジェクト。

承認申請通知 (subscription notification)

エンド・ユーザーが特定のページまたはアイテムをサブスクライブするための方法。そのページまたはアイテムが更新されると、通知ポートレット経由で通知される。ページ設計者は、ユーザーがページをサブスクライブできるように「承認申請」「Portal スマート・リンク」アイテム・タイプを含め、ユーザーがアイテムをサブスクライブできるように「承認申請」属性を表示する必要がある。さらに、ページ・グループ管理者はページ・グループに対して承認と通知を有効にする必要がある。

承認通知 (approval notification)

アイテムの作成または更新に承認が必要であることを示す通知ポートレット内のメッセージ。承認通知は、承認プロセスで指定された承認者のリストに送信される。承認者は、対象のアイテムを承認または却下することによって通知に応答する。

承認プロセス (approval process)

1つ以上の一連の手順。この中で、新しく作成または更新されたアイテムが公開前に承認される必要がある。承認プロセス内の各手順には1人以上の承認者が必要であり、各承認者は、ユーザーまたはグループのいずれかである。承認者へのルーティングは、順次（一度に1人ずつ）または並行（一度に全員）して実行できる。各手順は、任意の1人のメンバーまたはすべてのメンバーによる応答（承認または却下のいずれか）を必要とするように定義できる。1つの手順で必要な数の応答を受信すると、プロセスは次の手順に進む。アイテムが却下されるか、または最終手順に達してドキュメントが承認されると、プロセスは終了する。

情報モード (About mode)

ポートレット表示モードのオプションの1つ。ポートレットの著作権、バージョンおよび作成者に関する情報が表示される。

「シンプル PL/SQL」アイテム・タイプ (simple PL/SQL item type)

「PL/SQL」アイテム・タイプを単純にしたもの。含まれる属性が少ないため、すばやく簡単に作成できる。

「シンプル URL」アイテム・タイプ (simple URL item type)

「URL」アイテム・タイプを単純にしたもの。含まれる属性が少ないため、すばやく簡単に作成できる。

「シンプル・イメージ」アイテム・タイプ (simple image item type)

「イメージ」アイテム・タイプを単純にしたもの。含まれる属性が少ないため、すばやく簡単に作成できる。

「シンプル・テキスト」アイテム・タイプ (simple text item type)

「テキスト」アイテム・タイプを単純にしたもの。含まれる属性が少ないため、すばやく簡単に作成できる。

「シンプル・ファイル」アイテム・タイプ (simple file item type)

「ファイル」アイテム・タイプを単純にしたもの。含まれる属性が少ないため、すばやく簡単に作成できる。

「シンプル・ページ・リンク」アイテム・タイプ (simple page link item type)

「ページ・リンク」アイテム・タイプを単純にしたもの。含まれる属性が少ないため、すばやく簡単に作成できる。

スタイル (style)

ページおよびページ内の領域の色とフォントを制御する一連の値とパラメータ。スタイルの設定には、フォント・スタイル、サイズ、色、位置および背景色が含まれる。スタイルは、特定のページ・グループ用に作成することも、複数のページ・グループ内のページで使用される共有オブジェクトとして作成することもできる。

ストアド・プロシージャ (stored procedure)

データベース内に格納された一連の PL/SQL プロシージャ。

ストライプ (stripe)

特定のサブスクライバに割り当てられた仮想プライベート・ポータル上の保護スライス。

スナップショット (snapshot)

リモート・データベース内のマスター表と呼ばれる 1 つ以上の表に対する問合せの結果を含む表。

スナップショット・ログ (snapshot log)

スナップショットのマスター表と関連付けられた表であり、マスター表の変更を追跡する。

「スマート・テキスト」アイテム・タイプ (smart text item type)

「[Portal スマート・テキスト](#)」アイテム・タイプ (Portal smart text item type)」を参照。

「スマート・リンク」アイテム・タイプ (smart link item type)

「[Portal スマート・リンク](#)」アイテム・タイプ (Portal smart link item type)」を参照。

セッション (session)

「[ポータル・セッション](#) (portal session)」を参照。

全画面モード (Full Screen mode)

[ポートレット表示モード](#)のオプションの1つ。他のポートレットとページを共有しているときに、ポートレットにより多くのコンテンツが表示されるようにする。

属性 (attribute)

[アイテム](#)または[ページ](#)に関する情報 (メタデータ) を格納する。「作成日」、「有効期限日」、「管理者」などがある。[ページ・グループ管理者](#)は、[アイテム・タイプ](#)および[ページ・タイプ](#)の機能を拡張するために、カスタム属性を作成できる。たとえば、「表示名」はファイルの基本属性だが、そのファイルが機密であるかどうかを示すチェック・ボックスはカスタム属性である。カスタム属性は、検索可能な一意の識別子をアイテムに割り当てるために役立つ。

代替タグ (substitution tag)

タイトル、ヘッダーおよび他の要素をテンプレートに動的に埋め込むために、[非構成済 UI テンプレート](#)内で使用される特殊なポータル・タグ。

タイトル (title)

「[表示名](#) (display name)」を参照。

ダイレクト・アクセス URL (direct access URL)

オブジェクト・タイプと名前が含まれた [URL](#) を使用して、OracleAS Portal [オブジェクト](#) ([ページ](#)、[カテゴリ](#)、[パースペクティブ](#)、ドキュメントなど) にユーザーが直接アクセスしたり、ブックマークを付けたりできるようにする機能。ダイレクト・アクセス URL にはパラメータが含まれていないため、通常この URL は短く、次のように従来の URL 表記に近い。

<http://mymachine.mycompany.com:5000/pls/portal/url/page/toplevelsite>

「[動的 URL](#) (dynamic URL)」と対比。「[永続的なリンク](#) (durable link)」も参照。

妥当性チェックベースのキャッシュ (validation-based caching)

ポータル・キャッシュを使用して実行される**キャッシュ**方法。ポータル・キャッシュ内のアイテムが使用される前に、**PPE** または **mod_plsql** は、ポータルまたは**プロバイダ**にアクセスして、キャッシュされたアイテムがまだ有効かどうかを判断する。

「**有効期限ベースのキャッシュ (expiry-based caching)**」および「**失効化ベースのキャッシュ (invalidation-based caching)**」と対比。

タブ (tab)

ページで利用可能な実際の面積を効果的に2倍 (3倍、4倍など) にして、ページに表示できるコンテンツ量を増やすために使用する**ページ**上の領域。また、タブによって、対象となる分野、組織、特定のロールなどに共通するコンテンツをグループ化できる。

チェックアウト/チェックイン (check-out/check-in)

「**文書管理 (document control)**」を参照。

チャート (chart)

SQL **問合せ**の結果を、棒グラフ、円グラフまたは折れ線グラフとして表示する OracleAS Portal **ポートレット**。チャートは、表またはビューの2つ以上の列に基づいている。一方はチャートの各バーを表し、もう一方はチャートの各バーのサイズを計算する。

中間層 (middle tier)

OracleAS Portal アーキテクチャの一部。**HTTP** ユーザー・リクエストを適切な Portal データベースまたは**プロバイダ**に転送することによって処理し、Portal **ページ**を収集して Portal コンテンツの**キャッシュ**を管理する。

データ駆動のポートレット (data-driven portlet)

データ駆動の**ポートレット**を使用すると、Oracle データベースのオブジェクトに格納されたデータを表示、作成または更新できる。OracleAS Portal には、データ駆動のポートレットを作成するための作成ウィザードが用意されている。作成ウィザードは、データベースに格納される PL/SQL プロシージャを生成する。

データベース・アクセス記述子 (Database Access Descriptor)

「**DAD**」を参照。

データベース・オブジェクト (database object)

「**オブジェクト (object)**」を参照。

データベース管理者 (database administrator)

「**DBA**」を参照。

データベース・プロバイダ (database provider)

PL/SQL [ストアド・プロシージャ](#)として作成された[プロバイダ](#)のタイプ。データベースに常駐する[ポートレット](#)を作成するために使用される。データベース・プロバイダの例として、フォーム、レポートおよびチャートのポートレットを提供するために OracleAS Portal 内のウィザードを使用して構築されたプロバイダがある。

「[Web プロバイダ \(Web provider\)](#)」と対比。

データ・ポートレット (data portlet)

データをスプレッドシート形式で表示する OracleAS Portal [ポートレット](#)。

テーマ (theme)

[Oracle Text](#) によって生成される、ドキュメントを記述するスナップショット。特定の語句を含むドキュメントを検索するのではなく、対象がドキュメント内で明確に言及されていない場合でも、ユーザーは [Oracle Text](#) を使用して、一定の対象についてドキュメントを検索できる。

ディレクトリ情報ツリー (Directory Information Tree)

「[DIT](#)」を参照。

「テキスト」アイテム・タイプ (text item type)

[コンテンツ作成者](#)が[ページ](#)に追加できるデフォルトの[アイテム・タイプ](#)の1つ。テキスト・アイテムを作成するときには、アイテム・ウィザード内に最大 32KB のテキストを入力する。その後このテキスト・ブロックは、[Oracle Application Server Metadata Repository](#) の OracleAS Portal スキーマ内に格納される。

テスト (test)

特定の対象についてのユーザーの理解度を査定するために使用する一連の質問。質問の正解を提供し、各質問の配点を割り当てることができる。記述式の回答も手動で採点できる。

「[投票 \(poll\)](#)」および「[サーベイ \(survey\)](#)」と対比。

デフォルト以外のサブスクライバ (non-default subscriber)

ホスティングされた OracleAS Portal 上に [ASP](#) が提供する [ストライブ](#)がある [サブスクライバ](#)。

デフォルト・サブスクライバ (default subscriber)

OracleAS Portal のインストールに伴ってインストールされる基本 [サブスクライバ](#)。

デフォルト編集モード (Edit Defaults mode)

[ポートレット表示モード](#)のオプションの1つ。管理者がすべてのユーザーに対するポートレットのデフォルトを設定できるようにする。

「[編集モード \(Edit mode\)](#)」と対比。

テンプレート (template)

「[ページ・テンプレート \(page template\)](#)」または「[ユーザー・インタフェース \(UI\) テンプレート \(user interface \(UI\) template\)](#)」を参照。

