

Oracle9i Application Server

PL/SQL ゲートウェイの使用

リリース 1.0.2

2001 年 1 月

部品番号 : J02458-02

Oracle9i Application Server PL/SQL ゲートウェイの使用, リリース 1.0.2

部品番号 : J02458-02

原本名 : Oracle9i Application Server Using the PL/SQL Gateway, Release 1.0.2

原本部品番号 : A86263-02

原本著者 : Dave Mathews

原本協力者 : Ron Decker, Pushkar Kapasi, Sanjay Khanna, Eric Lee, Kannan Muthukkaruppan

Copyright © 2000, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

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

当プログラムのリバース・エンジニアリング等は禁止されています。

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

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

危険な用途への使用について

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

Restricted Rights Notice

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

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

目次

| | |
|------------|---|
| はじめに | v |
|------------|---|

1 PL/SQL ゲートウェイの概要

| | |
|---|------|
| PL/SQL ゲートウェイの設定 | 1-1 |
| Database Access Descriptor | 1-2 |
| クライアント・リクエストの処理 | 1-2 |
| PL/SQL ゲートウェイの実行 | 1-3 |
| POST 方式および GET 方式 | 1-4 |
| PL/SQL ゲートウェイの機能の概要 | 1-5 |
| 認証 | 1-5 |
| データベース制御による認証（基本認証モード） | 1-6 |
| 認証解除 | 1-6 |
| カスタム認証（Global OWA、Custom OWA および Per Package 認証モード） | 1-6 |
| authorize ファンクションのインプリメント | 1-6 |
| トランザクション・モデル | 1-8 |
| パラメータの受渡し | 1-8 |
| パラメータのオーバーロード | 1-8 |
| オーバーロードおよび PL/SQL 配列 | 1-9 |
| 柔軟なパラメータの受渡し | 1-10 |
| 大きなパラメータ | 1-11 |
| ファイルのアップロードおよびダウンロード | 1-12 |
| ドキュメント表の定義 | 1-12 |
| CONTENT 列のセマンティック | 1-13 |
| CONTENT_TYPE 列のセマンティック | 1-13 |
| LAST_UPDATED 列のセマンティック | 1-13 |
| DAD_CHARSET 列のセマンティック | 1-14 |

| | |
|--|------|
| 以前のスタイルの Document Table Definition | 1-14 |
| 関連パラメータ | 1-14 |
| document_path (ドキュメント・アクセス・パス) | 1-15 |
| document_proc (ドキュメント・アクセス・プロシージャ) | 1-15 |
| upload_as_long_raw | 1-15 |
| ファイルのアップロード | 1-16 |
| アップロード・ファイルの属性 (MIME タイプ) の指定 | 1-18 |
| 複数のファイルのアップロード | 1-18 |
| ファイルのダウンロード | 1-18 |
| パスのエイリアシング | 1-19 |
| キャッシング | 1-20 |
| 概要 | 1-20 |
| 妥当性チェック方式 | 1-21 |
| 期限方式 | 1-22 |
| PL/SQL ゲートウェイのキャッシュ | 1-22 |
| owa_cache パッケージ | 1-22 |
| 妥当性チェック・モデル | 1-23 |
| 期限モデル | 1-24 |
| システム・レベルとユーザー・レベルのキャッシュ | 1-25 |
| CGI 環境変数 | 1-26 |
| NLS | 1-27 |
| REQUEST_CHARSET CGI 環境変数 | 1-27 |
| REQUEST_IANA_CHARSET CGI 環境変数 | 1-28 |

2 PL/SQL ゲートウェイのインストール

| | |
|--|-----|
| システム要件 | 2-1 |
| インストールの前に | 2-2 |
| インストール | 2-2 |
| 必須パッケージのインストール | 2-2 |
| PL/SQL Web Toolkit パッケージ | 2-3 |
| Oracle Application Server 以外のインストール環境 | 2-3 |
| Oracle Application Server のインストール環境 | 2-3 |
| Oracle HTTP Server Listener の設定 | 2-5 |
| apachectl ファイル | 2-5 |
| httpd.conf ファイル | 2-6 |
| plsql.conf ファイル | 2-6 |
| wdbsvr.app ファイル | 2-6 |

| | |
|---|------|
| PL/SQL ゲートウェイの設定ページへのアクセス | 2-7 |
| plsql.conf 設定ファイル | 2-7 |
| Oracle HTTP Server Listener の起動と停止 | 2-8 |
| 3 PL/SQL ゲートウェイの設定 | |
| 全般設定 | 3-1 |
| Database Access Descriptor の設定 | 3-3 |
| DAD 管理のセキュリティ | 3-6 |
| キャッシュの設定 | 3-7 |
| PL/SQL キャッシュ | 3-8 |
| セッション Cookie キャッシュ | 3-9 |
| 4 PL/SQL ゲートウェイを WebDB とともに使用するための設定 | |
| 開始前の作業 | 4-1 |
| 5 PL/SQL Web Toolkit の使用 | |
| PL/SQL Web Toolkit のインストール | 5-1 |
| PL/SQL Web Toolkit のパッケージ | 5-2 |
| http および htf パッケージ | 5-3 |
| owa_image パッケージ | 5-4 |
| owa_opt_lock パッケージ | 5-4 |
| owa_custom パッケージ | 5-4 |
| owa_content パッケージ | 5-5 |
| owa_cache パッケージ | 5-6 |
| PL/SQL Web Toolkit のパラメータ名の表記規則 | 5-7 |
| HTML タグの属性 | 5-8 |
| PL/SQL ゲートウェイおよびアプレット | 5-8 |
| Cookie | 5-9 |
| LONG データ型 | 5-9 |
| http パッケージおよび htf パッケージの拡張 | 5-9 |
| 文字列のマッチングおよび操作 | 5-10 |
| owa_pattern.match | 5-11 |
| owa_pattern.change | 5-12 |

6 PL/SQL ゲートウェイ・チュートリアル

| | |
|------------------------------------|-----|
| ストアド・プロシージャの作成およびデータベースへのロード | 6-1 |
| アプリケーションを実行する HTML ページの作成 | 6-4 |

索引

はじめに

このマニュアルでは、PL/SQL Gateway for Oracle9i Application Server 1.0.2 のインストール、設定およびメンテナンスの方法について説明します。次の章が含まれています。

- | | |
|-------|--|
| 第 1 章 | PL/SQL ゲートウェイとその機能の概要を説明します。 |
| 第 2 章 | PL/SQL ゲートウェイのインストール方法について説明します。 |
| 第 3 章 | PL/SQL ゲートウェイのグローバルな設定と、各 Database Access Descriptor およびキャッシュの個別設定について説明します。 |
| 第 4 章 | Oracle WebDB リリース 2.1 および 2.2 を PL/SQL ゲートウェイとともに使用する場合の特別な考慮点について説明します。 |
| 第 5 章 | PL/SQL Web Toolkit（以前の OWA Toolkit）のインストール方法について説明します。PL/SQL ゲートウェイを使用する前に、Oracle データベースの SYS スキーマに、PL/SQL Web Toolkit のパッケージをインストールする必要があります。 |
| 第 6 章 | データベースの表の内容を HTML ページとして表示する簡単なアプリケーションの作成および実行方法について、ステップごとに説明します。 |

PL/SQL ゲートウェイの概要

Oracle9i Application Server は、Web アプリケーションの開発および運用のためのオラクル社の中間層製品群を 1 つのソリューションに統合します。Oracle9i Application Server リリース 1.0.2 の標準バージョンには、次のものが含まれます。

- Oracle HTTP Server powered by Apache およびサーブレット・エンジン
- Oracle Java Server Pages (JSP) エンジン
- PL/SQL ゲートウェイ
- Oracle Database Cache
- Oracle Tools (Enterprise Edition に付属)

PL/SQL ゲートウェイは、Web 上での PL/SQL ベースのアプリケーションの構築および運用をサポートします。PL/SQL ストアド・プロシージャにより、データベース表からデータを取得し、Web ブラウザで表示するデータおよびコードを含んだ HTTP レスポンスを生成することが可能です。PL/SQL ゲートウェイは、Oracle Portal 3.0 など、他の Oracle 製品をサポートし、数々の新機能を備えています。

PL/SQL ゲートウェイの設定

データベース・セッションの状態には、PL/SQL パッケージ変数の状態、アプリケーションの状態およびトランザクションの状態が含まれます。

状態を保持しない環境では、クライアントからの各 HTTP リクエストは新規のデータベース・セッションにマッピングされます。アプリケーションの状態は、通常は HTTP Cookie またはデータベース表で維持されます。トランザクションの状態は、複数のリクエスト間にまたがることはできません。PL/SQL プロシージャが正しく実行されると、暗黙的にコミットが実行されます。実行時にエラーが発生すると、暗黙的にロールバックが実行されます。

状態を保持する環境では、クライアントからの各 HTTP リクエストは同じデータベース・セッションにマッピングされます。アプリケーションの状態は、PL/SQL パッケージ変数に格納されます。暗黙的なコミットまたはロールバックは実行されないため、トランザクションは複数のリクエスト間にまたがる可能性があります。

Oracle9i Application Server では、PL/SQL ベースの Web アプリケーションの運用のために、次の 2 種類の設定が提供されています。

- Oracle9i Application Server で `mod_plsql` を使用する場合、状態を保持しないモードでの実行のみサポートされています。状態を保持しない PL/SQL ベースの Web アプリケーションを開発するユーザーには、この設定をお勧めします。`mod_plsql` は、状態を保持しないモードにおいて接続プーリング・メカニズムを備えており、複数の HTTP リクエストにまたがってデータベース・セッションを開いておくことが可能です。
- Oracle9i Application Server で `mod_ose` を使用する場合、状態を保持しないモードおよび状態を保持するモードでの実行がサポートされています。状態を保持する PL/SQL ベースおよび Java ベースの Web アプリケーションを開発するユーザーには、この設定をお勧めします。HTTP リクエストごとに新規のデータベース・セッションを作成および破棄する必要がないため、`mod_ose` を使用する場合には状態を保持するモードをお勧めします。詳細は、`mod_ose` のドキュメントを参照してください。

Database Access Descriptor

PL/SQL ゲートウェイの各リクエストは、Database Access Descriptor (DAD) に関連付けられています。DAD は、データベース・アクセスに使用される設定値の名前付きのセットです。DAD により、次のような情報が指定されます。

- データベース・エイリアス (Net8 サービス名)。
- データベースがリモートの場合、その接続文字列。
- ドキュメントのアップロードおよびダウンロード用のプロシージャ。

また、DAD にはユーザー名およびパスワード情報を指定できます。指定がない場合には、URL の実行時にユーザー名とパスワードを入力するプロンプトが表示されます。詳細は、1-5 ページの「[認証](#)」を参照してください。

クライアント・リクエストの処理

サーバーがリクエストを受信すると、次の処理が実行されます。

1. Web サーバーがクライアントから PL/SQL ゲートウェイのリクエストを受信し、そのリクエストを PL/SQL ゲートウェイに転送します。
2. PL/SQL ゲートウェイは、DAD の設定値（詳細は、3-1 ページの「[PL/SQL ゲートウェイの設定](#)」を参照）を使用し、データベースへの接続方法を決定します。
3. PL/SQL ゲートウェイは、データベースに接続し、コール・パラメータを準備して、データベース内の PL/SQL プロシージャを実行します。
4. PL/SQL プロシージャは、HTTP レスポンス（たとえば、HTML ページなど）を生成します。この HTTP レスポンスには静的データのみでなく、データベースの表から取り出した動的データを含めることができます。

5. プロシージャからの出力は、PL/SQL ゲートウェイとクライアントに返されます。

PL/SQL ゲートウェイが実行するプロシージャは、HTTP レスポンスをクライアントに返す必要があります。この作業を簡単にするために、PL/SQL ゲートウェイには PL/SQL Web Toolkit が付属しています。PL/SQL Web Toolkit は、ストアド・プロシージャ内でリクエスト情報の取得、HTML タグの作成およびクライアントへのヘッダー情報の返信などに使用可能なパッケージのセットです。すべてのユーザーがアクセスできるように、このツールキットは共通スキーマにインストールします。詳細は、5-1 ページの「[PL/SQL Web Toolkit の使用](#)」を参照してください。

PL/SQL ゲートウェイの実行

PL/SQL ゲートウェイを Web ブラウザで実行するには、一般的に次の形式の URL を使用する必要があります。

```
protocol://hostname[:port]/prefix/DAD/[[:!]] [schema.] [package.] proc_name[?query_string]
```

各項目の説明

protocol には、http または https のいずれかを使用できます。SSL の場合は、https を使用します。

hostname は、Web サーバーが稼動しているマシンです。

port は、アプリケーション・サーバーがリスニングしているポートです。指定しない場合、ポート 80 が使用されます。

prefix は、PL/SQL リクエストを処理するために Web サーバーで設定した仮想パスです。このパラメータのデフォルトの設定は *p1s* です。たとえば、Web サーバーで *p1s* を接頭辞に設定して、先頭に *p1s* が付いたリクエストをすべて PL/SQL ゲートウェイにルーティングすることが可能です。

DAD は、この URL に使用する DAD エントリです。

! 記号が存在する場合、柔軟なパラメータの受渡しスキームを使用する必要があることを示します。詳細は、1-10 ページの「[柔軟なパラメータの受渡し](#)」を参照してください。

schema は、データベース・スキーマ名です。指定しない場合、*package.proc_name* の名前解決は、URL リクエストが処理されたデータベース・ユーザーに基づいて行われます。

package は、PL/SQL ストアド・プロシージャを含んだパッケージです。指定しない場合、プロシージャはスタンドアロンになります。

proc_name には、実行する PL/SQL ストアド・プロシージャを指定します。ファンクションではなく、プロシージャを指定する必要があります。IN 引数のみ使用可能です。

?query_string には、ストアド・プロシージャのパラメータ（存在する場合）を指定します。この文字列は、GET 方式の書式に従います。たとえば、次のようになります。

- 複数のパラメータは & 文字で区切り、渡される値内の空白文字は + 文字に置き換えます。
- HTML フォームを使用して文字列を生成する場合（文字列を自分で生成するのではなく）、書式化は自動的に行われます。
- HTTP リクエストで HTTP の POST 方式を使用してデータを PL/SQL ゲートウェイに送信する場合もあります。詳細は、1-4 ページの「[POST 方式および GET 方式](#)」を参照してください。

たとえば、Web サーバーに pls が接頭辞として設定されており、ブラウザが次の URL を送信したとします。

```
http://www.acme.com:9000/pls/mydad/mypackage.myproc
```

この場合、www.acme.com で稼動し、ポート 9000 でリスニングしている Web サーバーがリクエストを処理します。Web サーバーは、リクエストを受信すると、そのリクエストを PL/SQL ゲートウェイに渡します。これは、Web サーバーが PL/SQL ゲートウェイを実行するよう設定されていることを pls 接頭辞が示しているためです。次に、PL/SQL ゲートウェイは、mydad に関連付けられている DAD を使用して、mypackage に格納されている myproc プロシージャを実行します。

URL は、DAD、スキーマまたはストアド・プロシージャ名を含めずに指定することもできます。たとえば、次のように指定したとします。

```
http://www.acme.com:9000/pls/mydad
```