問合せ (query)

データベース内の 1 つ以上の表またはビューから取得するデータを指定する SQL SELECT 文。

統合開発環境 (Integrated Development Environment)。

「[IDE](#)」を参照。

動的 URL (dynamic URL)

問合せ文字列 (1 つ以上のパラメータおよび ? や &) を含む URL。

「[ダイレクト・アクセス URL \(direct access URL\)](#)」と対比。

動的ページ (dynamic page)

ページに動的コンテンツを表示する OracleAS Portal [ポータルレット](#)。動的ページの作成ウィザードを使用すると、ページを作成するための HTML コード内に 1 つ以上の PL/SQL ブロックを指定できる。このコードはエンド・ユーザーがページをリクエストするときに実行される。

投票 (poll)

ユーザーから情報を引き出すために使用する一連の質問。

「[サーベイ \(survey\)](#)」および「[テスト \(test\)](#)」と対比。

トランスポート・セット (transport set)

[エクスポート](#)または[インポート](#)用の OracleAS Portal [オブジェクト](#)の集合。複数の[ページ・グループ](#)と複数の[ページ](#)など、特定タイプのオブジェクトを複数含めることができる。

トリガー (trigger)

表に関連付けられたデータベース・[オブジェクト](#)。1 つ以上の指定されたイベントの前または後に実行される。

内部イメージ名 (internal image name)

ポータルにアップロードされたイメージを識別するために使用する名前。アップロードされたイメージをポータル内で再使用するには、その内部名を参照する。

内部プロバイダ (internal provider)

[ページ・グループ・オブジェクト](#) (ページ、[ナビゲーション・ページ](#)など) を OracleAS Portal のインスタンス内で使用できるようにする [プロバイダ](#)のタイプ。

ナビゲーション・アイテム・タイプ (navigation item type)

ナビゲーションを提供し、ポータル固有のファンクションにアクセスまたはそれを実行するために使用する。

次の組込みナビゲーション・アイテム・タイプがある。

- 「スマート・リンク」アイテム・タイプ
- 「スマート・テキスト」アイテム・タイプ
- 「ログイン/ログアウト・リンク」アイテム・タイプ
- 「基本検索ボックス」アイテム・タイプ
- 「オブジェクトのリスト」アイテム・タイプ
- 「オブジェクト・マップ・リンク」アイテム・タイプ
- 「ページ・パス」アイテム・タイプ
- 「ページ・ファンクション」アイテム・タイプ

「アイテム・タイプ (item type)」も参照。「コンテンツ・アイテム・タイプ (content item type)」と対比。

ナビゲーション・ページ (navigation page)

ページ・グループ内の特殊な目的を持ったページ。通常は、ナビゲーション・バーやバナーなどの標準的なユーザー・インタフェース効果を実装するために、他のページまたはページ・テンプレートに埋め込まれる。ポータル内のナビゲーション用に通常ナビゲーション・アイテム・タイプが含まれる。

ナビゲータ (Navigator)

オブジェクトを検索し、OracleAS Portal と対話するための機能。ページ・グループ、プロバイダおよびデータベース・オブジェクトなど、ユーザーが権限を持っているオブジェクトへのアクセスを提供する。

認可 (authorization)

メッセージを送信したり、リクエストを行うためのセキュリティ制約の評価。認可では特定の基準（認証および制限）を使用して、リクエストを許可するかどうかを決定する。

認証されたユーザー (authenticated user)

OracleAS Portal にログインしたユーザー。デフォルトでは、認証されたユーザーは、ページなどの特定の OracleAS Portal オブジェクトにアクセスし、そのユーザーに付与された権限に基づいて操作を実行できる。

パブリック・コンテンツにのみアクセスできる「パブリック・ユーザー (public user)」と対比。

ページ (purge)

「システム・ページ (system purge)」を参照。

バージョンング (versioning)

Oracle Application Server Metadata Repository の OracleAS Portal スキーマ内に複数のバージョンの**アイテム**が同時に存在できるようにする機能。この機能は、ドキュメントについてバージョンの変更を追跡する場合、または必要に応じて前のバージョンに戻す場合に役立つ。

パースペクティブ (perspective)

複数のカテゴリにまたがる、**アイテム**のグループ化。パースペクティブを使用すると、特定のアイテムに関心のある人々が誰かという観点から、コンテンツを分類できる。たとえば、世界中の様々な観光地へのリンクを追加し、休暇の種類についてのアイテムに「北欧ファンのための休暇」、「古代ロマンの旅」、「冒険者向けの極限の旅」などのパースペクティブを割り当てることができる。パースペクティブ属性を含むアイテム・タイプを使用してコンテンツを公開するユーザーは、パースペクティブに値を指定しなくても、1つまたは複数の値を指定してもかまわない。

「**カテゴリ (category)**」と対比。

パートナー・アプリケーション (partner application)

認証を **Oracle Application Server Single Sign-On Server** に委譲したアプリケーション。SSO Server に登録すると、1つのログイン・ページを使用して、複数の**パートナー・アプリケーション**にログインできる。特定のセッションで、ユーザーが SSO Server によって認証された後は、別のパートナー・アプリケーションにアクセスするために再びログインする必要はない。

バインド変数 (bind variable)

SQL 文の変数。SQL 文が正しく実行されるためには、有効な値または値のアドレスに置き換えられる必要がある。通常、ポートレット開発者は、バインド変数 (:dept など) を使用して、OracleAS Portal **ポートレット**の**カスタマイズ・フォーム**内に**パラメータ入力フィールド**を表示する。入力フィールドで、エンド・ユーザーはポートレットに表示されるデータを選択できる。

パスの別名 (path aliasing)

「**ダイレクト・アクセス URL (direct access URL)**」を参照。

パッケージ (package)

PL/SQL 仕様と本体で構成されるデータベース・**オブジェクト**。仕様には、他のプログラム・ユニットから参照できるデータ型とサブプログラムが含まれている。本体には、パッケージの実際の実装が含まれている。

バッチ・ジョブ (batch job)

OracleAS Portal のバッチ・ジョブ機能を使用して、OracleAS Portal の **ポートレット** をバックグラウンドで実行すること。エンド・ユーザーは、ポートレットの **カスタマイズ・フォーム** にあるオプションを選択すると、ポートレットをバッチ・モードで実行できる。ポートレットが大量のデータに基づいている場合、ポートレットにデータの行が多数表示される場合、またはジョブの実行に長時間かかる可能性がある場合は、バッチ処理が役立つ。

バナー (banner)

「**領域バナー (region banner)**」を参照。「**ナビゲーション・ページ (navigation page)**」も参照。

パブリック・ページ (public page)

パブリック・ユーザー (OracleAS Portal にログインしないユーザー) が表示できる、**ページ・グループ**内の**ページ**。ページ管理者または**ページ・グループ管理者**は、ページをパブリックとして明示的に指定する必要がある。

パブリック・ユーザー (public user)

OracleAS Portal にアクセスできるが、ログインしないユーザー。ユーザーが初めて OracleAS Portal にアクセスするときは、ログインできるかどうかに関係なく、パブリック・ユーザーとしてアクセスする。パブリック・ユーザーは、パブリックとして指定されたすべての**ページ**を表示できるが、そのコンテンツをカスタマイズまたは編集したり、なんらかのアクセス制御があるページを表示することはできない。

「**認証されたユーザー (authenticated user)**」と対比。

パラメータ (parameter)

ページと**ポートレット**の間、またはポートレット間で渡される値。

ページ・パラメータは、ページ設計者によって作成されるページ・レベルのパラメータであり、その値はポートレット・パラメータにマップできる。

ポートレット・パラメータは、プロバイダによって宣言される。ページ設計者は、ページ・パラメータをポートレット・パラメータにマップする。**PPE** がプロバイダからのポートレットをリクエストすると、ポートレットで宣言され、ページ・パラメータにマップされたポートレット・パラメータのみが送信される。

パラメータ入力フィールド (parameter entry field)

OracleAS Portal **ポートレット**に渡される値をエンド・ユーザーが入力できる、**カスタマイズ・フォーム**上のフィールド。

非構成済 UI テンプレート (unstructured UI template)

特定のページにコンテンツと HTML コードを挿入するため、または OracleAS Portal ポートレットの外観を制御するために使用する共有ポートレット。HTML コードに基づいていて、実行時に、タイトル、ヘッダーおよびページを構成する他の要素を動的に埋め込む。

「代替タグ (substitution tag)」および「ユーザー・インタフェース (UI) テンプレート (user interface (UI) template)」も参照。「構成済 UI テンプレート (structured UI template)」と対比。

非武装地帯 (de-militarized zone)

「DMZ」を参照。

ビュー (view)

仮想的な表であり、その行はデータベース内に実際には存在しないが、データベース内に物理的に格納された表に基づいている。

表 (table)

リレーショナル・データベース内の基本的な記憶域構造。

表示名 (display name)

OracleAS Portal 全体で使用されるオブジェクトの外部名。ナビゲータ内、ページ上、ページ・エディタ内などで使用される。オブジェクトがポートレットとして公開されると、表示名はポートレット・リポジトリ内のポートレットのタイトルとして使用される。

表示モード (show mode)

情報を表示するためにポートレットをコールできる方法。次のような方法がある。

- 共有画面モード
- 編集モード
- デフォルト編集モード
- プレビュー・モード
- ヘルプ・モード
- 情報モード
- リンク・モード
- 全画面モード

表示モード (View mode)

ページの実行時の表示。

「編集モード (Edit mode)」と対比。

標準ページ (standard page)

アイテムおよびポートレットを含めて管理するために使用するページ・タイプ。

表領域 (tablespace)

データベース内の領域の割当て。

「ビルダー」ページ (Builder page)

「Portal ビルダー」ページ (Portal Builder page)」を参照。

ファイアウォール (firewall)

一連のコンピュータやネットワークを外部の攻撃から保護するための媒介として機能するマシン。外部から LAN 上のコンピュータへのアクセスを規制し、LAN 内から外部コンピュータへのアクセスを規制する。ファイアウォールの働きには、リクエストを転送してあたかもファイアウォール・マシンから出されたかのように見せるプロキシ・サーバーとして機能する方法、あるいはリクエストを検査して疑わしいものを排除する方法がある。

「ファイル」アイテム・タイプ (file item type)

コンテンツ作成者がページに追加できるデフォルトのアイテム・タイプの1つ。ユーザーがファイル・アイテムをページに追加すると、ファイルが Oracle Application Server Metadata Repository の OracleAS Portal スキーマにアップロードされ、ハイパーリンクとしてページに表示される。ユーザーが表示名リンクをクリックすると、ファイル・タイプとブラウザの構成に応じて、そのファイルがユーザーのコンピュータにダウンロードされるか、またはユーザーの Web ブラウザに表示される。

ファンクション (function)

指定された一連の操作を実行して値を戻す PL/SQL サブプログラム。通常、ファンクションは、大規模なアプリケーションの範囲内で特定のタスクを実行するために作成された、非常に限定された小さいコードのブロック。

エンド・ユーザーは、ページ内で PL/SQL アイテムまたはカスタム・アイテムのタイトルをクリックしてファンクションを実行する。

フォーム (form)

1つ以上のデータベース表、ビューまたはプロシージャへのトランザクション・インタフェースを提供する OracleAS Portal ポートレット。たとえば、OracleAS Portal 内のツールを使用して、人事データベースに新入社員情報を入力するためのフォームを作成できる。

「マスター・ディテール・フォーム (master-detail form)」も参照。

フォルダ (folder)

廃止された用語。「ページ (page)」を参照。

ブックマーク (bookmark)

「[お気に入り \(favorite\)](#)」を参照。

フレーム・ドライバ (frame driver)

2つのフレームに分割された Web ページで構成される OracleAS Portal [ポートレット](#)。ドライビング・フレームには、2番目のターゲット・フレームのコンテンツを決定する SQL の[問合せ](#)が含まれる。

ブレッダクラム (breadcrumbs)

「[ページ・パス](#)」[アイテム・タイプ \(page path item type\)](#)」を参照。

プレビュー・モード (Preview mode)

[ポートレット表示モード](#)のオプションの1つ。ユーザーがポートレットをページに追加する前にプレビューできるようにする。

プロキシ・サーバー (proxy server)

プロキシ・サーバーは、一般にネットワーク・[ファイアウォール](#)上に位置し、ファイアウォール内部のクライアントが Web リソースにアクセスするのを許可する。クライアントからのすべてのリクエストは、接続先サーバーに直接送られるのではなく、プロキシ・サーバーに送られる。プロキシ・サーバーは接続先サーバーにリクエストを転送し、受信した情報をクライアントに返す。プロキシ・サーバーはサイトでのすべての Web トラフィックを、単一のセキュアなポートを通して送る。これにより、組織は Web アクセスを許可する一方で内部マシンへのインターネット・アクセスを防ぎ、セキュアなファイアウォールを作成できる。

プロバイダ (provider)

OracleAS Portal と [ポートレット](#)間の通信リンク。プロバイダには、[Web プロバイダ](#)と [データベース・プロバイダ](#)の2種類がある。Web プロバイダは、ネットワーク上のどこにあってよく、[SOAP](#)を使用して処理される。Web プロバイダは、任意の Web テクノロジーを使用して実装できる。[PDK-Java](#) と [PDK-URL Services](#)を使用して、独自の Web プロバイダを構築できる。データベース・プロバイダは Oracle データベース内にあり、データ集約型の操作の実行中にポートレットを管理する。

プロバイダはポートレットのコンテナとして機能し、各ポートレットはプロバイダを通じて OracleAS Portal と通信する。プロバイダは、それ自体に含まれているポートレットも管理する。

プロバイダ・グループ (provider group)

プロバイダ・グループ・サービスによって定義された、[Web プロバイダ](#)の論理的な集合。[ポータル管理者](#)は、ポータルで使用するためにプロバイダ・グループを登録できる。登録すると、グループ内の各 [プロバイダ](#)を登録する処理が容易になる。この結果、Web プロバイダを作成する組織はプロバイダの登録の詳細を公開できるようになり、任意の OracleAS

Portal インスタンスへの自動登録が容易になる。ポータル管理者に提供する必要がある情報は、プロバイダ・グループの名前と位置のみである。

「[ローカル・プロバイダ・グループ \(local provider group\)](#)」および「[リモート・プロバイダ・グループ \(remote provider group\)](#)」も参照。

プロバイダ定義 (provider definition)

宣言の XML ベースの構成ファイル (provider.xml)。Web プロバイダ、そのポートレット、およびそのポートレット内に表示されるコンテンツの位置を記述する。この構成ファイルは、プロバイダとそのポートレットの動作も記述する。

プロバイダ・レコード (provider record)

ポートレットについて指定された情報が含まれた、データベース・プロバイダから戻されたレコード。

プロパティ・シート (property sheet)

アイテムの属性またはページのプロパティのサマリーが表示される組込み属性。

プロフィール (profile)

パスワード、ユーザー ID、権限など、OracleAS Portal ユーザーまたはグループに関する情報。

文書管理 (document control)

ユーザーがアイテムをチェックアウトして他のユーザーがそのアイテムを編集できないようにすることが可能。これにより、複数のユーザーが互いの変更を上書きするのを防ぐ。ユーザーがアイテムの編集を終了して、そのアイテムをチェックインして戻すと、再び他のユーザーがそのアイテムを編集できるようになる。

分類 (classification)

ユーザーが検索中にコンテンツを簡単に発見できるように、カテゴリおよびパースペクティブを使用してページのコンテンツが分類される。

「[カテゴリ \(category\)](#)」および「[パースペクティブ \(perspective\)](#)」も参照。

ページ (page)

ポートレットとアイテムが含まれている OracleAS Portal オブジェクト。ページは、表示されるたびに動的に収集され、そのページのポートレットおよび選択されたレイアウトに従って書式設定される。

「[ページ・タイプ \(page type\)](#)」も参照。

ページ・イベント (page event)

ポータル開発者によって定義されたユーザーの操作。ユーザーの操作には、Web ページ上のリンク、ボタン、その他のコントロールのクリックが含まれる。ページ設計者は、現在のページの再ロードまたは別のページのロードを実行し、新しくロードされたページにパラメータを渡すイベントを指定できる。

ページ管理者 (page manager)

「ページ設計者 (page designer)」を参照。

ページ・グループ (page group)

ページ、スタイル、ナビゲーション・ページおよびパースペクティブなどの関連するポータル・オブジェクトをグループ化してプロパティを設定する OracleAS Portal オブジェクト。ページ・グループには通常、コンテンツを編成するためのページとサブページの階層が含まれている。

ページ・グループ管理者 (page group administrator)

ページ・グループ全体に対するすべての権限を持つユーザー。ページ・グループを設定して管理し、ページ所有者を指定して分類を作成する。ページ・グループ内のすべてのページを表示して管理することもできる。

ページ・グループ・マップ (page group map)

ポータル内のすべてのページ・グループの階層的な編成を表示し、ユーザーがページ・グループ内の個別のページにアクセスできるようにする。ページ・グループ・マップは各ユーザーに対して個別に作成され、そのユーザーが表示したり編集することが承認されているページのみが表示される。

ページ・グループ割当て (page group quota)

「割当て制限 (quota)」を参照。

ページ設計者 (page designer)

ページに対する「管理」権限を持ったユーザー (ページ管理者とも呼ばれる)。この権限を持ったユーザーは、ページに対するすべての処理を実行し、そのページの下にサブページを作成できる。ページ設計者は、ページ・レイアウト (領域の構成) の設計、および他のユーザーへのページ権限の割当て (どのユーザーがページにコンテンツを追加できるかの決定など) を行うことがある。

ページがテンプレートに基づいている場合は、そのページに対してページ設計者が制御できる範囲が制限されることがある。

ページ・タイプ (page type)

ページのコンテンツとページについて格納される情報を定義する。OracleAS Portal に含まれる基となるページ・タイプには、**標準ページ**、**モバイル・ページ**、**PL/SQL ページ・タイプ**、**JSP** および **URL ページ**がある。カスタム・ページ・タイプは、基となるページ・タイプに用意されている機能を拡張し、ページに関する追加情報を格納するために、**ページ・グループ管理者**によって作成されるページ・タイプ。

ページ・ツールバー (page toolbar)

ページの**編集モード**で、ページの最上部に表示されるリンク。これを使用して、ページの様々な側面の編集、編集ビューの切り替え (**グラフィカル・ビュー**から**レイアウト・ビュー**になど)、**ページ・グループ**のプロパティの編集などを行うことができる。

ページ・テンプレート (page template)

テンプレートを使用する各**ページ**に対して共通の外観と共通のコンテンツを設定する OracleAS Portal **オブジェクト**。ページを作成するユーザーは、ページのレイアウトを定義してデフォルトのコンテンツを追加するために、ページ・テンプレートを選択できる。ページ・テンプレートにはページのすべての機能が含まれているため、これにより**アイテム**、**ナビゲーション・ページ**および**ポートレット**を含めることができる。

「**ユーザー・インタフェース (UI) テンプレート (user interface (UI) template)**」と対比。

ページの部分的なキャッシュ (partial page caching)

「**PPC**」を参照。

「ページ・パス」アイテム・タイプ (page path item type)

ユーザーが**ページ**に追加できる**ナビゲーション・アイテム・タイプ**。ページ・パスは、ページの参照名の連鎖。ページ・パス (ブレッドクラムとも呼ばれる) によって、完全なディレクトリ・パスが記述される。

「ページ・ファンクション」アイテム・タイプ (page function item type)

ユーザーが**ページ**に追加できる**ナビゲーション・アイテム・タイプ**。ページ・ファンクションは、ユーザーがカスタム・**ページ・タイプ**に追加できるプロシージャ・コール。現在のページに関連付けられたページ・ファンクションがない場合、このアイテム・タイプは表示されない。

ページ・メタデータ (page metadata)

ページに関して格納される情報、つまり属性。そのレイアウトとキャッシュを設定するために OracleAS Portal によって使用される。

「ページ・リンク」アイテム・タイプ (page link item type)

コンテンツ作成者がページに追加できるデフォルトのアイテム・タイプの1つ。ページ・リンクによって、ポータル内の別のページへのハイパーリンク経由のルートが提供される。ユーザーが表示名リンクをクリックすると、このアイテムが参照しているページがユーザーのブラウザに表示される。