この場合、mydad という DAD のデフォルトのホーム・ページ（PL/SQL ゲートウェイの設定ページで指定されている）が表示されます。

次のように指定したとします。

```
http://www.acme.com:9000/pls
```

この場合、DAD のデフォルトのホーム・ページが起動されます。

パラメータの受渡しは名前で行われるため、通常、URL または HTTP ヘッダー内での PL/SQL パラメータの指定順序を考慮する必要はありません。ただし、この規則にはいくつか例外があります。詳細は、1-8 ページの「[パラメータの受渡し](#)」を参照してください。

POST 方式および GET 方式

HTTP プロトコルの POST 方式および GET 方式は、パラメータ・データをアプリケーションに渡す方法（通常は名前と値のペア形式）をブラウザに対して指示します。通常、パラメータ・データは HTML フォームによって生成されます。

PL/SQL ゲートウェイ・アプリケーションでは、いずれの方式も使用できます。各方式の安全性は、使用する転送プロトコル（http または https）によって決定されます。

POST 方式を使用する場合、パラメータはリクエストの BODY で渡されます。GET 方式を使用する場合、パラメータは問合せ文字列を使用して渡されます。これらの方式については、HTTP 1.1 の仕様で説明されています。HTTP 1.1 仕様は、次に示す W3C の Web サイトで参照できます。

<http://www.w3.org/Protocols/HTTP/1.1/draft-ietf-http-v11-spec-rev-01.txt>

GET 方式には、使用するオペレーティング・システムにより、名前と値のペアの値の長さが環境変数の値の最大長を超えることができないという制限があります。さらに、オペレーティング・システムにより、定義可能な環境変数の数も制限されます。

一般的に、大量のパラメータ・データをサーバーに渡す場合は、POST 方式を使用することをお勧めします。

PL/SQL ゲートウェイの機能の概要

認証

PL/SQL ゲートウェイには、Web サーバーが備える認証以外にも、いくつかのレベルの認証が用意されています。Web サーバーは、ドキュメントや仮想パスなどを保護し、PL/SQL ゲートウェイは、ユーザーのデータベースへのログインや PL/SQL Web アプリケーションの実行を保護します。

それぞれの認証モードは、PL/SQL ゲートウェイの設定ページの「**Authentication Mode**」パラメータを使用して使用可能に設定できます。このパラメータには、次の値のいずれかを設定できます。

- Basic - 基本 HTTP 認証を使用して認証が実行されます。ほとんどのアプリケーションでは Basic 認証を使用します。
- Global Owa - 認証は、PL/SQL Web Toolkit パッケージが含まれているスキーマで実行されます。
- Custom Owa - 認証は、ユーザーのスキーマ、またはそれが見つからない場合は PL/SQL Web Toolkit パッケージが含まれているスキーマ内のパッケージおよびプロシージャを使用して実行されます。
- Per Package - 認証は、ユーザーのスキーマ内のパッケージおよびプロシージャによって実行されます。
- Single Sign-On - 認証は、Login Server の Oracle シングル・サインオン機能を使用して実行されます。このモードは、Login Server でアプリケーションを使用するように設定されている場合のみ使用可能です。

データベース制御による認証（基本認証モード）

PL/SQL ゲートウェイは、データベース・レベルの認証をサポートします。HTTP 基本認証を使用しますが、資格証明を使用してデータベースへのログオンを試みることによって資格証明を認証します。認証は、ユーザーのデータベース・アカウントに対して検証されます。次のいずれかのユーザー名とパスワードが使用されます。

- DAD に格納されている情報。エンド・ユーザーはログインする必要はありません。この方式は、パブリックの情報を提供する Web ページの場合に役立ちます。
- ブラウザ・ベースの HTTP 基本認証のダイアログ・ボックスでユーザーが入力した情報。エンド・ユーザーは、ダイアログ・ボックスにユーザー名とパスワードを入力する必要があります。

認証解除 PL/SQL ゲートウェイでは、ブラウザのすべてのインスタンスを終了せずに、PL/SQL プロシージャを使用してプログラムによってユーザーがログオフする（HTTP 認証情報を消去する）ことが可能です。この機能は Cookie を使用するため、Netscape 3.0 以上および Internet Explorer でサポートされています。その他のブラウザでは、ユーザーは認証を解除するためにブラウザを終了する必要がある場合があります。

認証を解除する別の方法として、URL 内の DAD の後に /logmeoff を追加することもできます。たとえば、次のようにします。

```
http://myhost:2000/pls/myDAD/logmeoff
```

カスタム認証（Global OWA、Custom OWA および Per Package 認証モード）

カスタム認証を使用すると、データベース・レベルではなく、アプリケーションにおいてユーザーを認証することが可能です。認証は、ユーザー作成の認証ファンクションを実行して行われます。

authorize ファンクションのインプリメント

カスタム認証には、設定ファイルに格納された静的なユーザー名 / パスワードが使用されます。動的なユーザー名 / パスワード認証と組み合わせて使用することはできません。

authorize ファンクションの構文は次のとおりです。

```
function authorize return boolean;
```

カスタム認証を使用可能にするには、次のようにします。

1. 「DAD Configuration」 ページで認証レベルを設定します。
2. authorize ファンクションをインプリメントします。

PL/SQL ゲートウェイは、DAD に設定されているユーザー名とパスワードを使用してデータベースにログインします。いったんログインすると、認証制御はアプリケーションに渡されます。次に、アプリケーション・レベルの PL/SQL フック（コールバック・ファンクショ

ン) がコールされます。これらのコールバック・ファンクションのインプリメンテーションは、アプリケーション開発者が行います。コールバック・ファンクションの戻り値により、認証が成功したかまたは失敗したかを判断します。TRUE が返された場合は、認証が正常に行われたことを示します。戻り値が FALSE の場合、認証は失敗し、アプリケーション内のコードは実行されません。

認証ファンクションをいつ実行するかにより、そのファンクションを様々な場所に配置できます。

すべてのユーザーおよびすべてのプロシージャに対して同じ認証ファンクションを実行する場合は、「DAD Configuration」ページの「**Authentication Mode**」リストで「**Global OWA**」を選択します。次に、PL/SQL Web Toolkit が含まれているスキーマ、つまり SYS 内に owa_custom.authorize ファンクションをインプリメントします。

各ユーザーおよびすべてのプロシージャごとに異なる認証ファンクションを実行する場合は、「DAD Configuration」ページの「**Authentication Mode**」リストで「**Custom OWA**」を選択します。その後、各ユーザーのスキーマに owa_custom.authorize ファンクションをインプリメントします。自分のスキーマにファンクションが存在しないユーザーの場合、PL/SQL Web Toolkit パッケージ・スキーマの owa_custom.authorize ファンクションがかわりに実行されます。

認証ファンクションをすべてのユーザーに対して、ただし特定のパッケージ内のプロシージャの場合のみまたは無名プロシージャの場合のみ実行する場合は、「DAD Configuration」ページの「**Authentication Mode**」リストで「**Per Package**」を選択します。その後、各ユーザーのスキーマ内のそのパッケージに authorize ファンクションをインプリメントします。プロシージャがパッケージ内に存在しない場合、無名の authorize ファンクションがかわりにコールされます。次の表に、パラメータの値のサマリーを示します。

| モード | アクセス制御の有効範囲 | コールバック・ファンクション |
|-------------|-------------|---|
| Global OWA | すべてのパッケージ | OWA パッケージ・スキーマ内の owa_custom.authorize |
| Custom OWA | すべてのパッケージ | ユーザーのスキーマ内の owa_custom.authorize、またはそのスキーマに存在しない場合は OWA パッケージ・スキーマ内の owa_custom.authorize |
| Per Package | 指定したパッケージ | ユーザーのスキーマ内の packageName.authorize、または anonymous.authorize がコールされます。 |

トランザクション・モデル

プロシージャを実行するために URL リクエストを処理した後、エラーがある場合、PL/SQL ゲートウェイによりロールバックが実行されます。そうでない場合、ゲートウェイによりコミットが実行されます。このメカニズムでは、トランザクションが複数の HTTP リクエストにまたがることはできません。この状態を保持しないモデルの場合、アプリケーションは、通常 HTTP Cookie またはデータベース表を使用して状態を維持します。状態を保持するモードと保持しないモードの詳細は、1-1 ページの「[PL/SQL ゲートウェイの設定](#)」を参照してください。

パラメータの受渡し

PL/SQL ゲートウェイは、次の処理をサポートしています。

- 名前によるパラメータの受渡し

プロシージャまたはファンクションを起動する URL 内の各パラメータは、一意の名前により識別されます。パラメータのオーバーロードがサポートされています。詳細は、1-8 ページの「[パラメータのオーバーロード](#)」を参照してください。

- 柔軟なパラメータの受渡し

プロシージャの先頭に '!' 記号が付加されます。詳細は、1-10 ページの「[柔軟なパラメータの受渡し](#)」を参照してください。

- 大きなパラメータ（最大 32KB）

詳細は、1-11 ページの「[大きなパラメータ](#)」を参照してください。

パラメータのオーバーロード

オーバーロードにより、名前が同じでパラメータの数、順序またはデータ型のファミリが異なる複数のサブプログラム（プロシージャまたはファンクション）を使用できます。オーバーロードされたサブプログラムをコールすると、PL/SQL コンパイラは渡されたデータ型に基づいて、どのサブプログラムをコールするかを決定します。

PL/SQL では、ローカル・サブプログラムとパッケージ・サブプログラムをオーバーロードできますが、スタンドアロン・サブプログラムはオーバーロードできません。PL/SQL のオーバーロードの詳細は、Oracle Server ドキュメントの『Oracle8i PL/SQL ユーザーズ・ガイドおよびリファレンス』を参照してください。

パラメータ数が同じサブプログラムをオーバーロードする場合は、パラメータに違う名前を付ける必要があります。HTML データはデータ型に関連付けられないため、PL/SQL ゲートウェイは、どのバージョンのサブプログラムをコールすればよいか判断できません。

たとえば、PL/SQL では、次の例に示す 2 つのプロシージャを定義できます。これらのプロシージャのパラメータ名が同じ場合、PL/SQL ゲートウェイで使用するとエラーが発生します。


```
-- legal PL/SQL, but not for the PL/SQL Gateway
CREATE PACKAGE my_pkg AS
  PROCEDURE my_proc (val IN VARCHAR2);
  PROCEDURE my_proc (val IN NUMBER);
END my_pkg;
```

エラーを回避するには、パラメータに異なる名前をつけます。たとえば、次のようになります。

```
-- legal PL/SQL and also works for the PL/SQL Gateway
CREATE PACKAGE my_pkg AS
  PROCEDURE my_proc (valvc2 IN VARCHAR2);
  PROCEDURE my_proc (valnum IN NUMBER);
END my_pkg;
```

最初のバージョンのプロシージャを実行する URL は次のようになります。

```
http://www.acme.com/pls/myDAD/my_pkg.my_proc?valvc2=input
```

2 番目のバージョンのプロシージャを実行する URL は次のようになります。

```
http://www.acme.com/pls/myDAD/my_pkg.my_proc?valnum=34
```

オーバーロードおよび PL/SQL 配列

パラメータ名が同じで、1 つのプロシージャのデータ型が `owa_util.ident_arr` (`varchar2` のテーブル)、もう 1 つのプロシージャのデータ型がスカラー型のオーバーロード PL/SQL プロシージャがあるとします。このような場合でも、PL/SQL ゲートウェイはこの 2 つのプロシージャを区別できます。たとえば、次のプロシージャがあるとします。

```
CREATE PACKAGE my_pkg AS
  PROCEDURE my_proc (val IN VARCHAR2); -- scalar data type
  PROCEDURE my_proc (val IN owa_util.ident_arr); -- array data type
END my_pkg;
```

これらのプロシージャには、それぞれ `val` という同じ名前のパラメータが 1 つあります。

PL/SQL ゲートウェイが `val` パラメータの値を 1 つだけ持つリクエストを受け取ると、スカラー・データ型のプロシージャが実行されます。`val` パラメータの値が複数存在するリクエストを受け取ると、配列データ型のプロシージャが実行されます。

例 1: 次の URL を送信したとします。

```
http://www.acme.com/pls/myDAD/my_proc?val=john
```

スカラー・バージョンのプロシージャが実行されます。

例 2: 次の URL を送信したとします。

```
http://www.acme.com/pls/myDAD/my_proc?val=john&val=sally
```

配列バージョンのプロシージャが実行されます。

確実に配列バージョンのプロシージャを実行できるようにする場合は、HTML ページで隠しフォーム・エレメントを使用してダミーの値を送信します。このダミーの値は、プロシージャ内でチェックされ、破棄されます。

柔軟なパラメータの受渡し

ユーザーが任意の数のエレメントを選択可能な HTML フォームを使用することができます。これらのエレメントの名前が異なる場合、可能性のある各組合せを処理するオーバーロード・プロシージャを作成する必要があります。あるいは、かわりに隠しフォーム・エレメントを挿入すると、ユーザーが選択するエレメントに関係なく、問合せ文字列の名前を毎回一貫させることが可能です。

PL/SQL ゲートウェイでは、柔軟なパラメータ受渡しスキームをサポートしているため、これがより簡単に行えます。URL ベースのプロシージャ実行で柔軟なパラメータの受渡しを使用するには、URL 内でプロシージャ名の先頭に '!' 記号を付加します。

PL/SQL ゲートウェイでは、2 つまたは 4 つのパラメータを持つ柔軟なパラメータをサポートしています。

注意: PL/SQL ゲートウェイで使用する場合、2 パラメータ・インタフェースの方がパフォーマンスが優れているため、こちらの使用をお勧めします。互換性維持のために 4 パラメータ・インタフェースもサポートされています。

4 パラメータのシグネチャを次に示します。

```
procedure [proc_name] is
    (num_entires IN NUMBER,
     name_array IN [array_type],
     value_array IN [array_type],
     reserved in [array_type]);
```

[*proc_name*] は、実行する PL/SQL プロシージャの名前です。

num_entries は、問合せ文字列内の *name_value* のペア数を示します。

name_array は、問合せ文字列から取り出される名前 (1 から順番に索引付けされる) を送信された順序で示します。

value_array は、問合せ文字列から取り出される値 (1 から順番に索引付けされる) を送信された順序で示します。

reserved は現在使用されていません。今後の使用のために確保されています。

[*array_type*] は、varchar2 型のテーブルによる任意の PL/SQL 索引です (owa.vc_arr など)。

2 パラメータのシグネチャを次に示します。

```

procedure [proc_name] is
    name_array IN  [array_type],
    value_array IN [array_type],

```

[*proc_name*] は、実行する PL/SQL プロシージャの名前です。

name_array は、問合せ文字列から取り出される名前（1 から順番に索引付けされる）を送信された順序で示します。

value_array は、問合せ文字列から取り出される値（1 から順番に索引付けされる）を送信された順序で示します。

[*array_type*] は、varchar2 型のテーブルによる任意の PL/SQL 索引です（owa.vc_arr など）。

例 1: 次の URL を送信したとします。

```
http://www.acme.com/pls/myDAD/!scott.my_proc?x=john&y=10&z=doe
```

'!' 接頭辞により、柔軟なパラメータの受渡しを使用する必要があることを PL/SQL ゲートウェイに知らせます。これにより、プロシージャ *scott.myproc* が実行され、次の 2 つの引数が渡されます。

```

name_array ==> ('x', 'y', 'z')
values_array ==> ('john', '10', 'doe')

```

例 2: *query_string* 部分で名前 "x" が重複している次の URL を送信したとします。

```
http://www.acme.com/pls/myDAD/!scott.my_pkg.my_proc?x=a&y=b&x=c
```