ヘルプ・モード (Help mode)

ポートレット表示モードのオプションの1つ。ポートレットの機能の使用方法に関する情報を表示する。

編集モード (Edit mode)

ページの編集の場合：編集モードでは、適切な権限を持つ認証されたユーザーが、ページ・プロパティの設定、そのページに対するポートレットおよびアイテムの追加、変更または削除を実行できる。編集モードに切り替えるには、ページ上の「編集」リンクをクリックする。編集モードのビューには、グラフィカル・ビュー、レイアウト・ビューおよびリスト・ビューがある。

「モバイル・プレビュー・モード (Mobile Preview mode)」および「保留中のアイテム・プレビュー・モード (Pending Items Preview mode)」も参照。

ポートレットの場合：ユーザーごと、インスタンスごとにポートレットのパーソナライズを可能にする、オプションのポートレット表示モード。

「デフォルト編集モード (Edit Defaults mode)」と対比。

ポータル (portal)

Web ベース・アプリケーションとの対話および個別のユーザーまたはユーザーのクラスに関連する情報のパーソナライズされた単一のポイントを提供する共通インタフェース（つまり、Web ページ）。OracleAS Portal を使用して構築されたポータルは、ページ・グループ内で管理されるページで構成され、ポートレットとアイテムを含んでいる。

ポータル管理者 (portal administrator)

OracleAS Portal 内で最高レベルの権限を持ったユーザー。ポータル管理者は、プライベートとマークされたページおよびデータベース・プロバイダも含め、OracleAS Portal 内のすべてのオブジェクトを表示および変更できる。（唯一の例外はグループ。ポータル管理者は PORTAL_ADMINISTRATORS および PORTAL_PUBLISHERS グループを変更できるが、グループの所有者でないかぎり、他のグループは変更できない。）

ポータル・キャッシュ (portal cache)

ポータル・キャッシュには、妥当性チェックベースのキャッシュを使用するオブジェクトのキャッシュ・エントリが格納される。さらに、オブジェクトが妥当性チェックベースのキャッシュと失効化ベースのキャッシュの両方を使用している場合は、メモリーベースの Oracle Application Server Web Cache に対するバックアップとしても機能する。

ポータル・セッション (portal session)

ブラウザと OracleAS Portal 間の対話の期間。最初のアクセスから、ログオフ、ブラウザ・ウィンドウの終了、または非アクティブ期間経過後のセッションの期限切れまでの間。

ポータル・ページ (portal page)

「[ページ \(page\)](#)」を参照。

ポータル・リポジトリ (portal repository)

「[Oracle Application Server Metadata Repository](#)」を参照。

ポートレット (portlet)

通常は Web コンテンツの一部を表示する、再使用可能なプラグgable Web コンポーネント。ポートレットは OracleAS Portal で作成する [ページ](#) の基本となる要素である。ウィザード・ベースのポートレット・ビルダーを使用すると、独自の [データ駆動のポートレット](#) を簡単に作成できる。OracleAS Portal には、プログラムを使用してポートレットを作成する方法と、あらゆる種類の Web コンテンツを統合する方法がいくつか用意されている。ポートレットは、Java、[JSP](#)、Java [サーブレット](#)、PL/SQL、Perl、ASP などの様々なテクノロジーを使用して実装される。[PDK](#) には、OracleAS Portal で提供される標準準拠のポートレット開発オプションがすべて含まれている。

ポートレット公開者 (portlet publisher)

OracleAS Portal [オブジェクト](#) ([ページ](#) またはそれ以外) を、[ページ](#) に含めることができるように [ポートレット](#) として公開できるユーザー。

ポートレット・プロバイダ (portlet provider)

「[プロバイダ \(provider\)](#)」を参照。

ポートレット・リポジトリ (Portlet Repository)

ローカル・[プロバイダ](#) および登録されたすべてのリモート・プロバイダが利用できる [ポートレット](#) が含まれた特別な [ページ・グループ](#)。プロバイダを登録すると、プロバイダとそのポートレットがポートレット・リポジトリに追加される。

ポートレット・レコード (portlet record)

実装スタイルや [表示モード](#) (PL/SQL) など、[ポートレット](#) に関する詳細情報が含まれたプログラムの構造。

ホーム・ページ (home page)

OracleAS Portal 内で定義される [ページ](#) で、通常はユーザーがログインしたとき、またはホーム・「[スマート・リンク](#)」[アイテム・タイプ](#) を選択したときに表示される。[ポータル管理者](#) は、このページを [パブリック・ユーザー](#) のために選択する。[認証されたユーザー](#) は自分自身のために選択できる。ポータル管理者が [モバイル・ページ](#) 設計を有効にすると、ポータルがモバイル・デバイスからアクセスされたときに表示される別のモバイル・ホーム・ページを指定できる。

ホスティング・サイト (hosted site)

「[ストライプ \(stripe\)](#)」を参照。

保存された検索 (saved search)

保存された検索では、すべての検索基準を1つの名前で保存できる。この機能により、検索基準を手動で再入力するかわりに、保存した検索名を選択することによって、検索をすばやく繰り返すことができる。[基本検索](#)、[拡張検索](#)または[カスタム検索](#)の結果を保存できる。保存された検索ポートレットには、ページ・グループ内のすべての保存された検索のリストが表示される。

保留中のアイテム・プレビュー・モード (Pending Items Preview mode)

承認待ちのアイテムを表示できるプレビュー・モード。このモードを使用すると、コンテンツ作成者は追加したがまだ承認されていないアイテムをプレビューでき、承認者はアイテムを承認または却下する前にそれをプレビューできる。

「[編集モード \(Edit mode\)](#)」も参照。

保留中の承認の監視 (Pending Approvals Monitor)

自分が管理しているページ・グループ内の保留中の承認をリストできるようにするポートレット。承認者別、日付別、ページ・グループ別または送信者別に、保留中の承認をリストできる。

翻訳 (translation)

別の言語でレンダリングされた[ページ・グループ](#)。[ページ・グループ管理者](#)が翻訳を作成すると、[コンテンツ作成者](#)はその言語でコンテンツを追加できる。ページ・グループ・ユーザーも、ブラウザ言語をサポートされる言語の1つに設定することで、翻訳されたコンテンツを表示できる。

マスター・ディテール・フォーム (master-detail form)

単一の HTML ページ内に、マスター表の行と複数のディテール行を表示する OracleAS Portal [ポートレット](#)。マスター行の値によって、問合せ、更新、挿入および削除用に表示されるディテール行が決まる。

「[フォーム \(form\)](#)」も参照。

マニフェスト (manifest)

[トランスポート・セット](#)およびその依存物の中にあるオブジェクトのリスト。また、インポート・モードの制御に関する精度レベルを提供する。

明示オブジェクト (explicit object)

エクスポート・オブジェクトのカテゴリ。ナビゲータまたは一括操作からエクスポートの対象として明示的に選択されたオブジェクト。

「[マニフェスト \(manifest\)](#)」および「[参照しているオブジェクト \(referenced object\)](#)」も参照。

メニュー (menu)

エンド・ユーザーがクリックして他のメニュー、他の OracleAS Portal ポートレット、または URL に移動できるオプションが含まれた Web ページを表示する OracleAS Portal **ポートレット**。

基となるページ・タイプ (base page type)

「**ページ・タイプ (page type)**」を参照。

モバイル XML (mobile XML)

「**Oracle Application Server Wireless XML**」を参照。

モバイル・プレビュー・モード (Mobile Preview mode)

モバイル・デバイス上でページがどのように表示されるかをプレビューできるようにするプレビュー・モード。

モバイル・ページ (mobile page)

ページ作成者が携帯電話などのモバイル・デバイス専用の**ページ**を作成できるようにする**ページ・タイプ**。

ユーザー / ロール (grantee)

別のユーザーによって**オブジェクト**に対する権限を付与されるユーザー。

ユーザー・インタフェース (UI) テンプレート (user interface (UI) template)

フル・ページ表示モードの OracleAS Portal **ポートレット**の外観を制御する**共有ポートレット**。ポートレットの構築中に UI テンプレートを選択すると、そのポートレットが表示されるページのタイトル、タイトルの背景、別の Web ページへのリンク、および背景色とイメージが自動的に選択される。

「**構成済 UI テンプレート (structured UI template)**」および「**非構成済 UI テンプレート (unstructured UI template)**」も参照。「**ページ・テンプレート (page template)**」と対比。

有効期限 (expiration period)

アイテムの期限が切れるまでの日数、または期限が切れる正確な日付。アイテムの期限が切れた後は、そのアイテムまたは**ページ**の所有者と**ページ・グループ管理者**のみが**編集モード**でそのアイテムを表示できる。期限切れアイテムは、すべての期限切れアイテムの**システム・ページ**時にデータベースから削除される。

有効期限ベースのキャッシュ (expiry-based caching)

保持期間を使用して、アイテムがキャッシュ内で有効である期間を指定する**キャッシュ**方法。この期間が経過すると、更新が必要になる。有効期限ベースのキャッシュを使用するページは、ユーザーのブラウザでもキャッシュされる場合がある。

「**失効化ベースのキャッシュ (invalidation-based caching)**」および「**妥当性チェックベースのキャッシュ (validation-based caching)**」も参照。

要旨 (gist)

題材全体を最もよく表現するいくつかの段落からなる、**Oracle Text** のサマリー。このようなサマリーを使用すると、テキストの主要な内容に目を通したり、テキストの題材で興味のあるものにアクセスできる。

ライブラリ (library)

1 つ以上の PL/SQL または Java プログラム・ユニットの集合。ライブラリは、複数のアプリケーションで同時に参照できる。

リスト・ビュー (list view)

すべての**ページ**・コンテンツのリストが表示され、複数の**オブジェクト**に対して操作（削除、移動、コピーなど）を実行できるオプションが用意されたページ編集ビュー。

「**グラフィカル・ビュー (graphical view)**」および「**レイアウト・ビュー (layout view)**」と対比。

リッチ・テキスト・エディタ (rich text editor)

コンテンツ作成者がテキスト・アイテムに簡単に書式を設定できるようにする WYSIWIG エディタ。リッチ・テキスト・エディタは、Internet Explorer 内でのみ使用できる。