'!' 接頭辞により、柔軟なパラメータの受渡しを使用する必要があることを PL/SQL ゲートウェイに知らせます。これにより、プロシージャ *scott.my_pkg.myproc* が実行され、次の 4 つの引数が渡されます。

```

num_entries ==> 3
name_array ==> ('x', 'y', 'x')
values_array ==> ('a', 'b', 'c')
reserved ==> ()

```

大きなパラメータ

前述の「オーバーロードおよび PL/SQL 配列」と「柔軟なパラメータの受渡し」で、スカラーまたは varchar2 引数のテーブルによる索引を持つプロシージャの実行に、PL/SQL ゲートウェイを使用できることを示しました。スカラー引数として渡される値と、varchar2 引数のテーブルによる索引の要素として渡される値に使用できるサイズは、最大 32KB です。

たとえば、柔軟なパラメータの受渡し（1-10 ページの「[柔軟なパラメータの受渡し](#)」を参照）を使用する場合、URL の `query_string` 部分の名前または値は、それぞれ実行されるプロシージャの `name_array` または `value_array` 引数のエレメントとして渡されます。これらの名前または値に使用できるサイズは、最大 32KB です。

ファイルのアップロードおよびダウンロード

PL/SQL ゲートウェイでは、次のことが可能です。

- キャラクタ・セットを変換せずに、ファイルをロー・バイト・ストリームとしてアップロードおよびダウンロードします。ファイルは、ドキュメント表にアップロードされます。PL/SQL アップロード・ハンドラ・ルーチンが表の適切な列を取得できるよう、主キーが渡されます。
- 異なるアプリケーションのファイルを混同しないよう、アプリケーションごとに 1 つ以上、アップロードされるファイル用の表を指定します。
- 問合せ文字列を使用しない形式の URL 経由でこれらの表のファイルにアクセスできるようにします。たとえば、次のようになります。

```
http://www.acme.com:9000/mysite/pls/docs/cs250/lecture1.htm
```

これは、URL の相互参照を含んだファイル・セットのアップロードをサポートするために必要です。

- 1 回のフォーム送信につき複数のファイルをアップロードします。
- ドキュメント表の LONG RAW および BLOB 型の列にファイルをアップロードします。

ドキュメント表の定義

PL/SQL ゲートウェイでは、DAD ごとにドキュメント格納表を指定できます。ドキュメント格納表には、次の定義が必要です。

```
CREATE TABLE [table_name] (  
    NAME                VARCHAR2(256) UNIQUE NOT NULL,  
    MIME_TYPE           VARCHAR2(128),  
    DOC_SIZE            NUMBER,  
    DAD_CHARSET         VARCHAR2(128),  
    LAST_UPDATED        DATE,  
    content_type        VARCHAR2(128),  
    [content_column_name] [content_column_type]  
    [ , [content_column_name] [content_column_type]]  
);
```

`table_name` は、ユーザーが選択できます。`content_column_type` 型には、LONG RAW または BLOB のいずれかを使用します。

`content_column_name` は、対応する `content_column_type` によって異なります。

- `content_column_type` が LONG RAW の場合、`content_column_name` は `CONTENT` にする必要があります。
- `content_column_type` が BLOB の場合、`content_column_name` は `BLOB_CONTENT` にする必要があります。

次に、有効なドキュメント表の定義例を示します。

```

NAME                VARCHAR(128)    UNIQUE NOT NULL,
MIME_TYPE            VARCHAR(128),
DOC_SIZE             NUMBER,
DAD_CHARSET          VARCHAR(128),
LAST_UPDATED         DATE,
CONTENT_TYPE         VARCHAR(128),
CONTENT              LONG RAW,
BLOB_CONTENT         BLOB ;

```

CONTENT 列のセマンティック

表の実際の内容は、CONTENT 列に格納されます。ドキュメント表には、複数の CONTENT 列を含めることが可能です。ただし、ドキュメント表の各行について、CONTENT 列は 1 つだけ使用されます。その他の CONTENT 列は NULL に設定されます。

CONTENT_TYPE 列のセマンティック

CONTENT_TYPE 列は、ドキュメントが格納されている CONTENT 列を追跡するために使用されます。ドキュメントがアップロードされると、PL/SQL ゲートウェイにより、この列の値が型名（つまり、ドキュメントのアップロード先の CONTENT 列の [content_column_type]）に設定されます。

たとえば、ドキュメントが BLOB_CONTENT 列にアップロードされた場合、ドキュメントの CONTENT_TYPE 列には文字列 "BLOB" が設定されます。

LAST_UPDATED 列のセマンティック

LAST_UPDATED 列には、ドキュメントの作成または最終更新日時が反映されます。ドキュメントがアップロードされると、PL/SQL ゲートウェイにより、ドキュメントの LAST_UPDATED 列にアップロード時のデータベース・サーバーの時間（sysdate() で取得）が設定されます。その後アプリケーションがドキュメントの内容または属性を変更した場合、LAST_UPDATED の時間も更新する必要があります。

LAST_UPDATED 列は、HTTP クライアント（ブラウザなど）がキャッシュに入っているバージョンのドキュメントを使用してもよいかどうかを PL/SQL ゲートウェイがチェックし、その結果を HTTP クライアントに知らせるために使用されます。これにより、ネットワーク・トラフィックと応答時間が削減され、サーバーのパフォーマンスとスケーラビリティが改善されます。

DAD_CHARSET 列のセマンティック

DAD_CHARSET 列は、ファイルのアップロード時のキャラクタ・セット設定を追跡します。

注意: この列は、今後の使用のために確保されています。

以前のスタイルの Document Table Definition

WebDB R2.x の以前のリリースで使用されているドキュメント・モデルとの下位互換性を維持するために、PL/SQL ゲートウェイでは、次のように CONTENT_TYPE、DAD_CHARSET および LAST_UPDATED 列が存在しない、以前のドキュメント格納表の定義もサポートしています。

```
/* older style document table definition (DEPRECATED) */
CREATE TABLE [table_name]
(
    NAME          VARCHAR2(128),
    MIME_TYPE     VARCHAR2(128),
    DOC_SIZE      NUMBER,
    CONTENT       LONG RAW
);
```

関連パラメータ

各 DAD について、次の設定パラメータがファイルのアップロード / ダウンロードに関係します。

document_table (document_table_name)

document_table パラメータは、この DAD によってファイルがアップロードされたときにドキュメントの格納に使用する表の名前を指定します。

構文

```
document_table = [document_table_name]
```

例

```
document_table = my_documents
```

または

```
document_table = scott.my_document_table
```

document_path (ドキュメント・アクセス・パス)

これは、ドキュメントにアクセスするための URL 内で、DAD 名の直後に付加されるパス・エレメントを指定します。たとえば、ドキュメントのアクセス・パスが docs の場合、ドキュメントにアクセスするための URL は、次のようになります。

```
http://neon/pls/myDAD/docs/myfile.htm
```

この場合、myDAD は DAD 名で、myfile.htm はファイル名です。ドキュメント・アクセス・パス・メカニズムにより、WebDB の Web サイト構築機能で必要な標準スタイルのドキュメント・アクセス URL が使用可能になります。

構文

```
document_path = [document_access_path_name]
```

document_proc (ドキュメント・アクセス・プロシージャ)

これは、パラメータを持たない、アプリケーション指定のプロシージャで、ドキュメント・アクセス・パスを持つ URL リクエストを処理します。ドキュメント・アクセス・プロシージャは、ファイルのダウンロードを開始するために、wpg_docload.download_file(filename) をコールする必要があります。ドキュメント・アクセス・プロシージャが、完全な URL 指定に基づいてファイル名を取得できるようにする必要があります。たとえば、これをアプリケーションで使用して、ファイル・レベルのアクセス制御やバージョン管理をインプリメントすることが可能です。このようなアプリケーションの例を、1-18 ページの「[ファイルのダウンロード](#)」に示します。

構文

```
document_proc = [document_access_procedure_name]
```

例

```
document_proc = my_access_procedure
```

または

```
document_proc = scott.my_pkg.my_access_procedure
```

upload_as_long_raw

DAD パラメータ upload_as_long_raw は、ファイル拡張子に基づいてファイルのアップロードを設定するために使用されます。DAD パラメータ upload_as_long_raw の値は、ファイル拡張子のカンマ (,) 区切りのリストです。これらの拡張子を持つファイルは、PL/SQL ゲートウェイにより、ドキュメント表の long_raw 型の CONTENT 列にアップロードされます。他の拡張子を持つファイルは、BLOB_CONTENT 列にアップロードされません。

ファイル拡張子には、テキスト定数 (jpeg, gif など) を使用できます。さらに、アスタリスク (*) を特別なファイル拡張子として使用できます。これは、upload_as_long_raw 設定で明示的にリストされていない拡張子を持つすべてのファイルに一致します。

構文

```
upload_as_long_raw = [file_extension][,[file_extension]]*
```

[file_extension] はファイル拡張子 ('.' 記号はあってもなくても結構で、たとえば, txt または .txt が使用可能です。) またはワイルド・カード文字 '*' です。

例

```
upload_as_long_raw = html, txt  
upload_as_long_raw = *
```

ファイルのアップロード

クライアント・マシンからデータベースにファイルをアップロードする場合は、次に示す内容を含んだ HTML ページを作成します。

- enctype 属性に multipart/form-data が設定され、action 属性にアクション・プロシージャと呼ばれる PL/SQL ゲートウェイ・プロシージャ・コールが関連付けられている FORM タグ。
- type 属性と name 属性に file が設定された INPUT エレメント。INPUT type="file" エレメントを使用すると、ユーザーはファイル・システムのファイルを参照して、そこからファイルを選択できます。

ユーザーが「送信」ボタンをクリックしてフォーム・アクションをトリガーすると、次のイベントが発生します。

1. ブラウザは、ユーザーが指定したファイルの内容とその他のフォーム・データをサーバーにアップロードします。
2. PL/SQL ゲートウェイは、ファイルの内容をデータベースのドキュメント格納表内に格納します。表の名前は、DAD パラメータ document_table 設定から導出されます。
3. FORM の action 属性で指定したアクション・プロシージャが、ファイルのアップロードを行わずに PL/SQL ゲートウェイ・プロシージャを実行する場合と同じように実行されます。

次の例に、アップロードするファイルをユーザーがファイル・システムから選択可能な HTML フォームを示します。このフォームには、ユーザーがファイルに関する情報を入力できるその他のフィールドも含まれています。


```

<html>
<head>
<title>test upload</title>
</head>
<body>
  <FORM enctype="multipart/form-data"
action="pls/myDAD/write_info"
method="POST">
<p>Author's Name:<INPUT type="text" name="who">
<p>Description:<INPUT type="text" name="description"><br>
<p>File to upload:<INPUT type="file" name="file"><br>
<p><INPUT type="submit">
</FORM>
</body>
</html>

```

ユーザーがフォームの「送信」ボタンをクリックすると、ブラウザにより、INPUT type="file" エレメントにリストされたファイルがアップロードされます。

次に、write_info プロシージャが実行されます。このプロシージャは、フォームのフィールドの情報をデータベース内の表に書き込み、ページをユーザーに返します。アクション・プロシージャは、ユーザーにレスポンスを返す必要はありませんが、アップロード処理が成功したか失敗したかをユーザーに知らせるようにすることをお勧めします。

次に、サンプルの write_info プロシージャを示します。

```

procedure write_info (
  who          in varchar2,
  description in varchar2,
  file         in varchar2) as
begin
  insert into myTable values (who, description, file);
  http.htmlopen;
  http.headopen;
  http.title('File Uploaded');
  http.headclose;
  http.bodyopen;
  http.header(1, 'Upload Status');
  http.print('Uploaded ' || file || ' successfully');
  http.bodyclose;
  http.htmlclose;
end;

```

名前の競合の可能性を低減するために、ブラウザから取得したファイル名の先頭に、生成されたディレクトリ名が付加されます。フォームで指定されたアクション・プロシージャにより、この名前を必要な名前に変更する必要があります。このため、たとえば /private/minutes.txt がアップロードされた場合、ゲートウェイによって表に格納された名前は、F9080/private/minutes.txt のようになります。アプリケーションは、

コールしたストアド・プロシージャの中で、これを任意の名前に変更できます。たとえば、アプリケーションにより、名前を `scott/minutes.txt` に変更できます。

アップロード・ファイルの属性（MIME タイプ）の指定

フォームのアクション・ターゲットであるストアド・プロシージャは、アップロード・ファイルの名前を変更する以外にも、ファイルに関するその他の属性を変更できます。たとえば、1-16 ページの「[ファイルのアップロード](#)」に示した例のフォームでは、アップロードするドキュメントの MIME タイプをユーザーが入力するフィールドとして表示できます。

MIME タイプは、`write_info` のパラメータとして受信できます。その場合、ドキュメント表には、ファイルのアップロード時に PL/SQL ゲートウェイが `multipart` フォームから解析したデフォルトの MIME タイプではなく、そのドキュメントの MIME タイプが格納されます。

複数のファイルのアップロード

1 回の送信アクションで複数のファイルをアップロードする場合、アップロード・フォームに複数の `<INPUT type="file" name="file">` エレメントが含まれている必要があります。複数の `INPUT` エレメントで、`name` に同じ名前を定義する場合、アクション・プロシージャでパラメータ名を `owa.vc_arr` 型として宣言する必要があります。ファイルの `INPUT` エレメントに一意の名前を定義することも可能で、その場合、アクション・プロシージャでそれぞれを `varchar2` として宣言する必要があります。たとえば、フォームに次のエレメントが含まれているとします。

```
<INPUT type="file" name="textfiles">
<INPUT type="file" name="textfiles">
<INPUT type="file" name="binaryfile">
```

この場合、アクション・プロシージャに次のパラメータを含める必要があります。

```
procedure handle_text_and_binary_files(textfiles IN owa.vc_arr,
binaryfile IN varchar2).
```

ファイルのダウンロード

ファイルをデータベースにアップロードした後、それらのファイルのダウンロード、データベースからの削除および属性の読み込みと書き込みが可能になります。

ファイルをダウンロードするには、ダウンロードを開始するために `wpg_docload.download_file(file_name)` をコールする、パラメータを持たないストアド・プロシージャを作成します。ドキュメントのダウンロード・パッケージは、PL/SQL Web Toolkit に含まれています。詳細は、2-2 ページの「[必須パッケージのインストール](#)」を参照してください。

ユーザーに表示される HTML ページには、ドキュメント・アクセス・パスを含み、ダウンロードするファイルを指定する URL へのリンクが示されます。

たとえば、webview という DAD のドキュメント・アクセス・パスに docs、ドキュメント・アクセス・プロシージャに webview.process_download が指定されている場合、ユーザーが次のような URL をクリックすると、webview.process_download プロシージャがコールされます。

```
http://www.acme:9000/pls/webview/docs/myfile.htm.
```

次に、process_download のインプリメンテーションの例を示します。

```
procedure process_download is
  v_filename varchar2(255);
begin
  -- getfilepath() uses the SCRIPT_NAME and PATH_INFO cgi
  -- environment variables to construct the full pathname of
  -- the file URL, and then returns the part of the pathname
  -- following '/docs/'
  v_filename := getfilepath;
  select name into v_filename from plsql_gateway_doc
  where UPPER(name) = UPPER(v_filename);
  -- now we call docload.download_file to initiate
  -- the download.
  wpg_docload.download_file(v_filename);
exception
  when others then
    v_filename := null;
end process_download;
```

ゲートウェイで実行中のプロシージャから wpg_docload.download_file(filename) をコールすると、ファイル filename のダウンロードが開始されます。ただし、ファイルのダウンロードが開始されると、プロシージャによって生成される HTML (HTTP インタフェースによって生成される) がブラウザに返されないという制約があります。

PL/SQL ゲートウェイは、ドキュメント表からそのファイルのファイル名を探します。ドキュメント表内には、NAME 列がファイル名に一致する一意の行が存在する必要があります。PL/SQL ゲートウェイは、ドキュメント表の MIME_TYPE 列の情報に基づき、該当する HTTP レスポンス・ヘッダーを生成します。CONTENT_TYPE 列の値により、ドキュメントの内容がどの CONTENT 列から取得されるかが決まります。ドキュメントの内容は、HTTP レスポンスの本体として送信されます。

パスのエイリアシング

パスのエイリアシングを使用すると、PL/SQL ゲートウェイを使用するアプリケーションで、単純な URL を使用した直接オブジェクト・リファレンスを実現できます。PL/SQL ゲートウェイにより、ドキュメント・アクセス・パスおよびドキュメント・アクセス・プロシージャを使用して、アプリケーション内から直接ドキュメントにアクセスすることが可能です。

たとえば、次の URL の docs キーワードは、このリクエストがドキュメントにアクセスするものであることを PL/SQL ゲートウェイに知らせます。

```
http://<HostName>[:Port]/<DADName>/docs/<FolderName>/Document>
```

ここでは、ドキュメント・アクセス・パスが docs であることを前提としています。

パスのエイリアシングを使用すると、ドキュメント以外のアプリケーション・オブジェクトへ直接アクセスすることも可能で、その場合も同様に機能します。パスのエイリアシングは、Database Access Descriptor の設定情報の次の 2 つのフィールドでサポートされます。

- Path Alias
- Path Alias Procedure

PL/SQL ゲートウェイが受信した URL に「**Path Alias**」フィールドのキーワードが含まれていると、「**Path Alias Procedure**」フィールドに入力されたプロシージャが実行されます。

たとえば、次の URL を受信したとします。

```
http://www.acme.com:9000/portal_DAD/URL/path_alias_URL
```

Path Alias が URL の場合、PL/SQL ゲートウェイは **Path Alias Procedure** を実行し、キーワード URL より後の部分をすべて実行したプロシージャに渡します。

パスのエイリアシングを使用するアプリケーションで、**Path Alias Procedure** をインプリメントする必要があります。このプロシージャにより、キーワード URL より後の URL (path_alias_URL) をすべて 1 つのパラメータとして受け取ります。また、このプロシージャには、URL からオブジェクト・リファレンスを解除する機能が必要で、その機能を完全に実行する必要があります。

このプロシージャの名前と場所に制約はありませんが、データ型が varchar2 のパラメータ p_path のみ受け付けることが可能です。

キャッシング

PL/SQL Web アプリケーションのパフォーマンスを改善するために、PL/SQL ゲートウェイのキャッシュ機能を利用できます。この機能により、PL/SQL プロシージャの Web コンテンツを中間層の PL/SQL ゲートウェイにキャッシュできます。そのコンテンツに対するその後のリクエストはキャッシュから取得し、データベースの作業負荷を軽減できます。この場合、データベースからの妥当性チェックを使用することも使用しないことも可能です。キャッシュを使用可能にすると、Web アプリケーションのスケーラビリティが向上します。

概要

HTTP のプロトコルには、多数のキャッシュ・メカニズムが存在します。この項では、既存の方法の概要を示します。

HTTP プロトコルは、リクエストとレスポンスで構成されます。Web ブラウザなどのユーザー・エージェントは、リクエスト・ヘッダーでメタデータを指定できます。

PL/SQL プロシージャなどのコンテンツ・プロバイダは、1つ以上の HTTP レスポンス・ヘッダーを使用して、キャッシュ制御のメタデータを指定できます。その後の HTTP リクエストでは、このメタデータはユーザー・エージェントからコンテンツ・プロバイダに渡され、これによりコンテンツ・プロバイダは、ユーザー・エージェントのキャッシュ・エントリが有効であるかどうかを判断できます。

Expires レスポンス・ヘッダーなどの場合、メタデータにより、そのコンテンツが一定期間有効であることが示されます。その後のリクエストは、その期間が経過するまで行われる必要がありません。コンテンツ・プロバイダはこの期間アクセスする必要があることを示していますが、ユーザー・エージェントがアクセスすることは可能です。

妥当性チェック方式 最初に Web ページが生成されたときに、Web ページに **Last-Modified** レスポンス・ヘッダーが含まれます。このヘッダーには、コンテンツが要求された日付がサーバーに対する相対日付で表示されます。キャッシュ機能を持つユーザー・エージェントの場合、コンテンツとともにこの日付情報が保存されます。それ以降この Web ページの URL が要求されたときに、ユーザー・エージェントは次の処理を行います。

- キャッシュされたバージョンが存在するかどうかを調べます。
- 日付情報を抽出します。
- リクエスト・ヘッダー **If-Modified-Since** を生成します。
- 最後に、コンテンツ・プロバイダにリクエストを送信します。

キャッシュを使用可能なコンテンツ・プロバイダは、**If-Modified-Since** ヘッダーを探し、コンテンツの日付とその日付を比較します。2つが一致すれば、"HTTP/1.1 304 Not Modified" などの HTTP レスポンス・ステータス・ヘッダーが生成され、コンテンツは送信されません。このステータス・コードを受信した場合、キャッシュ・エントリが検証されたことになり、ユーザー・エージェントはそのキャッシュ・エントリを再利用できます。

2つが一致しない場合、"HTTP/1.1 200 OK" などの HTTP レスポンス・ヘッダーが生成され、新しい **Last-Modified Response** ヘッダーとともに新しいコンテンツが送信されます。このステータス・コードを受信した場合、ユーザー・エージェントはキャッシュ・エントリを新しいコンテンツと新しい日付情報に置換する必要があります。

HTTP プロトコルには、**ETag** (Entity Tag) レスポンス・ヘッダーとリクエスト・ヘッダーという別の妥当性チェックの方法が存在します。このヘッダーの値は、ユーザー・エージェントから参照可能な文字列です。コンテンツ・プロバイダは、アプリケーション・タイプに基づき、この文字列を生成します。これは、日付の値のみ含められる **If-Modified-Since** ヘッダーよりも一般的な妥当性チェックの方法です。

ETag 方式は、日付方式と非常に似ています。コンテンツ・プロバイダは、レスポンス・ヘッダーの一部として、**ETag** ヘッダー値を生成します。ユーザー・エージェントは、受信したコンテンツとともにこの参照可能なヘッダー値を格納します。このコンテンツに対する次のリクエストを受信したときに、ユーザー・エージェントは格納済みの参照可能な値とともに **If-Match** ヘッダーをコンテンツ・プロバイダに渡します。この参照可能な値はコンテンツ・プロバイダによって生成されたため、コンテンツ・プロバイダは、ユーザー・エージェント

に何を返すべきかを判断できます。残りは前述の **Last-Modified** 妥当性チェックの方法と全く同じです。

期限方式 Web ページに Expires レスポンス・ヘッダーが含まれている場合、ユーザー・エージェントはこの日付の値と Date レスポンス・ヘッダーを組み合わせて、レスポンスの有効期限を判断できます。これによって妥当性チェックの基準が設定されるため、この期間中はコンテンツ・プロバイダと通信する必要はありません。そのため、ユーザー・エージェントはそのリクエストに対し、キャッシュされたコンテンツを直接送り返すことが可能です。

PL/SQL ゲートウェイのキャッシュ

設計の基本に HTTP プロトコルを使用しているため、PL/SQL ゲートウェイをユーザー・エージェント、PL/SQL プロシージャをコンテンツ・プロバイダであると考えられます。この場合も、HTTP と同様に、ユーザー・エージェントとコンテンツ・プロバイダ間の通信メカニズムはヘッダーと環境変数です。

前提の 1 つに、キャッシュされる物理 URL は異なるユーザー間で同じであっても、キャッシュされるコンテンツは多様で、通常はユーザーごとに保護されているという点があります。さらに、コンテンツが異なる言語である可能性もあります。これらの前提を満たすため、URL のみを使用してキャッシュ・キーを作成する HTTP/1.1 プロトコルとは若干設計が異なります。PL/SQL ゲートウェイでは、URL とともにユーザーおよび言語を使用して、キャッシュ・キーを作成します。

各リクエストごとに、2 つのレベルのキャッシュが存在します。

- **ユーザー・レベルのキャッシュ**は、ログインした特定のユーザー用です。格納されたキャッシュは、そのユーザー専用です。そのキャッシュを使用できるのは、そのユーザーのみです。
- **システム・レベルのキャッシュ**は、そのキャッシュを共有するユーザーのグループ用です。

たとえば、カスタマイズ可能な PL/SQL Web アプリケーションをカスタマイズしたユーザーがいない場合、アプリケーションの出力はシステム・レベルのキャッシュに格納できます。そのため、システム上の全ユーザーについて、1 つのキャッシュ・コピーのみ存在します。

ただし、いずれかのユーザーがアプリケーションをカスタマイズすると、そのユーザーについてのみ、新しいユーザー・レベルのキャッシュが格納されます。他のすべてのユーザーはシステム・レベルのキャッシュを使用します。これに関しては、1-25 ページの「[システム・レベルとユーザー・レベルのキャッシュ](#)」で詳細に説明します。

owa_cache パッケージ owa_cache パッケージには、特別なキャッシュ・ヘッダーと環境変数を設定および取得するためのファンクションとプロシージャが含まれています。これにより、開発者は PL/SQL ゲートウェイ・キャッシュをより簡単に使用できます。このパッケージは、すでにデータベースにインストールされています（詳細は、2-2 ページの「[必須パッケージのインストール](#)」を参照してください）。

owa_cache パッケージの完全な仕様は、5-6 ページの「[owa_cache パッケージ](#)」を参照してください。

次に、コールされる主なファンクションを示します。

- owa_cache.set_cache(p_etag IN varchar2, p_level IN varchar2)
このファンクションは、キャッシュの妥当性チェック・モデルのヘッダーを設定します。p_etag パラメータは、生成されたコンテンツのタグを作成する文字列です。p_level パラメータは、使用するキャッシュ・レベルです。
- owa_cache.set_expires(p_expires IN number, p_level IN varchar2)
このファンクションは、キャッシュの期限モデルのヘッダーを設定します。p_expires パラメータは、生成されたコンテンツが有効な時間（分）です。p_level パラメータは、使用するキャッシュ・レベルです。
- owa_cache.set_not_modified
このファンクションは、妥当性チェック・モデルの場合のみ有効です。キャッシュ済みのコンテンツを使用するようにゲートウェイに通知するヘッダーを設定します。
- owa_cache.get_level
このファンクションは、妥当性チェック・モデルの場合のみ有効です。キャッシュ・レベル ("USER" または "SYSTEM") を取得します。
- owa_cache.get_etag
このファンクションは、妥当性チェック・モデルの場合のみ有効です。キャッシュに入っているコンテンツに関連付けられたタグを取得します。

妥当性チェック・モデル このモデルは、HTTP の ETag キャッシュ方式と非常に似ています。そのため、PL/SQL ゲートウェイは、コンテンツが変更されたかどうかを毎回 PL/SQL プロシージャに問い合わせます。

PL/SQL プロシージャが PL/SQL ゲートウェイを通じて初めてコールされたとします。PL/SQL ゲートウェイはプロシージャを実行し、通常の CGI 環境変数を渡します。プロシージャはコンテンツを生成し、それを返します。生成されたコンテンツがキャッシュ可能であるとプロシージャが判断した場合、owa_cache プロシージャをコールして、タグとキャッシュ・レベルを次のように設定します。

```
owa_cache.set_cache(p_etag, p_level);
```

各項目の説明：

p_etag は、コンテンツにタグを設定するためにプロシージャが生成する文字列です。

p_level は、キャッシュ・レベルです（システム・レベルの場合は "SYSTEM"、ユーザー・レベルの場合は "USER"）。

`set_cache` ファンクションは、必要なヘッダーを設定し、返されたコンテンツがキャッシュ可能であることを PL/SQL ゲートウェイに通知します。これにより、PL/SQL ゲートウェイはコンテンツをブラウザに送信する際に、そのコンテンツをタグおよびキャッシュ・レベル情報とともにローカル・ファイル・システムにキャッシュします。

次に、同じ PL/SQL プロシージャに対する 2 回目のリクエストを受信したとします。PL/SQL ゲートウェイは、そのリクエストのコンテンツがキャッシュされていることを検出します。この場合、特殊な手順を実行します。前回同じプロシージャを実行した際に取得したタグおよびキャッシュ・レベル情報を CGI 環境変数の一部として渡します。すると、プロシージャはこれらのキャッシュ CGI 環境変数を使用して、コンテンツが変更されているかどうかをチェックします。そのために、次の `owa_cache` ファンクションをコールします。

```
owa_cache.get_etag;  
  
owa_cache.get_level;
```

これらのファンクションは、それぞれタグおよびキャッシュ・レベルを取得します。PL/SQL プロシージャはこれらを前回生成しているため、コンテンツを再生成する必要があるかどうかを判断するために、任意の処理を行います。

コンテンツが同じ場合、プロシージャは、次の `owa_cache` プロシージャをコールします。

```
owa_cache.set_not_modified;
```

この場合はコンテンツを生成しません。これにより、PL/SQL ゲートウェイに、このリクエストについてはキャッシュに入っているコンテンツを使用するよう知らせます。そして、キャッシュ内のコンテンツが直接ブラウザに送信されます。

一方、コンテンツが変更されたら PL/SQL プロシージャによって判断された場合、新しいコンテンツおよび新しいタグとキャッシュ・レベルが生成されます。PL/SQL ゲートウェイのコンテンツのコピーは失効しているため、`owa_cache.set_not_modified` はコールされません。かわりに、PL/SQL ゲートウェイは失効したキャッシュ内のコピーを新しく生成されたものに置き換え、関連付けられているタグとキャッシュ・レベル情報を更新します。

期限モデル 妥当性チェック・モデルを使用した場合、PL/SQL ゲートウェイは、キャッシュのコンテンツが使用可能かどうかを判断するため、毎回 PL/SQL プロシージャに対する問合せが発生します。期限モデルでは、プロシージャはコンテンツの有効期限を先に設定します。これにより、PL/SQL ゲートウェイは、プロシージャに問い合わせずに、キャッシュのコンテンツを使用できます。データベースとの対話処理が必要ないため、これによってパフォーマンスが一層向上します。

前述の妥当性チェック・モデルと同じ状況で、プロシージャがキャッシュに期限モデルを使用しているとします。いったんコンテンツを生成すると、プロシージャは次の `owa_cache` プロシージャをコールします。

```
owa_cache.set_expires(p_expires, p_level);
```

各項目の説明：

p_expires は、コンテンツの有効期間（分）です。

`p_level` はキャッシュ・レベルです。

`set_expires` プロシージャにより、期限モデルのキャッシュが使用されていることを PL/SQL ゲートウェイに通知する適切なヘッダーが設定されます。その後、PL/SQL ゲートウェイはコンテンツとともに、有効期限およびキャッシュ・レベル情報をファイル・システムにキャッシュします。