リモート・データベース (remote database)

接続文字列またはデータベース・リンクを通じてネットワークを介してアクセスできる別個のマシンで実行されるデータベース。

リモート・プロバイダ・グループ (remote provider group)

OracleAS Portal のローカル・インスタンス外で定義された**プロバイダ**の集合。

「**プロバイダ・グループ (provider group)**」を参照。「**ローカル・プロバイダ・グループ (local provider group)**」と対比。

領域 (region)

標準ページ上の切り分けられた区域。ページ・レイアウトを定義し、ページ・コンテンツ（**ポートレット**と**アイテム**）を定義し、領域内に表示されるコンテンツの**スタイル**と**属性**を制御するために使用する。標準ページには、1 つ以上の領域を含めることができる。領域は、別の領域の上、下または横に作成できる。

次のタイプの領域を作成できる。

- 未定義領域。特定のタイプが割り当てられていない領域。
- アイテム・タイプ領域。テキスト、イメージ、ファイルなどのアイテムを追加できる領域。
- ポートレット・タイプ領域。ポートレットを含めることができる領域。
- サブページ・リンク領域。現在のページのサブページのリストを表示できる領域。
- タブ・タイプ領域。**タブ**を含めることができる領域。

領域バナー (region banner)

OracleAS Portal ページの領域に表示される、タイトルが付いた色付きの水平バー。バナーは、ページの視覚的な流れを分割し、その下に表示される関連アイテムをグループ化する。

リンク・モード (Link mode)

ポートレット表示モードのオプションの1つ。携帯電話などのモバイル・デバイス上でポートレットがレンダリングされるようにする。

ルーティング方法 (routing method)

OracleAS Portal には、次の3つの承認ルーティング方法が用意されている。

- 全員、並行。この方法を選択すると、OracleAS Portal によって、手順内の受信者全員に一度に承認が送信される。アイテム承認が次の手順に移動する前に、すべての受信者が承認に対して応答する必要がある。
- 全員、順次。この方法を選択すると、OracleAS Portal によって、手順内の受信者に、指定された順序で一度に1人ずつ承認が送信される。アイテム承認が次の手順に移動する前に、すべての受信者が承認に対して応答する必要がある。
- 1人、並行。この方法を選択すると、OracleAS Portal によって、手順内の受信者全員に一度に承認が送信される。ただし、アイテム承認が次の手順に移動する前に、受信者の1人のみが承認に対して応答する必要がある。

「承認プロセス (approval process)」も参照。

ルート・ページ (root page)

ページ・グループ内のページ階層のトップ・レベル。ページ・グループ内の他のすべてのサブページが含まれる。ページ・グループのホーム・ページとも呼ばれる。

レイアウト・ビュー (layout view)

ページ上の領域を追加、配置および削除できるようにするページ編集ビュー。さらに、このビュー内でコンテンツの非表示、表示、削除または移動を行うことができる。

「グラフィカル・ビュー (graphical view)」および「リスト・ビュー (list view)」と対比。

レベル (level)

モバイル・ページに構造を提供するため、およびモバイル・デバイスの小型画面に表示されるコンテンツの量を制限するための方法として使用される。ユーザーは、モバイル・ページのレベルをドリルダウンして、詳細なコンテンツを表示する。

レポート (report)

SQL の問合せの結果を表形式で表示する OracleAS Portal ポートレット。

連携された Portal アダプタ (Federated Portal Adapter)

「FPA」を参照。

ローカル・プロバイダ・グループ (local provider group)

OracleAS Portal のインスタンス内で定義された**プロバイダ**の集合。プロバイダ・グループを使用すると、OracleAS Portal のあるインスタンス内に定義または登録されたプロバイダを別の OracleAS Portal インスタンスと簡単に共有できるようになる。

「**プロバイダ・グループ (provider group)**」も参照。「**リモート・プロバイダ・グループ (remote provider group)**」と対比。

ロード・バランス・ルーター (load-balancing router)

「**LBR**」を参照。

「ログイン/ログアウト・リンク」アイテム・タイプ (login/logout link item type)

他のユーザーがポータルにログインまたはポータルからログアウトできるように、**ページ**上にユーザーが配置できる**ナビゲーション・アイテム・タイプ**。

ロック (lock)

編集中に OracleAS Portal **ポートレット**に自動的に適用される設定。この設定によって、他のユーザーはそのポートレットを編集できなくなる。

WebDAV では、他のユーザーがファイルを編集できないようにする操作。WebDAV 内でファイルをロックすると、ポータル内の対応する**アイテム**がクライアントによってチェックアウトされる。

ワイヤレス・ポータル (wireless portal)

携帯電話などのワイヤレス・デバイスからアクセス可能な**ポータル**。

「**Oracle Application Server Wireless**」も参照。

割当て制限 (quota)

アップロードされたドキュメントを格納するために、ページ・グループ内または **Oracle Application Server Metadata Repository** の OracleAS Portal スキーマ内に用意される容量。

索引

A

ACL, 5-78, 5-82
 権限, 5-82
 プロバイダの権限, 5-83
 ポートレットの権限, 5-83
ALIGN_LEFT 代替タグ, A-136
ALIGN_RIGHT 代替タグ, A-136
Apache Struts
 概要, 5-112
 ポートレットの作成, 5-115
API, 1-15

B

BLOB, A-26, A-63
BODY 代替タグ, A-136

C

Cookie, A-119
 Web クリップング, 4-20
CSS
 共有画面モードでのガイドライン, 5-4

D

DBA グループ, A-126
DIRECTION 代替タグ, A-136
doDelete, A-118
doInsert, A-118
doUpdate, A-118

H

HELPLINK 代替タグ, A-136
HELPSCRIPT 代替タグ, A-136
HTML
 共有画面モードでのガイドライン, 5-3
httpd.conf, 5-85
HTTPS, 5-87
 構成, 5-88
HTTP エラー・コード 407
 Web クリップング, 4-21

I

IFrame
 Web クリップング, 4-22
IMAGE_PREFIX 代替タグ, A-136
init.ora パラメータ, A-109
interMedia リッチ・コンテンツ, A-26, A-62, A-63

J

J2EE アプリケーション・サーバー, 2-10
Java
 ポートレットのガイドライン, 5-2
Java Community Process, 2-6
JavaScript
 Web クリップング, 4-20
 共有コンポーネントの下に作成, A-129
 妥当性チェックに関するガイドライン, A-128
 フォーム・ポートレットへの追加, A-129
JavaScript イベント・ハンドラ, A-115
 記述, A-115
 ボタン, A-27

例, A-116
列, A-30
Java ポートレット
イベントのサポート, 2-28
開発ツール, 2-18
外部アプリケーションに対する認証, 2-31
キャッシュ・スタイル, 2-17
コンテンツのインラインでのレンダリング, 2-24
コンテンツの取得, 2-22
セキュリティ, 2-29
多言語サポート, 2-29
チャート作成, 2-25
パラメータのサポート, 2-27
必要な専門知識, 2-9
ページ区切りサポート, 2-30
他のポートレット・ビルダーとの比較, 2-2
ユーザー・インタフェース, 2-21
用途適性, 2-6
Java ポートレット仕様
「JPS」を参照
JPS, 5-11
Oracle JDeveloper のウィザード, 5-17
ポートレットの作成, 5-15
ポートレットのデプロイ, 5-26
JSR-168, 2-2

L

LDAP
「Oracle Internet Directory」を参照

M

MLS
「多言語ポートレット」を参照

N

NLS
「多言語ポートレット」を参照

O

OmniPortlet, 1-13
「イベント」タブ, 3-24
イベントのサポート, 2-28
開発ツール, 2-18

外部アプリケーションに対する認証, 2-30
概要, 3-3
箇条書きレイアウト, 3-21
キャッシュ・スタイル, 2-17
コンテンツのインラインでのレンダリング, 2-23
コンテンツの取得, 2-22
セキュリティ, 2-28
「ソース」タブ, 3-5
「タイプ」タブ, 3-5
多言語サポート, 2-29
チャート作成, 2-25
チャート・レイアウト, 3-16
データソースの使用, 3-5
データのフィルタリング, 3-12
ニュース・レイアウト, 3-19
パラメータのサポート, 2-27
必要な専門知識, 2-8
「表示」タブ, 3-13
表レイアウト, 3-15
「フィルタ」タブ, 3-12
フォーム・レイアウト, 3-22
ページ区切りサポート, 2-30
ポータル・ページへの追加, 3-27
他のポートレット・ビルダーとの比較, 2-2
ユーザー・インタフェース, 2-21
用途適性, 2-6
「レイアウト」タブ, 3-14
レイアウトの変更, 3-41
Oracle Internet Directory, 5-88
Oracle JDeveloper
JPS ポートレットのデプロイ, 5-26
PDK-Java ポートレットのデプロイ, 5-46
WAR ファイルのデプロイ, 5-28, 5-48
アプリケーション・サーバー接続, 5-26, 5-47
ポータル・アドイン, 5-14
Oracle Struts ポートレット, 5-113
OracleAS Portal
Oracle JDeveloper 用のアドイン, 5-14
データソースの使用, 3-5
リポジトリ, 2-14
OracleAS Web Cache
失効化ベースのキャッシュ, 2-17
失効ポート, 5-98, 5-100
ORDAUDIO, A-62
ORDIMAGE, A-62
ORDVIDEO, A-62

owa_cookie パッケージ
サブプログラム, A-119
データ型, A-119

P

p_session, A-118
p_value, A-118
PAGE.BASE.URL 代替タグ, A-137
PAGE.BASE 代替タグ, A-137
PAGE.BGCOLOR 代替タグ, A-137
PAGE.BGIMAGE 代替タグ, A-137
PAGE.CUSTOMIZEPAGE.LABEL 代替タグ, A-138
PAGE.CUSTOMIZEPAGE.URL 代替タグ, A-138
PAGE.CUSTOMIZEPAGE 代替タグ, A-138
PAGE.EDITPAGE.LABEL 代替タグ, A-138
PAGE.EDITPAGE.URL 代替タグ, A-138
PAGE.EDITPAGE 代替タグ, A-138
PAGE.REFRESH.LABEL 代替タグ, A-138
PAGE.REFRESH.URL 代替タグ, A-138
PAGE.REFRESH 代替タグ, A-138
PAGE.STYLE.URL 代替タグ, A-137
PAGE.STYLE 代替タグ, A-137
PAGE.SUBPAGELINK 代替タグ, A-137
Parallel Page Engine (PPE), 2-14
PDK, 1-15
PDK-Java
 ポートレット・ウィザード, 5-36
 ポートレットとプロバイダのテスト, 5-45
 ポートレットのカスタマイズ, 5-59
 ポートレットのデプロイ, 5-46
 他のポートレット・ビルダーとの比較, 2-2
 レンダリング・モード, 5-56
PDK-PL/SQL
 他のポートレット・ビルダーとの比較, 2-2
PL/SQL
 Gateway, A-119
 owa_cookie パッケージ, A-119
 値の取得と設定, A-117
 ポートレットに対する生成, A-96
plsql.conf ファイル, A-131
PL/SQL イベント・ハンドラ, A-115
 記述, A-116
 ボタン, A-27, A-116

PL/SQL コード・オプション
 チャート・ポートレット, A-87
 フォーム・ポートレット, A-33
 レポート・ポートレット, A-60
PL/SQL ポートレット
 イベントのサポート, 2-28
 開発ツール, 2-18
 外部アプリケーションに対する認証, 2-31
 キャッシュ・スタイル, 2-17
 構築に必要な専門知識, 2-9
 コンテンツのインラインでのレンダリング, 2-24
 コンテンツの取得, 2-22
 セキュリティ, 2-29
 多言語サポート, 2-29
 チャート作成, 2-25
 パラメータのサポート, 2-27
 ページ区切りサポート, 2-30
 他のポートレット・ビルダーとの比較, 2-2
 ユーザー・インタフェース, 2-21
 用途適性, 2-7
Portal Center
 Web クリッピング, 4-14
PORTAL_DEVELOPERS グループ, A-126
PORTAL.ACCOUNTINFO.LABEL 代替タグ, A-138
PORTAL.ACCOUNTINFO.URL 代替タグ, A-138
PORTAL.ACCOUNTINFO 代替タグ, A-138
PORTAL.COMMUNITY.IMAGE 代替タグ, A-138
PORTAL.COMMUNITY.LABEL 代替タグ, A-138
PORTAL.COMMUNITY.URL 代替タグ, A-138
PORTAL.COMMUNITY 代替タグ, A-138
PORTAL.HELP.IMAGE 代替タグ, A-137
PORTAL.HELP.LABEL 代替タグ, A-137
PORTAL.HELP.URL 代替タグ, A-137
PORTAL.HELP 代替タグ, A-137
PORTAL.HOME.IMAGE 代替タグ, A-137
PORTAL.HOME.LABEL 代替タグ, A-137
PORTAL.HOME.URL 代替タグ, A-137
PORTAL.HOME 代替タグ, A-137
PORTAL.LOGOUT.LABEL 代替タグ, A-137
PORTAL.LOGOUT.URL 代替タグ, A-137
PORTAL.LOGOUT 代替タグ, A-137
PORTAL.NAVIGATOR.IMAGE 代替タグ, A-137
PORTAL.NAVIGATOR.LABEL 代替タグ, A-137
PORTAL.NAVIGATOR.URL 代替タグ, A-137
PORTAL.NAVIGATOR 代替タグ, A-137
portalstandards.oracle.com, 5-34

Portal スキーマ

パスワードと Web クリッピング, 4-22

Portal ツール

OmniPortlet の概要, 3-3

PortletRenderer, 5-60

provider.xml

session, 5-77

イベント, 5-70

失効化ベースのキャッシュのアクティブ化, 5-100

設定情報, 5-65

多言語, 5-106

パラメータ, 5-67

プロバイダのリソース・バンドル, 5-106

ポートレットのリソース・バンドル, 5-108

レンダリング・モードの更新, 5-57

S

set_value, A-118

Simple Object Access Protocol (SOAP), 2-10

SOAP, 2-10

SQL

データソースとして使用, 3-7

ポートレットの定義, 3-28

SSL, 5-87

構成, 5-88

Struts

JDeveloper でポートレットを作成, 5-110

OracleAS Portal との統合, 5-113

概要, 5-112

ポートレットの作成, 5-115

U

URL

Web クリッピングへの指定, 4-5

URL ベースのポートレット

Web クリッピングへの移行, 4-2

概要, A-16

URL リライティング・フィールド

Web クリッピング, 4-10

USER.FULLNAME 代替タグ, A-136

USER 代替タグ, A-136

V

VERSION 代替タグ, A-136

W

WAR ファイル

JPS, 5-28

PDK-Java 用, 5-48

Web Services for Remote Portlets

「WSRP」を参照

web.xml ファイル

Web クリッピング, 4-22

Web クリッピング

イベントのサポート, 2-28

開発ツール, 2-18

外部アプリケーションに対する認証, 2-30

キャッシュ・スタイル, 2-17

個人用ページ, 4-13

コンテンツのインラインでのレンダリング, 2-23

コンテンツの取得, 2-22

制限, 4-20

セキュリティ, 2-28

多言語サポート, 2-29

チャート作成, 2-25

テスト・ページ, 4-21

トラブルシューティング, 4-21

パラメータのサポート, 2-27

必要な専門知識, 2-8

フォーム, 4-10, 4-17

プロキシ・サーバー, 4-21, 4-23

プロキシ認証, 4-23

他のポートレット・ビルダーとの比較, 2-2

ユーザー・インタフェース, 2-21

用途適性, 2-5

ロード・バランサ, 4-22

ロギング・レベル, 4-24

Web クリッピング・スタジオ, 4-3

使用, 4-5

Web クリッピング・プロバイダ, 4-1

Web クリッピング・ポートレット, 1-9, 4-1

追加, 4-3

プロパティ, 4-10, 4-16

Web コンテンツ

期限切れ, 4-10

再利用, 4-1

指定, 4-5

タイムアウト値, 4-10

ブラウザ, 4-5

ページへの追加, 4-3

Web サービス
データソースとして使用, 3-9
ポートレットの定義, 3-31
Web ソース, 1-2
Web ブラウザ
推奨するイメージ設定, xviii
推奨するキャッシュ設定, xvii
推奨バージョン, xvii
Web プロバイダ, 2-11
Web ページ
データソースとして使用, 3-10, 3-34
WSRP, 2-2, 2-10, 5-11
プロバイダ, 2-12

X

XML
コンポーネント・ポートレット, A-15
データソースとして使用, 3-8
XML プロバイダ定義
「provider.xml」を参照

あ

アイテム・タイプ
ファイルのアップロード (*interMedia*), A-63
アイテム・タイプ、フォーム・ポートレット
イメージ, A-26
空白, A-26
コンボ・ボックス, A-26
水平定規, A-26
チェック・ボックス, A-26
テキスト・ボックス, A-26
テキスト領域, A-26
パスワード, A-26
非表示, A-26
ファイルのアップロード (*interMedia*), A-26
ファイルのアップロード (バイナリ), A-26
ボタン, A-26
ラベルのみ, A-26
アクセス制御リスト
「ACL」を参照
値リスト
コンボ・ボックス, A-90
作成, A-88
静的, A-88
チェック・ボックス, A-90

動的, A-88
バインド変数, A-91
複数選択, A-90
ポートレット, A-16
ポップアップ・リスト, A-90
ラジオ・グループ, A-90
暗号化
メッセージ, 5-79

い

イベント, 5-66
OmniPortlet での使用, 3-25, 3-43
provider.xml に生成されたコード, 5-70
送信, 5-69
ポートレットへの追加, 3-24
イメージ
Web クリッピングのクリップ, 4-22
仮想ディレクトリ, A-131
推奨するブラウザ設定, xviii
定義の作成, A-131
「イメージ」アイテム・タイプ、フォーム・ポート
レット, A-26
イメージ・データ, A-62
色、定義の作成, A-130
インライン・レンダリング, 2-23
Web クリッピング, 4-2

え

永続的な状態変数, A-119
エクスポート / インポート・ユーティリティ
Web クリッピング, 4-20
エラー
Web クリッピング, 4-21, 4-24

お

オーディオ・データ, A-62

か

階層ポートレット, A-15
ガイドライン
共有画面モードでの CSS, 5-4
共有画面モードでの HTML, 5-3
情報モード, 5-10

- 全画面モード, 5-9
- デフォルト編集モード, 5-7
- プレビュー・モード, 5-8
- ヘルプ・モード, 5-9
- 編集モード, 5-5
- リンク・モード, 5-10
- 外部アプリケーション
 - Web クリッピング, 4-1, 4-11
 - 認証, 5-81
- カスタマイズ
 - ウィザードで生成されたコード, 5-61
 - 生成されたコードの変更, 5-62
 - 「表示」ページ, 5-64
 - 「編集」ページと「デフォルトの編集」ページ, 5-61
 - ポートレット, 5-59
- カスタマイズ権限, A-101
- 仮想ディレクトリ、イメージ, A-131
- カレンダー・ポートレット, A-15
- 「管理」共有コンポーネント権限, A-126
- 管理権限, A-101

き

- 期限切れ
 - ポートレット・コンテンツ, 4-10
- キャッシュ, 5-93
 - アクティブ化, 5-95
 - システム・レベル, 2-16
 - 失効化ベース, 2-17, 5-94
 - 失効化ベースのプロバイダ・サブレット, 5-97
 - 失効化ベース用の provider.xml, 5-100
 - 失効ポート, 5-98
 - 手動によるキャッシュの失効化, 5-101
 - 推奨するブラウザ設定, xvii
 - スタイル, 2-16
 - 妥当性チェック・ベース, 2-16, 5-95
 - 追加, 5-96, 5-97, 5-102
 - 有効期限ベース, 2-16, 5-94
 - ユーザー・レベル, 2-16
- 共有画面モード, 5-3
- 共有コンポーネント, A-125
 - JavaScript の作成, A-129
 - アクセス権限の付与, A-126
 - 編集, A-125

- 共有コンポーネント・プロバイダ
 - イメージ定義の作成, A-131
 - 色定義の作成, A-130
 - フォント定義, A-132
 - ユーザー・インタフェース・テンプレート, A-133
- 拒否されたログオンのメッセージ
 - Web クリッピング, 4-22