次に、ブラウザを通じて同じプロシージャが再び実行されたとします。PL/SQL ゲートウェイは、期限ベースのコンテンツのコピーがキャッシュに入っていることを検出し、次に、現在の時刻とキャッシュ・ファイルが作成された時刻の違いを調べることで、妥当性をチェックします。この違いが有効期間内であれば、キャッシュ内のコピーは有効であり、データベースとの対話処理を行わずにこのコピーが送信されます。

違いが有効期間内ではない場合、キャッシュ内のコピーは失効しています。この場合、PL/SQL ゲートウェイはプロシージャを実行します。プロシージャは再び期限ベースのキャッシュを使用すべきかどうかを決定します。かわりに妥当性チェック・モデルのキャッシュを使用したり、キャッシュをまったく使用しない場合もあります。

システム・レベルとユーザー・レベルのキャッシュ

PL/SQL プロシージャは、生成されたコンテンツがシステム・レベルであるか、またはユーザー・レベルであるかを判断します。これにより、複数のユーザーが同じコンテンツを参照している場合、PL/SQL ゲートウェイのキャッシュに格納する重複するファイルを減らすことが可能です。

PL/SQL プロシージャは、生成されたコンテンツがシステム・レベルでキャッシュ可能なコンテンツであるか、またはユーザー・レベルであるかを判断します。

- システム・レベルのコンテンツの場合、プロシージャは文字列 "SYSTEM" をキャッシュ・レベル・パラメータとして `owa_cache` ファンクション（妥当性チェック・モデルの場合は `set_cache`、期限モデルの場合は `set_expires`）に渡します。
- ユーザー・レベルのコンテンツの場合、文字列 "USER" をキャッシュ・レベルのパラメータとして渡します。

PL/SQL ゲートウェイのシステム・レベルのキャッシュとユーザー・レベルのキャッシュの違いは、ユーザー情報の使用方法にあります。システム・レベルのキャッシュの場合、キャッシュは複数のユーザーによって使用されるため、ユーザー情報は使用されません。そのため、ユーザー情報は、システム・レベルのキャッシュ・ヒットの基準ではありません。

ユーザー・レベルのキャッシュの場合、ユーザー情報はユーザー・レベルのキャッシュ・ヒットの基準です。ユーザー・レベルのキャッシュは、常にシステム・レベルのキャッシュより優先されます。特定のユーザーについて、システム・レベルとユーザー・レベルのキャッシュのコピーの両方が存在する場合、ユーザー・レベルが使用されます。

CGI 環境変数

OWA_UTIL パッケージには、CGI 環境変数の値を取得するための API が付属しています。CGI 環境変数は、PL/SQL ゲートウェイによって実行されるプロシージャに一種のコンテキストを提供します。PL/SQL ゲートウェイは CGI によって処理されませんが、PL/SQL ゲートウェイから実行される PL/SQL アプリケーションは、これらの CGI 環境変数にアクセス可能です。

PL/SQL ゲートウェイには、次の CGI 環境変数が用意されています。

- REMOTE_USER
- DAD_NAME
- DOC_ACCESS_PATH
- PATH_INFO
- SCRIPT_NAME
- SERVER_PORT
- SERVER_NAME
- REQUEST_METHOD
- REMOTE_HOST
- REMOTE_ADDR
- SERVER_PROTOCOL
- HTTP_USER_AGENT
- HTTP_PRAGMA
- HTTP_HOST
- HTTP_ACCEPT
- HTTP_ACCEPT_ENCODING
- HTTP_ACCEPT_LANGUAGE
- HTTP_ACCEPT_CHARSET
- REQUEST_CHARSET (詳細は、1-27 ページの「[REQUEST_CHARSET CGI 環境変数](#)」を参照してください)
- REQUEST_IANA_CHARSET
- DOCUMENT_TABLE (詳細は、1-14 ページの「[document_table \(document_table_name\)](#)」を参照してください)
- AUTHORIZATION

- PATH_ALIAS
- SCRIPT_PREFIX
- REQUEST_PROTOCOL
- HTTP_COOKIE

PL/SQL アプリケーションは、owa_util.get_cgi_env インタフェースを使用して CGI 環境変数の値を取得できます。

構文:

```
owa_util.get_cgi_env(param_name in varchar2) return varchar2;
```

各項目の説明:

param_name は、CGI 環境変数の名前です。param_name には大文字・小文字の区別があります。

NLS

mod_plsql では、次の制約のうちいずれかが適用されます。

- データベースの NLS_LANG パラメータが Oracle HTTP Server (powered by Apache) のパラメータと一致する必要がある。
- データベースおよび Oracle HTTP Server (powered by Apache) の NLS_LANG パラメータの文字幅が固定であり、両方同じサイズである必要がある。

REQUEST_CHARSET CGI 環境変数

PL/SQL ゲートウェイに対するリクエストは、すべて DAD に関連付けられます。CGI 環境変数 REQUEST_CHARSET は、次の規則に従って設定されます。

- PL/SQL ゲートウェイのグローバル設定情報の一部に NLS_LANG が指定されると、REQUEST_CHARSET CGI 環境変数は、グローバルな NLS_LANG パラメータのキャラクタ・セット部分に設定されます。
- 上記以外の場合、REQUEST_CHARSET は、使用中のデフォルト・キャラクタ・セットに設定されます。
 - 埋込みゲートウェイの場合、これはデータベースのデフォルト・キャラクタ・セットです。
 - 中間層で運用されているゲートウェイの場合 (WebDB リスナーまたは Oracle HTTP Server の一部として)、これは WebDB リスナー・プロセスの NLS_LANG 環境変数から導出されたキャラクタ・セット情報になります。

PL/SQL アプリケーションは、フォームのファンクション・コールによってこの情報にアクセスできます。

```
owa_util.get_cgi_env('REQUEST_CHARSET');
```

REQUEST_IANA_CHARSET CGI 環境変数

これは、REQUEST_CHARSET CGI 環境変数に相当する IANA (Internet Assigned Number Authority) の環境変数です。IANA は、インターネットで使用されるキャラクタ・セットの標準をグローバルに調整する機関です。

PL/SQL ゲートウェイのインストール

システム要件

次に、PL/SQL ゲートウェイのインストールおよび実行のための推奨および最低条件を示します。

オペレーティング・システム

- Windows NT 4.0 (Service Pack 3 以上を適用)
- Solaris 2.6 以上
- IBM AIX 4.3.2/4.3.3
- Compaq Tru64 4.0d
- Solaris Intel 2.7

Oracle データベース

- Oracle8i (リリース 8.1.6 または 8.1.7)

注意: PL/SQL ゲートウェイでは、Oracle 8.1.7 クライアント・ライブラリが PL/SQL ゲートウェイと同じ Oracle ホームにインストールされている必要があります。これらのライブラリがインストールされている場合、PL/SQL ゲートウェイをリモートの Oracle 8.0.5 以上のデータベースに対して実行することが可能です。たとえば、PL/SQL ゲートウェイを使用して、リモートの 8.0.5 データベースにインストールされている PL/SQL プロシージャを実行できます。

Web リスナー

- Solaris の場合 - Oracle HTTP Server (powered by Apache) 1.3.12 for Oracle9i Application Server リリース 1.0.2
- Windows NT の場合 - Oracle HTTP Server (powered by Apache) 1.3.12 for Oracle9i Application Server リリース 1.0.2

Web ブラウザ

- Netscape 4.0.8 以上
- Microsoft Internet Explorer 4.0.1 (Service Pack 1 以上を適用)

インストールの前に

Oracle9i Application Server R1.0.2 の Oracle Universal Installer を使用して PL/SQL ゲートウェイをインストールする前に、次の前提条件を満たしている必要があります。

- PL/SQL ゲートウェイに必要な PL/SQL Web Agent パッケージをロードするデータベースの SYS ユーザー・パスワードが必要です。
- PL/SQL ゲートウェイを接続するデータベースが稼動している必要があります。
- Oracle Universal Installer を実行するマシンに十分なディスク領域が必要です。
- Oracle Universal Installer が oraInventory データを書き込むディレクトリの書込み許可が必要です。

インストール

Oracle Universal Installer を開始するには、製品 CD またはステージング・エリアに入っている runInstaller アプリケーションを実行します。Oracle9i Application Server R1.0.2 のインストール先ディレクトリの選択を含め、インストレーション・アプリケーションの各ステップの手順に従います。このインストール・ディレクトリは、選択後は <ORACLE_HOME> と呼ばれます。

必須パッケージのインストール

インストール後に、owaload.sql スクリプトを使用して、追加の必須パッケージを手動でインストールする必要があります。

1. owaload.sql ファイルが入っているディレクトリに移動します。このディレクトリは、<ORACLE_HOME>/Apache/modplsql/owa です。
2. SQL*Plus を使用して、SYS ユーザーで Oracle データベースにログインします。
3. SQL プロンプトで、次のコマンドを実行します。

```
@owaload.sql log_file
```

各項目の説明：

log_file は、インストール・ログ・ファイルです。

owaload.sql は、SYS スキーマに PL/SQL Web Toolkit パッケージをインストールします。また、データベースの全ユーザーがアクセスできるよう、パブリック・シノニムを

作成してパッケージをパブリックにします。このため、1つのデータベースにつき1回のインストールのみ必要です。

PL/SQL Web Toolkit パッケージ

Oracle 8.1.7 および Oracle Internet Application Server 8i R1.0.1 以降には、追加機能を持つ新しい一連の PL/SQL Web Toolkit パッケージが用意されています。

- Oracle 8.1.7 のインストールまたはアップグレードにより、新しい PL/SQL Web Toolkit パッケージが自動的に SYS スキーマにインストールされます。
- Oracle Internet Application Server 8i R1.0.1 以降では、`mod_plsql` のユーザーはこれらのパッケージを SYS スキーマに手動でインストールする必要があります。
- Oracle 8.1.7 をインストールする場合、新しい PL/SQL Web Toolkit パッケージは、`$ORACLE_HOME/rdbms/admin` ディレクトリに入っています。
- Oracle Internet Application Server 8i R1.0.1 以降をインストールする場合、新しい PL/SQL Web Toolkit パッケージは、`$IAS_HOME/Apache/modplsql/owa` ディレクトリに入っています。

Oracle Portal 3.0 は、新しい PL/SQL Web Toolkit パッケージに依存しているため、注意してください。

Oracle Application Server 以外のインストール環境 インストール環境に Oracle Application Server が存在せず、以前は Oracle Internet Application Server 8i または WebDB リスナー 2.5 以下を実行していた場合、古い PL/SQL Web Toolkit パッケージがインストールされていたスキーマを削除します。SYS にインストールされた新しい PL/SQL Web Toolkit パッケージが使用されます。複数の PL/SQL Web Toolkit パッケージがインストールされていると、問題が発生します。スキーマを削除する前に、スキーマ内にユーザー・データ（PL/SQL Web Toolkit パッケージ以外のもの）が存在しないことを確認してください。

Oracle Application Server のインストール環境 Oracle 8.1.7 および Oracle9i Application Server R1.0.2 に付属の新しい PL/SQL Web Toolkit パッケージでは多くの点が改善されているため、これをインストールすることをお薦めします。

すでに新しい PL/SQL Web Toolkit パッケージがインストールされ（自動的に、または手動で）、以前に Oracle Application Server を実行していた場合、Oracle 8.1.7 のインストールまたはアップグレード、あるいは `mod_plsql` の手動インストールにより、新しい PL/SQL Web Toolkit パッケージが SYS スキーマに作成され、これらの新しいパッケージを参照する PL/SQL Web Toolkit のパブリック・シノニムが再作成されるため、注意が必要です。しかし、Oracle Application Server の PL/SQL カートリッジの実行時に問題が発生した場合、以前の PL/SQL Web Toolkit パッケージのパブリック・シノニムを再作成する必要があります。以前のパブリック・シノニムを再作成するには、次のステップを実行します。

1. SQL*Plus から、SYS として接続します。

2. 次の SQL 文を実行します。これにより、アップグレード・プロセス中に作成された PL/SQL Web Toolkit のパブリック・シノニムがすべて削除されます。

```
drop public synonym OWA_CUSTOM;  
drop public synonym OWA_GLOBAL;  
drop public synonym OWA;  
drop public synonym HTF;  
drop public synonym HTP;  
drop public synonym OWA_COOKIE;  
drop public synonym OWA_IMAGE;  
drop public synonym OWA_OPT_LOCK;  
drop public synonym OWA_PATTERN;  
drop public synonym OWA_SEC;  
drop public synonym OWA_TEXT;  
drop public synonym OWA_UTIL;  
drop public synonym OWA_INIT;  
drop public synonym OWA_CACHE;  
drop public synonym WPG_DOCLOAD;
```

3. 古い PL/SQL Web Toolkit パッケージのインストレーション・スキーマ（通常は OAS_PUBLIC）に接続します。
4. 次の SQL 文を実行します。これにより、アップグレード・プロセス中に変更された PL/SQL Web Toolkit のパブリック・シノニムが再作成され、以前の PL/SQL Web Toolkit パッケージを参照するようになります。

```
create public synonym OWA_CUSTOM for OWA_CUSTOM;  
create public synonym OWA_GLOBAL for OWA_CUSTOM;  
create public synonym OWA for OWA;  
create public synonym HTF for HTF;  
create public synonym HTP for HTP;  
create public synonym OWA_COOKIE for OWA_COOKIE;  
create public synonym OWA_IMAGE for OWA_IMAGE;  
create public synonym OWA_OPT_LOCK for OWA_OPT_LOCK;  
create public synonym OWA_PATTERN for OWA_PATTERN;  
create public synonym OWA_SEC for OWA_SEC;  
create public synonym OWA_TEXT for OWA_TEXT;  
create public synonym OWA_UTIL for OWA_UTIL;  
create public synonym OWA_INIT for OWA_CUSTOM;  
create public synonym OWA_CACHE for OWA_CACHE;  
create public synonym WPG_DOCLOAD for WPG_DOCLOAD;
```

新しい PL/SQL Web Toolkit パッケージがインストールされていない Oracle Application Server が存在し、Oracle9i Application Server も同時に使用する場合、新しい PL/SQL Web Toolkit パッケージをインストールすることをお勧めします。Oracle9i Application Server の mod_plsql を使用するために以前の PL/SQL Web Toolkit パッケージの使用を継続する場合、次の SQL 文を実行する必要があります（注意：これらの

SQL 文は新しい PL/SQL Web Toolkit パッケージの一部であり、新しい PL/SQL Web Toolkit パッケージをインストールしたことがなく、以前の PL/SQL Web Toolkit パッケージを引き続き使用する場合のみ必要となります。

1. SQL*Plus から、SYS として接続します。
2. 新しい PL/SQL Web Toolkit パッケージを探し、次のパッケージをインストールします。

```
wpiutl.sql
wpgdocs.sql
wpgdocb.sql
```

3. wpg_docload の execute 権限をパブリックに付与します。
4. wpg_docload のパブリック・シノニム wpg_docload を作成します。

これらのステップにより、mod_plsql を以前の PL/SQL Web Toolkit パッケージで実行するために必要なパッケージがインストールされます。この設定では、新しい PL/SQL Web Toolkit パッケージの新機能の一部は使用できません。

Oracle HTTP Server Listener の設定

Oracle9i Application Server のインストールにより、ユーザーが編集可能な設定ファイルが作成されます。これには、PL/SQL ゲートウェイに影響を与える次のファイルも含まれます。

apachectl ファイル

apachectl は、Solaris での Oracle HTTP Server の起動および停止に使用されます。次の場所に存在します。

<ORACLE_HOME>/Apache/Apache/bin/apachectl

このファイル内に、PL/SQL ゲートウェイに影響するパラメータが 3 つあります。

- ORACLE_HOME: PL/SQL ゲートウェイが稼動する Oracle ホーム。
デフォルト: <ORACLE_HOME>
- LD_LIBRARY_PATH: PL/SQL ゲートウェイが必要とする Oracle ライブラリ。これは、Oracle 8.1.7 クライアント・ライブラリがインストールされている場所を指している必要があります。このパラメータは、Solaris でのみ使用します。
デフォルト: <ORACLE_HOME>/lib
- WV_GATEWAY_CFG: PL/SQL ゲートウェイの設定ファイル。
Solaris のデフォルト: <ORACLE_HOME>/Apache/modplsql/cfg/wdbsvr.app
Windows NT のデフォルト: <APACHE_HOME>%modplsql%cfg%wdbsvr.app

Windows NT では、Oracle HTTP Server はサービスとしてインストールされます。このサービスにアクセスするには、「スタート」→「設定」→「コントロール パネル」→「システム」の順にクリックします。「環境」タブをクリックし、設定ファイルを指す WV_GATEWAY_CFG というシステム変数を作成してください。

注意 : PL/SQL ゲートウェイを別の Oracle ホームで実行する場合、必ず ORACLE_HOME および LD_LIBRARY_PATH 設定の両方を変更してください。

httpd.conf ファイル

この設定ファイルは、Oracle HTTP Server powered by Apache の動作を定義します。ポート番号およびその他のサーバー設定が設定可能です。次の場所に存在します。

<ORACLE_HOME>/Apache/Apache/conf/httpd.conf

plsql.conf ファイル

この設定ファイルでは、PL/SQL ゲートウェイ・モジュールの設定を指定します。次の場所に存在します。

<ORACLE_HOME>/Apache/modplsql/cfg/plsql.conf

次の設定が設定可能です。

- LoadModule plsql_module <MOD_PATH>: PL/SQL ゲートウェイ・モジュールの場所。
Solaris のデフォルト : <ORACLE_HOME>/Apache/modplsql/bin/modplsql.so
Windows NT のデフォルト : %ORACLE_HOME%\bin\modplsql.dll
- Location <MOUNT_PATH>: PL/SQL ゲートウェイが起動される URL 内の接頭辞。
デフォルト : /pls

wdbsvr.app ファイル

この設定ファイルは、PL/SQL ゲートウェイ・モジュールの設定を指定します。次の場所に存在します。

<ORACLE_HOME>/Apache/modplsql/cfg/wdbsvr.app

これは、PL/SQL ゲートウェイのメインの設定ファイルです。ここにすべての DAD 情報が含まれています。このファイルは、直接編集しないでください。必ず PL/SQL ゲートウェイの設定ページを使用して編集してください。PL/SQL ゲートウェイの設定ページには、次のようにしてブラウザからアクセスできます。

PL/SQL ゲートウェイの設定ページへのアクセス

PL/SQL ゲートウェイの設定ページにアクセスするには、Web ブラウザで次の URL を入力します。

```
http://<hostname>:<port>/pls/DAD/<admin_path>/gateway.htm
```

各項目の説明

<hostname> は、アプリケーション・サーバーが稼動しているマシンです。

<port> には、アプリケーション・サーバーがリスニングしているポートを指定します。指定しない場合、ポート 80 が使用されます。

<admin_path> には、admin ページを識別する URL のパス・エレメントを指定します。デフォルトは admin_ です。たとえば、admin_path が /admin_/ の場合、次の URL により PL/SQL ゲートウェイの設定ページが実行されます。ただし、実行しているユーザーが管理者の設定リストに含まれている必要があります。

```
http://www.myserver.com/pls/admin_/gateway.htm
```

設定は、管理セキュリティ設定によって保護されています。Web 管理ページは、設定ファイルの管理者設定に名前が含まれているユーザーによってのみ実行されます。詳細は、3-1 ページの「[PL/SQL ゲートウェイの設定](#)」を参照してください。

plsql.conf 設定ファイル

Oracle HTTP Listener の設定ファイルには、mod_plsql 設定ファイルである plsql.conf が含まれています。plsql.conf の内容は、次のとおりです。

```
#
# Directives added for the PL/SQL Gateway
#
LoadModule plsql_module %APACHE_HOME%/modplsql/bin/modplsql.so

#
# Enable handling of all virtual paths beginning with "/pls" by mod-plsql
#
<Location /pls>
    SetHandler pls_handler
    Order deny,allow
    Allow from all
</Location>
```

Oracle HTTP Server Listener の起動と停止

Apache リスナーを起動するには、次の行を入力します。

```
<ORACLE_HOME>/Apache/Apache/bin/httpdctl start
```

Apache リスナーを SSL サポートで起動するには、次の行を入力します。

```
<ORACLE_HOME>/Apache/Apache/bin/httpdctl startssl
```

Apache リスナーを停止するには、次の行を入力します。

```
<ORACLE_HOME>/Apache/Apache/bin/httpdctl stop
```

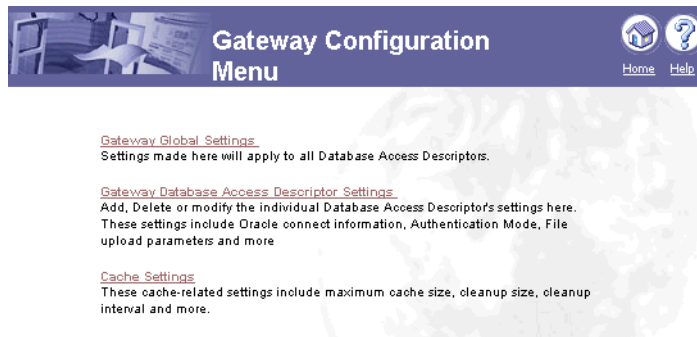
PL/SQL ゲートウェイの設定

PL/SQL ゲートウェイには、Database Access Descriptor (DAD) を設定するための Web ページが用意されています。DAD は、HTTP リクエストを満たすために PL/SQL ゲートウェイがデータベース・サーバーに接続する方法を指定する値の集合です。

全般設定

「Gateway Configuration Menu」にアクセスする URL は、次のとおりです。

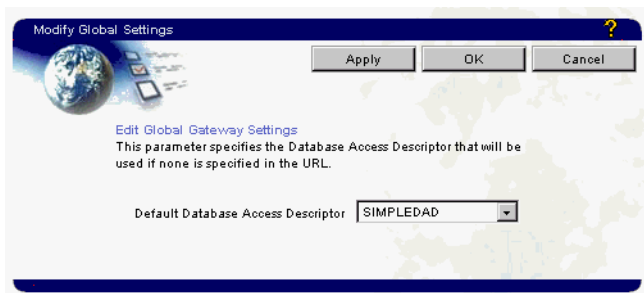
`http://<hostname>:<port>/pls/admin_/gateway.htm`



このページから、「**Gateway Global Settings**」リンクをクリックして、「Global Gateway Settings」ページにアクセスします。または、直接「Global Gateway Settings」ページにアクセスするには、次の URL を使用します。

`http://<hostname>:<port>/pls/admin_/globalsettings.htm`

次に、「Global Gateway Settings」ページを示します。



Global Gateway Settings Default Database Access Descriptor (DAD)

デフォルトの DAD を指すパスを指定します。エンド・ユーザーが DAD 名を指定せずに URL を入力すると、デフォルトの DAD のホーム・ページが表示されます。

デフォルト = なし。DAD 名を変更する場合は、このフィールドに新しい DAD 名を入力します。

Administrators

管理ページを表示できるユーザーを指定します。デフォルトでは「ALL」が設定されており、誰でも管理ページを表示できます。管理ページのセキュリティを強化するためには、これを scott, mike のようにカンマ区切りのユーザー・リストに変更する必要があります。この例で、scott と mike はローカル・データベース・ユーザー名です。または、scott, mike@orcl のようにします。この場合、orcl はリモート・データベースの接続文字列です。

注意: この設定は、設定ファイルでのみアクセス可能です。PL/SQL ゲートウェイの Web ページからはアクセスできません。

Admin path

管理ページを識別する URL のパス・エレメントを指定します。これは、通常は変更せずに、/admin_/ のままにしておきます。

注意: この設定は、設定ファイルでのみアクセス可能です。PL/SQL ゲートウェイの Web ページからはアクセスできません。

Database Access Descriptor の設定

「Database Access Descriptor Settings」ページにアクセスするには、PL/SQL ゲートウェイの設定ページの「**Gateway Database Access Descriptor Settings**」リンクをクリックするか、または次の URL から直接アクセスします。

`http://<hostname>:<port>/pls/admin_/dadentries.htm`

次に、「Database Access Descriptor Settings」ページの一部を示します。



Database Access Descriptor Settings

Database Access Descriptor Name

この DAD の名前が表示されます。この名前は、インストール中または新しい Web サイトの作成中に設定されます。名前を変更する場合は、このフィールドに新しい名前を入力します。

注意: 次のいずれも実行しない場合、**#none#** と入力します。

Database Access Descriptor の名前の設定

対応する Global Settings パラメータの値の継承

| | |
|----------------------------|---|
| Oracle User Name | Oracle データベース・アカウントのユーザー名が表示されます。ユーザー名は、通常、インストール中または新しい Web サイトの作成中に設定されます。名前を変更する場合は、この入力フィールドに新しい名前を入力します。 |
| Oracle Password | <p>Oracle データベース・アカウントのパスワードが表示されます。パスワードは、通常、インストール中に設定されますが、この入力フィールドに新しいパスワードを入力することによって変更できます。</p> <p>注意：「Oracle User Name」と「Oracle Password」は、Web サイトまたはページにログインするためのデフォルトのユーザー名とパスワードです。「Oracle User Name」と「Oracle Password」入力フィールドを空白のままにした場合、ユーザーは初めてログインするときにユーザー名とパスワードを入力するよう求められます。</p> |
| Oracle Connect String | リモート・データベースを使用している場合、SQL*Net の別名を入力します。データベースがローカルの場合はこのフィールドを空白のままにしておきます。 |
| Authentication Mode | <p>このパラメータには、次の値のうちいずれかを設定できます。</p> <ul style="list-style-type: none">■ Basic - 基本 HTTP 認証を使用して認証が実行されます。ほとんどのアプリケーションでは Basic 認証を使用します。■ Global Owa - 認証は、OWA パッケージ・スキーマで実行されます。■ Custom Owa - 認証は、ユーザーのスキーマ、またはそれが見つからない場合は OWA パッケージ・スキーマ内のパッケージおよびプロシージャを使用して実行されます。■ Per Package - 認証は、ユーザーのスキーマ内のパッケージおよびプロシージャによって実行されます。■ Single Sign-On - 認証は、Login Server の Oracle シングル・サインオン機能を使用して実行されます。このモードは、Login Server でアプリケーションを使用するよう設定されている場合のみ使用可能です。 |
| Session Cookie Name | 分散環境で使用される Oracle Portal 3.X の場合のみ、セッション Cookie 名を入力します。 |
| Create a Stateful Session? | 各データベース・リクエストについてデータベース・パッケージ / セッションの状態を維持する場合、「Yes」を選択します。各リクエストの後リセットする場合、「No」を選択します。PL/SQL ゲートウェイでは、このパラメータは「No」に設定する必要があります。 |

| | |
|---------------------------------------|---|
| Enable Connection Pooling? | <p>1つの URL リクエストの処理後に、その後のリクエスト処理のためにデータベース接続をオープンにしておくかどうかを選択します。ほとんどの設定では、最大のパフォーマンスを最高にするために「Yes」を指定します。</p> <p>PL/SQL ゲートウェイのクリーンアップ・スレッドは、15 分間使用されなかったデータベース・セッションをクリーンアップします。</p> |
| Default (Home) Page | <p>URL の一部として PL/SQL プロシージャが指定されていない場合に実行される PL/SQL プロシージャを入力します。たとえば、myapp.home というデフォルト・ホーム・ページが指定されていて、エンド・ユーザーがブラウザに次の URL を入力したとします。</p> <p><code>http://myapp.myserver.com:2000/pls/myapp/</code></p> <p>この場合、URL は自動的に次のように更新されます。</p> <p><code>http://myapp.myserver.com:2000/pls/myapp/myapp.home</code></p> |
| Document Table | <p>それを使用して作成済みの Web サイトにアップロードされたファイルの格納先となるデータベース表の名前を入力します。この入力フィールドのデフォルト値は、サイトを作成したスキーマの名前に基づいて決まります。</p> |
| Document Access Path | <p>URL 内のパスを入力します。これは、ドキュメントが参照されていることを示すために使用されます。たとえば、次のような URL があるとします。</p> <p><code>http://myapp.myserver.com:2000/pls/my_site/docs/folder1/presentation.htm</code></p> <p>この場合、docs がドキュメント・アクセス・パスです。</p> |
| Document Access Procedure | <p>ドキュメントのアップロードおよびダウンロードに使用されるプロシージャを入力します。</p> |
| Extensions to be Uploaded as Long Raw | <p>LONG RAW としてアップロードするファイルの拡張子を指定します。</p> |
| Path Alias | <p>パスの別名用に PL/SQL アプリケーションによって使用されません。</p> <p>WebDB R2.x をご使用の場合の注意: DAD が既存の WebDB 2.x Web サイト用の場合、このフィールドは空白のままにする必要があります。</p> |

Path Alias Procedure

パスの別名用に PL/SQL アプリケーションによって使用されます。

WebDB R2.x をご使用の場合の注意 : DAD が既存の WebDB 2.x Web サイト用の場合、このフィールドは空白のままにする必要があります。

DAD 管理のセキュリティ

Database Access Descriptors (DAD) の管理ページへのアクセス権限付与を行うデータベース管理者は、これらのページを一般のアクセスから保護する必要があります。保護しないと、すべてのユーザーが PL/SQL ゲートウェイの機能を制御する DAD エントリを作成、編集および削除できてしまいます。

PL/SQL ゲートウェイの設定セクションを次のように編集すると、Oracle Portal のデータベース管理者またはデータベース管理者レベルの権限を持つユーザーのみが、これらのページにアクセスできるようになります。

1. `wdbsvr.app` という名前の PL/SQL ゲートウェイ設定ファイルを開きます。この設定ファイルにより、PL/SQL ゲートウェイ・モジュールの設定が定義されます。このファイルは、次の場所に存在します。

```
<ORACLE_HOME>/Apache/modplsql/cfg/wdbsvr.app
```

<ORACLE_HOME> は、Oracle9i Application Server のインストール先です。

2. [WVGATEWAY] セクション（通常はファイルの一番上）で、`admindad` パラメータを探します。
3. シングル・サインオンが使用可能（`enable Basis= yes`）になっているスキーマの有効な DAD 名を入力します。通常、この名前には、Oracle Portal オブジェクトがインストールされている DAD の名前を設定します。デフォルトの名前は `portal30` です。

`wdbsvr.app` ファイル内で、Oracle Portal 3.0 ゲートウェイのセキュリティ・パラメータは次のように設定されています。

```
administrators = all
adminPath = /admin_/
admindad = portal30
debugModules = all
defaultDAD = simplifiedad
```