く

- 「空白」アイテム・タイプ、フォーム・ポートレット, A-26
- グループ
 - DBA, A-126
 - PORTAL_DEVELOPERS, A-126

け

- 権限, 5-82
 - 「管理」共有コンポーネント, A-126
 - 共有コンポーネントに対するアクセス権限, A-126
 - 個別に付与, A-103
 - 「新規作成」共有コンポーネント, A-126
 - プロバイダ, 5-83
 - プロバイダから継承, A-102
 - ポートレット, 5-83
 - ポートレット構築, A-101
 - ポートレットに対する付与, A-102
- 言語サポート
 - 「多言語ポートレット」を参照

こ

- コード
 - コール・インタフェースの表示, A-98
 - 追加機能, A-113
 - バインド変数, A-114
 - パッケージ仕様の表示, A-97
 - ポートレットのソースの表示, A-97
 - 本体の表示, A-97
- 個人用ページ
 - Web クリッピング, 4-13
- コンテンツ
 - インラインでのレンダリング, 2-23
 - ページへの追加, 4-3
- コンボ・ボックス、LOV, A-90
- 「コンボ・ボックス」アイテム・タイプ、フォーム・ポートレット, A-26

し

- 失効化ベースのキャッシュ, 2-17
- 実行権限, A-101
- 状態
 - Web クリッピングの確認, 4-21
- 情報モード, 5-10
- 書式および妥当性チェック・オプション
 - フォーム・レベル, A-23
 - ボタン, A-24
 - 列, A-27
- 書式条件, A-47
- 書式マスク
 - 数値, A-45
 - 日付, A-45
- 「新規作成」共有コンポーネント権限, A-126
- シングル・サインオン, 5-80
 - 外部アプリケーション, 2-30, 5-81
 - パートナー・アプリケーション, 5-80
 - ポートレット・ビルダーの比較, 2-30

す

- 「水平定規」アイテム・タイプ、フォーム・ポートレット, A-26
- スキーマ, A-2
 - アクセス権限, A-5, A-8
 - 一時表領域, A-4
 - 権限の付与, A-5
 - 作成, A-3
 - デフォルト表領域, A-4
 - デフォルト・プロファイル, A-4
 - 命名, A-3
 - ロールへの登録, A-8
- スプレッドシート
 - データソースとして使用, 3-6

せ

- 制限
 - Web クリッピング, 4-20
- セキュリティ, 5-78
 - 機能, 5-78
 - サーバー, 5-85
 - 通信, 5-79
 - 認可, 5-78
 - 認証, 5-78

- マネージャ, 5-83
- メッセージ認証, 5-86
- セクション
 - Web クリッピング, 4-7
- セクションのクリッピング, 4-7
- セッション
 - 永続的な状態, A-119
- セッション情報, 5-74
 - provider.xml で有効にする, 5-77
 - 記憶域の実装, 5-75
 - 有効なセッションのチェック, 5-77
- 接続
 - Oracle JDeveloper のアプリケーション・サーバー, 5-26, 5-47
- 全画面モード, 5-9

そ

- ソースの表示権限, A-101
- その場で開発するポートレット, 2-3, 2-19, A-14

た

- 代替タグ, A-136
 - ALIGN_LEFT, A-136
 - ALIGN_RIGHT, A-136
 - BODY, A-136
 - DIRECTION, A-136
 - HELPLINK, A-136
 - HELPScript, A-136
 - IMAGE_PREFIX, A-136
 - PAGE.BASE, A-137
 - PAGE.BASE.URL, A-137
 - PAGE.BGColor, A-137
 - PAGE.BGImage, A-137
 - PAGE.CUSTOMIZEPAGE, A-138
 - PAGE.CUSTOMIZEPAGE.LABEL, A-138
 - PAGE.CUSTOMIZEPAGE.URL, A-138
 - PAGE.EDITPAGE, A-138
 - PAGE.EDITPAGE.LABEL, A-138
 - PAGE.EDITPAGE.URL, A-138
 - PAGE.REFRESH, A-138
 - PAGE.REFRESH.LABEL, A-138
 - PAGE.REFRESH.URL, A-138
 - PAGE.STYLE, A-137
 - PAGE.STYLE.URL, A-137
 - PAGE.SUBPAGELINK, A-137

PORTAL.ACCOUNTINFO, A-138
PORTAL.ACCOUNTINFO.LABEL, A-138
PORTAL.ACCOUNTINFO.URL, A-138
PORTAL.COMMUNITY, A-138
PORTAL.COMMUNITY.IMAGE, A-138
PORTAL.COMMUNITY.LABEL, A-138
PORTAL.COMMUNITY.URL, A-138
PORTAL.HELP, A-137
PORTAL.HELP.IMAGE, A-137
PORTAL.HELP.LABEL, A-137
PORTAL.HELP.URL, A-137
PORTAL.HOME, A-137
PORTAL.HOME.IMAGE, A-137
PORTAL.HOME.LABEL, A-137
PORTAL.HOME.URL, A-137
PORTAL.LOGOUT, A-137
PORTAL.LOGOUT.LABEL, A-137
PORTAL.LOGOUT.URL, A-137
PORTAL.NAVIGATOR, A-137
PORTAL.NAVIGATOR.IMAGE, A-137
PORTAL.NAVIGATOR.LABEL, A-137
PORTAL.NAVIGATOR.URL, A-137
USER, A-136
USER.FULLNAME, A-136
VERSION, A-136
「タイムアウト」フィールド
 Web クリッピング, 4-10
タグ、代替, A-136
多言語ポートレット
 provider.xml, 5-106
 比較, 2-29
 リソース・バンドル, 5-103
 レンダラの更新, 5-105
妥当性チェック
 JavaScript, A-127
 JavaScript に関するガイドライン, A-128
 フィールド・レベル, A-127
 フィールド・レベルの作成, A-128
 フォーム・レベル, A-127
 フォーム・レベルの作成, A-128
妥当性チェック・オプション、フォーム・ポート
 レット
 更新可能, A-29
 書式マスク, A-29
 挿入可能, A-29
 デフォルト値, A-29
 デフォルト値タイプ, A-29

必須, A-29
 フィールド・レベルの妥当性チェック, A-29
 フォーム・レベルの妥当性チェック, A-30
妥当性チェック・ベースのキャッシュ, 2-16

ち

チェック・ボックス、LOV, A-90
「チェック・ボックス」アイテム・タイプ、フォーム・
 ポートレット, A-26
チャート・ポートレット, A-15
 PL/SQL コード・オプション, A-87
 アクセス可能な表およびビュー, A-67
 値, A-64, A-69
 カスタマイズ・フォームの書式オプション, A-81
 カスタマイズ・フォームの表示オプション, A-79
 カスタマイズ・フォームのボタン・オプション,
 A-83
 共通の表示オプション, A-75
 集計関数, A-69
 書式条件, A-73
 問合せウィザード, A-64
 比較, 2-25
 フル・ページの表示オプション, A-76
 ポートレットの表示オプション, A-77
 補足テキスト・オプション, A-85
 モバイルの表示オプション, A-78
 ラベル, A-64, A-69
 リンク, A-64, A-69
 例, A-147
 列条件, A-71

つ

通信
 HTTPS, 5-87
 セキュリティ, 5-79

て

ディレクトリ、仮想, A-131
データ
 フィルタリング, 3-12
データ・コンポーネント・ポートレット, A-16
データソース
 SQL の使用, 3-7
 Web サービスの使用, 3-9

XML の使用, 3-8
既存の Web ページの使用, 3-10
スプレッドシートの使用, 3-6
データのフィルタリング, 3-12
ポートレットでの使用, 3-5
データベース・プロバイダ, 2-12, 2-13
テキスト・オプション
 チャート・ポートレット, A-85
 フォーム・ポートレット, A-31
 レポート・ポートレット, A-58
「テキスト・ボックス」アイテム・タイプ、フォーム・
 ポートレット, A-26
「テキスト領域」アイテム・タイプ、フォーム・ポート
 レット, A-26
テクノロジー、ポートレット, 2-1
テスト
 PDK-Java のポートレットとプロバイダ, 5-45
 ポートレット, A-105
テスト・ページ
 Web クリッピング, 4-21
デバッグ情報
 Web クリッピング, 4-24
「デフォルトの編集」ページ, 5-61
デフォルト・プロファイル, A-4
デフォルト編集モード, 5-6
デプロイ
 JPS ポートレット用の WAR ファイル, 5-28
 PDK-Java ポートレット用の WAR ファイル, 5-48
テンプレート, A-57
 「ユーザー・インタフェース・テンプレート」も
 参照, A-133

と

問合せウィザード、チャート, A-64
動的ページ・ポートレット, A-15
登録
 JPS ポートレット, 5-29
 PDJ-Java ポートレット, 5-50
 portalstandards.oracle.com, 5-34
 ローカルの OracleAS Portal, 5-29
トラブルシューティング
 Web クリッピング, 4-21

な

ナビゲーション
 Web クリッピング, 4-1

に

入力パラメータ
 Web クリッピング, 4-10, 4-17
認可, 5-78
認証, 5-78
 外部アプリケーション, 5-81
 サーバー, 5-79
 シングル・サインオン, 5-80
なし, 5-82
 パートナ・アプリケーション, 5-80
 プロキシ
 Web クリッピング, 4-23
 ポートレット・ビルダーの比較, 2-30
 メッセージ, 5-79, 5-86

は

バージョン管理, A-99
パートナ・アプリケーション, 5-80
バインド変数
 LOV, A-91
 使用, A-114
「パスワード」アイテム・タイプ、フォーム・ポート
 レット, A-26
「バッチ」ボタン、追加, A-110
バッチ・モード、ポートレットの実行, A-109
パフォーマンス
 キャッシュ, 5-93
パブリック・ポートレット・パラメータ, 2-26
パラメータ, 5-66
 OmniPortlet での使用, 3-25, 3-43
 provider.xml に生成されたコード, 5-67
 Web クリッピング, 4-10, 4-17
 種類, 2-26
 追加, 5-66
パラメータとイベント
 OmniPortlet での使用, 3-25, 3-43