`portal30` は、ご使用の Oracle Portal 環境が含まれているスキーマ名です。

4. portal30 を、ご使用の Oracle Portal の DAD 名に変更します。

ユーザーが DAD の管理ページにアクセスすると、シングル・サインオンによって認証が実行されます。ログオンしたユーザーがこれらのページへのアクセスを許可されている場合、DAD 設定のメイン・ページが表示されます。ユーザーが認証されない場合、シングル・サインオン・ページで、ユーザー名とパスワードの入力を求められます。入力した情報でユーザーがこれらのページへのアクセスを許可されない場合、エラー・メッセージが表示されます。

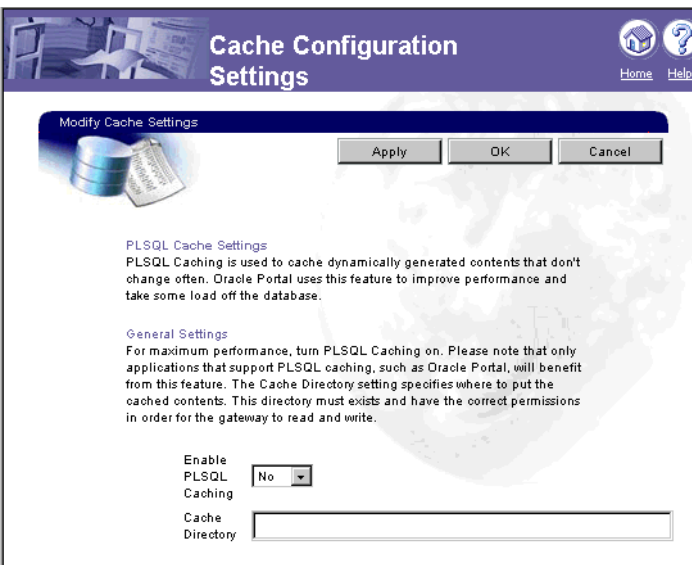
キャッシュの設定

「Cache Configuration Settings」ページにアクセスするには、PL/SQL ゲートウェイの設定ページの「Cache Settings」リンクをクリックするか、または次の URL から直接アクセスします。

`http://<hostname>:<port>/pls/admin_/cache.htm`

このページは、PL/SQL キャッシュとセッション Cookie キャッシュの 2 種類のキャッシュ用の 2 つのサブセクションに分かれています。

次に、「Cache Configuration Settings」ページの一部を示します。



Cache Configuration Settings

Home Help

Modify Cache Settings

Apply OK Cancel

PLSQL Cache Settings

PLSQL Caching is used to cache dynamically generated contents that don't change often. Oracle Portal uses this feature to improve performance and take some load off the database.

General Settings

For maximum performance, turn PLSQL Caching on. Please note that only applications that support PLSQL caching, such as Oracle Portal, will benefit from this feature. The Cache Directory setting specifies where to put the cached contents. This directory must exist and have the correct permissions in order for the gateway to read and write.

Enable PLSQL Caching

Cache Directory

PL/SQL キャッシュ

PL/SQL Cache Settings Enable PL/SQL Caching

パフォーマンスを最大にするために PL/SQL キャッシュを使用可能にするには、「**Yes**」を選択します。この機能に関して問題がある場合は「**No**」を選択します。

Cache Directory

PL/SQL キャッシュによってキャッシュされたコンテンツ・ファイルの格納に使用するディレクトリを入力します。

注意: このディレクトリが存在し、PL/SQL ゲートウェイがこのディレクトリのファイルの読み込み / 書き込み権限を持っていることを確認してください。

Total Cache Size (in bytes)

PL/SQL キャッシュに使用可能なディスク領域の合計を指定します。

注意: この設定は厳密な制限ではありません。キャッシュが一時的にこの制限を超える可能性があります。

Maximum Cacheable File Size (in bytes)

キャッシュされるすべてのファイルに対する最大サイズを指定します。動的に生成されたコンテンツでこの制限を超えたものは、キャッシュされません。

Total Cleanup Limit Size (in bytes)

クリーンアップの発生後に維持するキャッシュの合計サイズを指定します。これにより、頻繁にアクセスされるコンテンツはクリーンアップ後もキャッシュに確実に残されます。

Cleanup Interval (in seconds)

クリーンアップの実行周期の秒数を指定します。数値が大きいが方がパフォーマンスが改善されますが、キャッシュの合計サイズを超える可能性があります。小さい数値を指定した場合、パフォーマンスは低下しますが、キャッシュの合計サイズを超える可能性は低くなります。

セッション Cookie キャッシュ

Session Cookie Cache Settings Enable Session Caching

パフォーマンスを最大にするためにセッション Cookie キャッシュを使用可能にするには、「**Yes**」を選択します。この機能に関して問題がある場合は「**No**」を選択します。

Cache Directory

セッション Cookie キャッシュによってキャッシュされたコンテンツ・ファイルの格納に使用するディレクトリを入力します。

注意: このディレクトリが存在し、PL/SQL ゲートウェイがこのディレクトリのファイルの読み込み / 書き込み権限を持っていることを確認してください。

Total Cache Size (in bytes)

セッション Cookie キャッシュに使用可能なディスク領域の合計を指定します。

注意: この設定は厳密な制限ではありません。キャッシュが一時的にこの制限を超える可能性があります。

Total Cleanup Limit Size (in bytes)

クリーンアップの発生後に維持するキャッシュの合計サイズを指定します。これにより、頻繁にアクセスされるコンテンツはクリーンアップ後もキャッシュに確実に残されます。

Cleanup Interval (in seconds)

クリーンアップの実行周期の秒数を指定します。数値が大きい方がパフォーマンスが改善されますが、キャッシュの合計サイズを超える可能性があります。小さい数値を指定した場合、パフォーマンスは低下しますが、キャッシュの合計サイズを超える可能性は低くなります。

PL/SQL ゲートウェイを WebDB とともに 使用するための設定

この項は、PL/SQL ゲートウェイ経由で WebDB リリース 2.x (2.1、2.2) を実行する WebDB ユーザーを対象としています。

開始前の作業

- OWA_PUBLIC または OAS_PUBLIC 内に存在する旧版の OWA パッケージをすべて削除します。
- PL/SQL ゲートウェイ付属の最新の OWA パッケージをインストールします。これを行うには、SYS ユーザーでデータベースに接続し、コマンド・プロンプトから次のコマンドを実行します。

```
@owaload.sql log.txt
```

これにより、既存の PL/SQL プロシージャの一部が無効になる可能性があります。その場合、再コンパイルする必要があります。詳細は、2-2 ページの「[必須パッケージのインストール](#)」を参照してください。

- wdbsvr.app 設定ファイル内の WebDB 2.x スキーマの DAD 設定に、次のものを設定してください。

Authentication Mode = Basic

Document Table = schema.www_document

Extensions to be Uploaded as Long Raw = *

「Add for WebDB 2.x configuration」ページ (http://<hostname>:<port>/pls/admin_/dadentries.htm にリンク) を使用して DAD を設定すると、これらは自動的に設定されます。

- WebDB 2.x サイトを使用可能にするには、サイトの所有者としてデータベースに接続し、wwwdocs.sql および wwwdocb.plb を実行します。これらのファイルは、owaload.sql ファイルと同じディレクトリに入っています。詳細は、2-2 ページの「[必須パッケージのインストール](#)」を参照してください。

PL/SQL Web Toolkit の使用

PL/SQL ゲートウェイを使用する前に、Oracle データベースの SYS スキーマに、PL/SQL Web Toolkit のパッケージをインストールする必要があります。（**注意**：Oracle 8.1.7 データベースがインストールされている場合、パッケージはデフォルトでこのスキーマにインストールされています。）パブリック・シノニムを使用することにより、ユーザーはこの共通スキーマのオブジェクトを実行できます。ユーザーは、共通スキーマの権限ではなく、自分の権限を使用して共通スキーマのオブジェクトを実行します。

複数の PL/SQL Web Toolkit のインスタンスがデータベースにインストールされている場合、各スキーマから以前のバージョンのパッケージを削除することをお勧めします。

PL/SQL Web Toolkit のインストール

PL/SQL ゲートウェイのインストール時に PL/SQL Web Toolkit をインストールしなかった場合は、owaload.sql インストレーション・スクリプトを使用してインストールできます。詳細は、2-2 ページの「[必須パッケージのインストール](#)」を参照してください。

PL/SQL Web Toolkit のパッケージ

PL/SQL Web Toolkit には、次のパッケージが含まれています。

| パッケージ | 説明 |
|--------------|---|
| htf および http | <p>http（ハイパーテキスト・プロシージャ）パッケージには、HTML タグを生成するプロシージャが含まれています。たとえば、<code>http.anchor</code> プロシージャは HTML のアンカー・タグ <code><A></code> を生成します。</p> <p>htf（ハイパーテキスト・ファンクション）パッケージには、http パッケージのプロシージャのファンクション・バージョンが入っています。ファンクション・バージョンは、Web ページに出力を直接生成しません。かわりに、出力を戻り値として実行元の PL/SQL 文に渡します。コールをネストする必要がある場合、これらのファンクションを使用します。</p> <p>htf ファンクションの結果を出力するには、<code>http.print</code> プロシージャからそれらのファンクションをコールします。これにより、生成された Web ページにパラメータ値が出力されます。</p> |
| owa | PL/SQL ゲートウェイが必要とするサブプログラムが含まれています。 |
| owa_content | コンテンツ・サービス・リポジトリへの問合せを行い、ドキュメント・プロパティを操作するためのファンクションとプロシージャが含まれています。 |
| owa_sec | <p>PL/SQL ゲートウェイがリクエストを認証するために使用するサブプログラムが含まれています。</p> <p>注意：このパッケージは、Oracle Application Server とともに PL/SQL Web Toolkit をインストールした場合に含まれています。PL/SQL ゲートウェイでは使用しません。</p> |
| owa_util | <p>ユーティリティ・サブプログラムが含まれています。このパッケージの内容は、次の領域に分かれています。</p> <ul style="list-style-type: none">動的 SQL ユーティリティ。これにより、動的に生成される SQL コードを含んだ HTML を生成できます。HTML ユーティリティ。これにより、CGI 環境変数の値を取り出して、URL リダイレクトを実行できます。日付ユーティリティ。これにより、日付の処理を修正できます。日付の値は、HTML では単純な文字列として扱われますが、Oracle データベースではデータ型として正しく取り扱う必要があります。 |
| owa_pattern | 正規表現の機能を使用した文字列のマッチングと操作に使用するサブプログラムが含まれています。 |
| owa_text | 文字列を処理するために <code>owa_pattern</code> が使用するサブプログラムが含まれています。これらは外部化されているため、直接使用できます。 |

| パッケージ | 説明 |
|--------------|---|
| owa_image | ユーザーがクリックしたイメージの位置の座標を取得するサブプログラムが含まれています。リンク先により PL/SQL ゲートウェイが実行されるイメージ・マップが存在する場合は、このパッケージを使用します。 |
| owa_cookie | HTTP の Cookie の送信およびクライアントのブラウザからの取得を行うサブプログラムが含まれています。Cookie は、複数の HTTP コール間の状態を管理するためにブラウザに送信される不透過の文字列です。状態は、クライアントのセッションが確立されている間のみ、または有効期限が指定されている場合は有効期限まで保持されます。システム日付は、owa_custom パッケージに指定されている情報を参照して計算されます。 |
| owa_opt_lock | 更新情報の消失を防ぐために、データベースに簡単なロック機構を付加するサブプログラムが含まれています。ユーザーが行を選択して更新しようとしている間に別のユーザーがその値を変更した場合、更新情報が消失する可能性があります。 |
| owa_custom | authorize ファンクションと、Cookie が使用するタイムゾーン定数が含まれています。 注意: このパッケージは、Oracle Application Server とともに PL/SQL Web Toolkit をインストールした場合に含まれています。PL/SQL ゲートウェイでは使用しません。 |
| owa_cache | PL/SQL Web アプリケーションのパフォーマンスを改善するために PL/SQL ゲートウェイのキャッシュ機能を使用可能にするファンクションとプロシージャが含まれています。 |

http および htf パッケージ

http パッケージおよび htf パッケージには、ストアド・プロシージャから HTML タグを生成できるサブプログラムが含まれています。たとえば、次のコマンドを実行すると、簡単な HTML ドキュメントが生成されます。

```

create or replace procedure hello AS
BEGIN
    http.htmlopen;           -- generates <HTML>
    http.headopen;           -- generates <HEAD>
    http.title('Hello');     -- generates <TITLE>Hello</TITLE>
    http.headclose;          -- generates </HEAD>
    http.bodyopen;           -- generates <BODY>
    http.header(1, 'Hello');  -- generates <H1>Hello</H1>
    http.bodyclose;          -- generates </BODY>
    http.htmlclose;          -- generates </HTML>
END;
```

また、これらのパッケージには出力プロシージャ (http.print など) が存在し、引数を現行ドキュメントに書き出すことができます。これらの出力プロシージャを使用すると、標準以外の HTML の生成やファンクションの戻り値を表示したり、あるいは HTML ドキュメントにそのまま表示されるハードコードされたテキストを渡すことができます。生成されたテキストは PL/SQL ゲートウェイに渡され、その後、ユーザーのブラウザに送信されます。

owa_image パッケージ

owa_image パッケージには、ユーザーがクリックしたイメージの位置の座標を取得するサブプログラムが含まれています。このパッケージは、PL/SQL ゲートウェイを実行するイメージ・マップに使用します。プロシージャは次のようになります。

```
create or replace procedure process_image
(my_img in owa_image.point)
as
  x integer := owa_image.get_x(my_img);
  y integer := owa_image.get_y(my_img);
begin
  /* process the coordinate */
end
```

owa_opt_lock パッケージ

owa_opt_lock パッケージには、更新情報の消失を防ぐために、データベースに簡単なロック機構を付加するサブプログラムが含まれています。ユーザーが行を選択して更新しようとしている間に別のユーザーがその値を変更した場合、更新情報が消失する可能性があります。

HTTP は状態を保持しないプロトコルであるため、PL/SQL ゲートウェイは、従来型のデータベースのロック構造を使用できません。owa_opt_lock パッケージでは、次の2つの方法でこの更新情報の消失の問題を回避できます。

- 隠しフィールドを使用する場合、HTML ページの隠しフィールドに以前の値を格納します。ユーザーが更新を要求すると、PL/SQL ゲートウェイは、データベースの現在の状態と格納済みの値が一致するかどうかをチェックします。更新処理は、値が一致した場合にのみ実行されます。この方法を使用するには、owa_opt_lock.store_values プロシージャをコールします。
- チェックサムを使用する場合、値そのものを格納するのではなく、チェックサムを格納します。この方法を使用するには、owa_opt_lock.checksum ファンクションをコールします。

これらの方法は、楽観的なものです。つまり、他のユーザーが更新できないようにするのはなく、更新の割込みが発生した場合に現行の更新を無効にします。

owa_custom パッケージ

owa_custom パッケージには、authorize ファンクションと、Cookie が使用するタイムゾーン定数が含まれています。Cookie はグリニッジ標準時 (GMT) で定義された有効期限を使用します。GMT を使用していない場合は、この2つの定数のうちのいずれかを使用して自分用のタイムゾーンを設定できます。

ご使用のタイムゾーンを Oracle が認識する場合は、`dbms_server_timezone` を使用して直接指定できます。この値はタイムゾーンの省略形です。（認識されるタイムゾーンのリストは、『Oracle8i SQL リファレンス』を参照してください。）たとえば、太平洋標準時の場合は、次のように使用します。

```
dbms_server_timezone constant varchar2(3) := 'PST';
```