ひ

- ビデオ・データ, A-62
- 「非表示」アイテム・タイプ、フォーム・ポートレット, A-26
- 「表示」ページ, 5-64
- 表示モード, 5-3
 - リスト, 5-2
- 表領域
 - 一時, A-4
 - デフォルト, A-4

ふ

- 「ファイルのアップロード (*interMedia*)」アイテム・タイプ
 - フォームへの組込み, A-63
 - フォーム・ポートレット, A-26
- 「ファイルのアップロード (*バイナリ*)」アイテム・タイプ、フォーム・ポートレット, A-26
- ファジー・マッチング
 - Web クリップिंग, 4-1
- フォーム
 - Web クリップिंग, 4-10, 4-17
 - ポートレット・ビルダーによる構築, A-19
- フォーム・ポートレット, A-15, A-19
 - doDelete, A-118
 - doInsert, A-118
 - doUpdate, A-118
 - interMedia* リッチ・コンテンツ, A-62
 - JavaScript イベント・ハンドラ, A-27, A-30
 - JavaScript の追加, A-129
 - p_session, A-118
 - p_value, A-118
 - PL/SQL イベント・ハンドラ, A-27
 - PL/SQL コード・オプション, A-33
 - set_value, A-118
 - 値の取得と設定, A-117
 - イメージ・データ, A-62
 - オーディオ・データ, A-62
 - 書式および妥当性チェック・オプション, A-23
 - ビデオ・データ, A-62
 - 補足テキスト・オプション, A-31
- フォント定義、作成, A-132
- 複数選択、LOV, A-90
- 付与, A-5
- プライベート・ポートレット・パラメータ, 2-26

- ブラウザ
 - 推奨するイメージ設定, xviii
 - 推奨するキャッシュ設定, xvii
 - 推奨バージョン, xvii
- ブラウズ
 - Web クリップिंग, 4-7
- プリファレンス・ストア, 5-60
- フレーム・ドライバ・ポートレット, A-16
- プレビュー・モード, 5-8
- プロキシ・サーバーの設定
 - Web クリップिंग, 4-21, 4-23
- プロキシ認証
 - Web クリップिंग, 4-20, 4-23
- プロバイダ, A-2
 - Portal DB プロバイダ, A-21
 - アーキテクチャ, 2-14
 - 共有コンポーネント, A-125
 - 権限, 5-83
 - 権限の継承, A-102
 - スキーマ, A-2
 - データベース・プロバイダ, 2-13
 - ポートレットのデプロイ, 2-10
 - ローカルで構築されたプロバイダ, A-10
- プロバイダ・サブレット
 - 失効化ベースのキャッシュ, 5-97
- プロパティ
 - Web クリップिंग・ポートレット, 4-10, 4-16

へ

- ページ管理者
 - 「ページ設計者を参照」
- ページ設計者、定義, xiv
- ページ・パラメータ, 2-26
 - OmniPortlet での使用, 3-43
- ページ・メタデータ, 2-14
- ベスト・プラクティス
 - Java ポートレット, 5-2
 - 共有画面モードでの CSS, 5-4
 - 共有画面モードでの HTML, 5-3
 - 情報モード, 5-10
 - 全画面モード, 5-9
 - デフォルト編集モード, 5-7
 - プレビュー・モード, 5-8
 - ヘルプ・モード, 5-9
 - 編集モード, 5-5
 - リンク・モード, 5-10

ヘルプ・モード, 5-9
編集権限, A-101
「編集」ページ, 5-61
編集モード, 5-5
変数、永続的な状態, A-119

ほ

ポータル

概要, 1-2

データソースの使用, 3-5

ポータル開発者、定義, xiv

ポートレット

Java ガイドライン, 5-2

Java 標準, 2-2

JPS 準拠の作成, 5-15

JPS 準拠のデプロイ, 5-26

OmniPortlet, 1-13

OmniPortlet の概要, 3-3

Oracle JDeveloper の JPS ウィザード, 5-17

PDK-Java, 5-36

PDK-Java 準拠のデプロイ, 5-46

PDK-Java のテスト, 5-45

PL/SQL の生成, A-96

SQL に基づく定義, 3-28

Struts ポートレットの作成, 5-115

URL, A-16

Web クリッピング, 1-9

Web サービスに基づく定義, 3-31

Web ソース, 1-2

Web ページに基づく定義, 3-34

XML コンポーネント, A-15

アーカイブ・バージョンのアクセス, A-99

「アクセス」タブ, A-94

値リスト, A-16

イベントの追加, 3-24

階層, A-15

「開発」タブ, A-94

開発ツール, 2-18

箇条書きレイアウト, 3-21

カスタマイズ, 5-59

カスタマイズ権限, A-101

カスタマイズ・フォームによる実行, A-107

カレンダー, A-15

管理権限, A-101

「管理」タブ, A-94

機能と特性, 2-1

権限, 5-83

権限の付与, A-102

構築, A-16

コール・インタフェースの表示, A-98

コピー, A-96

削除, A-95

実行, A-105

実行権限, A-101

新規として編集, A-99

スキーマ, A-2

セキュリティ, 5-78

セキュリティ・マネージャ, 5-83

ソース・コードの表示, A-97

ソースの表示権限, A-101

その場で開発する, A-14

タイプ, A-14

多言語, 5-103

チャート, A-15

チャート・レイアウト, 3-16

追加機能, A-113

定義, 1-2

データ・コンポーネント, A-16

テクノロジー, 2-1

テスト, A-105

デプロイ, 2-10

動的ページ, A-15

名前の変更, A-95

ニュース・レイアウト, 3-19

バージョンの管理, A-99

パッケージ仕様および本体の表示, A-97

「バッチ」ボタンの追加, A-110

バッチ・モードでの実行, A-109

ビュー・オプションの変更, 3-13

表レイアウト, 3-15

フォーム, A-15, A-19

フォーム・レイアウト, 3-22

複雑な設定が不要な, 1-7

フル・ページとして実行, A-106

フレーム・ドライバ, A-16

プロバイダ, 2-10, A-2

ページへの OmniPortlet の追加, 3-27

ヘッダーとフッターの追加, 3-13

編集, A-92, A-99

編集権限, A-101

ポートレット管理ページ, A-94

ポートレット構築権限, A-101

ポートレットとして実行, A-106

ポータルレット・ビルダー, 1-14, A-1 ~ A-150
メニュー, A-16
モードのリスト, 5-2
リンク, A-16
レイアウトの変更, 3-14
レポート, A-15
ロックの管理, A-98
ポータルレット・ウィザード
JPS, 5-17
PDK-Java, 5-36
カスタマイズ用に生成されたコード, 5-61
生成されたコードの変更, 5-62
ポータルレットでのページ区切りサポート、比較, 2-30
ポータルレット・ビルダー
イベントのサポート, 2-28
開発ツール, 2-18
キャッシュ・スタイル, 2-17
コンテンツのインラインでのレンダリング, 2-24
セキュリティ, 2-29
多言語サポート, 2-29
チャート作成, 2-25
パラメータのサポート, 2-27
必要な専門知識, 2-9
ページ区切りサポート, 2-30
ポータルレットの構築, A-1
他のポータルレット・ビルダーとの比較, 2-2
ユーザー・インタフェース, 2-21
用途適性, 2-7
ポータルレットを管理する「アクセス」タブ, A-94
ポータルレットを管理する「開発」タブ, A-94
ポータルレットを管理する「管理」タブ, A-94
「ボタン」アイテム・タイプ、フォーム・ポータル
レット, A-26
ポップアップ・リスト、LOV, A-90

ま

マトリックス、ポータルレット・テクノロジー, 2-1

め

メッセージ
暗号化, 5-79
認証, 5-86
メニュー・ポータルレット, A-16

も

モード
共有画面, 5-3
情報, 5-10
全画面, 5-9
デフォルト編集, 5-6
表示モードのリスト, 5-2
プレビュー, 5-8
ヘルプ, 5-9
編集, 5-5
リンク, 5-10
レンダリングの追加, 5-56
モバイル・デバイスの表示オプション, A-53

ゆ

有効期限フィールド
Web クリップング, 4-10
有効期限ベースのキャッシュ, 2-16
ユーザー
ページ設計者, xiv
ポータル開発者, xiv
ユーザー・インタフェース・テンプレート, A-133
構成済, A-133
代替タグ, A-136
非構成済, A-135

ら

ラジオ・グループ、LOV, A-90
「ラベルのみ」アイテム・タイプ、フォーム・ポータル
レット, A-26

り

リソース・バンドル, 5-103
レンダラの更新, 5-105
リンク・ポータルレット, A-16
リンク・モード, 5-10

れ

例

- チーム詳細レポート, A-144
- チーム賞与レポート, A-148
- 平均給与チャート, A-147

レイアウト

- OmniPortlet, 3-14
- OmniPortlet による変更, 3-41
- 箇条書き, 3-21
- チャート, 3-16
- ニュース, 3-19
- 表, 3-15
- フォーム, 3-22

列条件, A-41

レポート・ポートレット, A-15

- interMedia* リッチ・コンテンツ, A-62
 - PL/SQL コード・オプション, A-60
 - イメージ・データ, A-62
 - オーディオ・データ, A-62
 - カスタマイズ・フォーム, A-54, A-56
 - 行順序の表示オプション, A-52
 - 共通の表示オプション, A-49
 - 構築, A-36
 - 書式条件, A-47
 - 数値書式マスク, A-45
 - テンプレート, A-57
 - 日付書式マスク, A-45
 - ビデオ・データ, A-62
 - フル Web ページの表示オプション, A-50
 - 補足テキスト・オプション, A-58
 - モバイル・デバイスの表示オプション, A-53
 - 例, A-144, A-148
 - レイアウト・タイプ, A-43
 - 列条件, A-41
 - 列の表示書式オプション, A-43
 - 列ブレイクの表示オプション, A-51
- ### レンダリング・モード, 5-56
- provider.xml の更新, 5-57

ログ・ファイル

- Web クリップिंग, 4-24
- ロック、ポートレット, A-98

ろ

ロード・バランサ

- Web クリップिंग, 4-22

ロール, A-8

ロギング・レベル

- Web クリップिंग, 4-24