Oracle が認識しないタイムゾーンを使用する場合は、`dbms_server_gmtdiff` を使用して GMT との時間差を指定します。使用するタイムゾーンが GMT よりも進んでいる場合は正の数、遅れている場合は負の数を指定します。

```
dbms_server_gmtdiff constant number := NULL;
```

必要な変更を行った後、パッケージをリロードします。

owa_content パッケージ

注意：このパッケージは、Oracle Application Server とともに PL/SQL Web Toolkit をインストールした場合に含まれています。PL/SQL ゲートウェイでは使用しません。

`owa_content` パッケージには、コンテンツ・サービス・リポジトリへの問合せを行い、ドキュメント・プロパティを操作するためのファンクションとプロシージャが含まれています。このパッケージは、次のような作業に使用します。

- ドキュメントの説明の設定
- ドキュメントの削除
- ドキュメントの属性の削除
- 属性情報の取出し
- ドキュメントの属性のリスト
- ドキュメントのコンテンツ・タイプの取出し

`owa_content` パッケージを使用する PL/SQL プロシージャおよびパッケージをコンパイルする場合、次のエラー・メッセージが表示される場合があります。

```
PLS-00201
identifier 'WEBSYS.OWA_CONTENT' must be declared
```

このエラーが発生しないようにするには、非ローカル・データベースを使用する新規 DAD の作成中に DBA ユーザーの入力プロンプトが表示されたときに、SYS ユーザー名と対応するパスワードを入力します。SYSTEM ユーザーを入力すると、データベース・ユーザーに対して正しい権限を付与できなくなります。DBA ユーザーとして SYSTEM を入力した場合は、次のように権限付与オプションを明示的に実行する必要があります。

```
SQL>grant all on WEBSYS.OWA_CONTENT to scott;
```

既存のデータベース・ユーザーを使用して DAD を作成した場合、OWA_CONTENT パッケージの使用前に、前述の権限付与操作を手動で実行する必要があります。

PL/SQL サンプルでは、OWA_CONTENT パッケージを使用するため、これらのステップは、PL/SQL サンプルをインストールする前に実行する必要があります。

owa_cache パッケージ

owa_cache パッケージには、PL/SQL Web アプリケーションのパフォーマンスを改善するために PL/SQL ゲートウェイのキャッシュ機能を使用可能にするファンクションとプロシージャが含まれています。この項では、これらのファンクションおよびプロシージャの仕様について説明します。(PL/SQL ゲートウェイのキャッシュ機能の詳細は、1-20 ページの「[キャッシング](#)」を参照してください。)

- `owa_cache.disable`
この特定のリクエストについてキャッシュを使用禁止にします。
- `owa_cache.set_expires(p_expires IN number, p_level IN varchar2)`
期限モデル・キャッシュ・タイプ用のキャッシュ・ヘッダーを設定します。

パラメータ

`p_expires IN` - このコンテンツの有効期間 (分)。

`p_level IN` - キャッシュ・レベル。

例外

次のいずれかの場合、VALUE_ERROR が発生します。

`p_expires` が負の数値または 0 (ゼロ) である。

`p_level` が 'USER' または 'SYSTEM' ではない。

`p_expires` が > 525600 (1 年) である。

- `owa_cache.set_cache(p_etag IN varchar2, p_level IN varchar2)`
妥当性チェック・モデル・キャッシュ・タイプ用のキャッシュ・ヘッダーを設定します。

パラメータ

`p_etag IN` - このコンテンツに関連付けられたタグ。

`p_level IN` - キャッシュ・レベル。

例外

次のいずれかの場合、VALUE_ERROR が発生します。

`p_etag` が 55 より大きい。

p_level が 'USER' または 'SYSTEM' ではない。

- owa_cache.set_not_modified
変更されていないキャッシュ・ヒット用のヘッダーを設定します。妥当性チェック方式でのみ使用されます（詳細は 1-21 ページの「[妥当性チェック方式](#)」を参照してください）。

例外

ETag が渡されない場合、VALUE_ERROR が発生します。

- owa_cache.get_level
キャッシュ・レベルを返します。妥当性チェック・モデルでのみ使用されます（詳細は 1-21 ページの「[妥当性チェック方式](#)」を参照してください）。

戻り値

キャッシュ・ヒットの場合はキャッシュ・レベルの文字列 ('USER' または 'SYSTEM')。それ以外の場合は NULL。

- owa_cache.get_etag
キャッシュされたコンテンツに関連付けられたタグを返します。妥当性チェック方式でのみ使用されます（詳細は 1-21 ページの「[妥当性チェック方式](#)」を参照してください）。

戻り値

キャッシュ・ヒットの場合はタグ。それ以外の場合は NULL。

PL/SQL Web Toolkit のパラメータ名の表記規則

PL/SQL Web Toolkit では、パラメータの最初の文字がパラメータのデータ型を示します。

表 5-1

| 最初の文字 | データ型 | 例 |
|-------|----------|-------------------|
| c | VARCHAR2 | cname IN VARCHAR2 |
| n | INTEGER | nsize IN INTEGER |
| d | DATE | dbuf IN DATE |

HTML タグの属性

多くの HTML タグに多くのオプションの属性が存在するため、これらが個々のパラメータとしてハイパーテキスト・プロシージャまたはファンクションに渡されると、コールの扱いが難しくなります。また、標準以外の属性をサポートしているブラウザもあります。そのため、HTML タグを生成する各ハイパーテキスト・プロシージャまたはファンクションには、一番最後のパラメータとしてオプションの `cattributes` が存在します。このパラメータを使用すると、必要な HTML 属性のテキストをそのまま PL/SQL プロシージャに渡すことが可能です。

たとえば、`http.em` の構文は次のとおりです。

```
http.em(ctext, cattributes);
```

次に、HTML3.0 の属性を使用したコールの例を示します。

```
http.em('This is an example','ID="SGML_ID" LANG="en"');
```

これにより、次の結果が生成されます。

```
<EM ID="SGML_ID" LANG="en">This is an example</EM>
```

PL/SQL ゲートウェイおよびアプレット

HTML ファイルで `APPLET` タグを使用してアプレットを参照する場合、サーバーはその HTML ファイルの入っているディレクトリで、そのアプレットのクラス・ファイルを探します。そのアプレットのクラス・ファイルが別のディレクトリに存在する場合は、`APPLET` タグの `CODEBASE` 属性を使用してそのディレクトリを指定します。

PL/SQL ゲートウェイから HTML ページを生成し、その HTML ページがアプレットを参照する場合は、`CODEBASE` 属性を指定します。これは、PL/SQL ゲートウェイには現行ディレクトリという概念がなく、アプレットのクラス・ファイルの検索場所がわからないためです。

次の例では、`http.appletopen` を使用して `APPLET` タグを生成しています。この例では、`CODEBASE` の値を指定するために `cattributes` パラメータを使用します。

```
http.appletopen('myapplet.class', 100, 200, 'CODEBASE="/applets"')
```

これにより、次の結果が生成されます。

```
<APPLET CODE="myapplet.class" height=100 width=200 CODEBASE="/applets">
```

`/applets` は、`myapplet.class` ファイルが存在する仮想パスを示します。

Cookie

Cookie は、クライアントのブラウザからの持続的な状態変数を保持するために使用できます。

http://home.netscape.com/newsref/std/cookie_spec.html
<http://www.virtual.net/Projects/Cookies/>

owa_cookie パッケージを使用して、HTTP ヘッダー内で Cookie の送信と取出しができます。これには、Cookie の値の設定と取得に使用するサブプログラムが含まれています。

- owa_cookie.cookie データ型には、Cookie の名前と値のペアが含まれます。
- owa_cookie.get ファンクションは、指定された Cookie の値を取得します。
- owa_cookie.get_all プロシージャは、Cookie の名前と値のペアをすべて取得します。
- owa_cookie.remove プロシージャは、指定された Cookie を削除します。

LONG データ型

http.print、http.prn、http.prints、http.ps または owa_util.cellsprint などのプロシージャまたはファンクションで LONG データ型の値を使用する場合、LONG データの最初の 32KB のみ使用される点に注意してください。このような制約は、プロシージャまたはファンクションで LONG データが varchar2 データ型にバインドされているために発生しています。

http パッケージおよび htf パッケージの拡張

http パッケージおよび htf パッケージでは、カスタムの拡張機能を使用できます。このため、HTML の標準の変更に合せて、ハイパー・テキスト・プロシージャおよびファンクション・パッケージに似た新機能を追加して、変更を反映することが可能です。

次に、標準以外の <BLINK> タグと架空の <SHOUT> タグを使用してカスタマイズされたパッケージの例を示します。

```
create package nsf as
    function blink(cbuf in varchar2) return varchar2;
    function shout(cbuf in varchar2) return varchar2;
end;

create package body nsf as
    function blink(cbuf in varchar2) return varchar2 is
        begin return ('<BLINK>' || cbuf || '</BLINK>');
    end;
```

```
function shout(cbuf in varchar2) return varchar2 is
    begin return ('<SHOUT>' || cbuf || '</SHOUT>');
end;

create package nsp as
    procedure blink(cbufin varchar2);
    procedure shout(cbufin varchar2);
end;

create package body nsp as
    procedure blink(cbufin varchar2) is
        begin http.print(nsf.blink(cbuf));
    end;
    procedure shout(cbufin varchar2) is
        begin http.print(nsf.shout(cbuf));
    end;
end;
```

これにより、前述のプロシージャとファンクションを自分のプロシージャ内で使用できるようになりました。

```
create procedure nonstandard as
begin
    nsp.blink('Gee this hurts my eyes!');
    http.print('And I might ' || nsf.shout('get mad!'));
end;
```

文字列のマッチングおよび操作

owa_pattern パッケージには、正規表現の機能を使用した文字列のマッチングと操作に使用するプロシージャおよびファンクションが含まれています。このパッケージには、次のサブプログラムがあります。

- owa_pattern.match ファンクションは、文字列中に正規表現が存在するかどうかを判断します。TRUE または FALSE が返されます。
- owa_pattern.amatch ファンクションは、owa_pattern.match ファンクションの、さらに高度なバリエーションです。これにより文字列内の一致の位置を指定できます。このファンクションは正規表現が見つかった場合にその文字列の終了位置を返します。正規表現が見つからなかった場合は、0 を返します。
- owa_pattern.change ファンクションおよびプロシージャは、一致した正規表現の文字列の部分を新しい文字列に置き換えます。ファンクションとしてコールした場合は、正規表現が見つかって置換された回数を返します。

これらのサブプログラムはオーバーロードされます。つまり、各サブプログラムには複数のバージョンがあり、使用するパラメータにより区別されています。具体的には、MATCH には 6 つ、AMATCH と CHANGE にはそれぞれ 4 つのバージョンがあります。サブプログラムでは、次のパラメータを使用します。

- line: 一致の調査対象となるターゲット。名前に反して、2 行以上のテキストまたは owa_text.multi_line データ型の場合があります。
- pat: サブプログラムが line 中に設定するパターン。このパターンには正規表現を使用できます。
- owa_pattern.change ファンクションおよびプロシージャでは、このパラメータは from_str という名前になっているため注意してください。
- flags: 検索で大文字・小文字を区別するか、またはグローバル置換を行うかどうかを指定。

owa_pattern.match

このファンクションの正規表現は、VARCHAR2 または owa_pattern.pattern データ型のいずれかになります。owa_pattern.getpat プロシージャを使用して、文字列から owa_pattern.pattern データ型を作成できます。

owa_text.stream2multi プロシージャを使用して、長い文字列から multi_line データ型を作成できます。multi_line を使用する場合、rlist パラメータによって、見つかった一致部分のリストを指定します。

line が multi_line でなく文字列の場合、backrefs という名前のオプションの出力パラメータを追加できます。このパラメータは、正規表現中の一連のトークンに一致したターゲットの各文字列を格納する row_list を指定します。次に、owa_pattern.match ファンクションの例を示します。

```
boolean foundMatch;
foundMatch := owa_pattern.match('KAZOO', 'zoo.*', 'i');
```

このファンクションの意味: KAZOO は正規表現 zoo.* の検索ターゲットです。ピリオドは改行以外のすべての文字を表し、アスタリスクはその前にある文字の 0 回以上の繰返しと一致します。この場合、このアスタリスクは改行以外のすべての文字と一致します。

つまり、この正規表現は一致するターゲットに zoo が含まれ、zoo は改行で終了しない、または改行を含んでいない（改行はピリオドとは一致しない）任意の文字セットで構成されることを示します。i は、この検索で大文字・小文字の区別をしないことを示すフラグです。この場合、このファンクションは TRUE を返し、一致が見つかったことを示します。

owa_pattern.change

owa_pattern.change は、実行方法によってプロシージャにもファンクションにもなります。ファンクションとして使用すると、変更回数が返されます。フラグ g を使用しない場合、この数値は 0 または 1 のみになります。フラグ g は、一致したものがすべて正規表現で置き換えられることを示します。これ以外の場合は、最初に一致したもののみ置き換えられます。

置換文字列には、正規表現に一致したターゲットの一部が置換式に入ることを示すトークンのアンパサンド (&) を使用できます。たとえば、次のようになります。

```
owa_pattern.change('Cats in pajamas', 'C.+in', '& red ');
```

この正規表現は、文字列 Cats in にマッチします。次に、この文字列は & red に置換されます。アンパサンド文字 & は、正規表現によって一致した Cats in を示します。このように、このプロシージャは、文字列 Cats in pajamas を Cats in red に置き換えます。これをプロシージャではなくファンクションとしてコールすると、返される値は Cats in red ではなく 1 になり、置換が 1 回行われたことを示します。

PL/SQL ゲートウェイ・チュートリアル

この項では、データベースの表の内容を HTML テーブルとして表示する簡単なアプリケーションの作成方法について、ステップごとに説明します。このアプリケーションは、PL/SQL Web Toolkit で定義済みのファンクションおよびプロシージャをコールするストアド・プロシージャを実行します。

このチュートリアルの前提は、次のとおりです。

- 2-2 ページの「[必須パッケージのインストール](#)」の項を完了していること。
- サーバーに admin ユーザーとしてログインできること。これは、サーバーの設定に新しい設定を追加するために必要です。接続するデータベースに、すでに PL/SQL Web Toolkit がインストールされていることが必要です。詳細は、5-1 ページの「[PL/SQL Web Toolkit のインストール](#)」を参照してください。
- Oracle データベースに SCOTT スキーマが存在すること。PL/SQL カートリッジは、ユーザー名に scott、パスワードに tiger を使用してデータベースにログインします。SCOTT スキーマが存在しない場合は、データベースの既存のスキーマを使用するか、CREATE SCHEMA コマンドを使用して SCOTT を作成します。

スキーマは、表、ビュー、プロシージャ、ファンクションなどのデータベース・オブジェクトの集まりを含んだユーザー・アカウントです。スキーマ内の各オブジェクトは、同じスキーマ内の他のオブジェクトにアクセスできます。

ストアド・プロシージャの作成およびデータベースへのロード

アプリケーションが実行するストアド・プロシージャは、current_users（次の例で定義されている）です。このプロシージャは、all_users 表の内容を取り出し、HTML テーブルとして書式化します。

ストアド・プロシージャを作成するには、current_users.sql というファイルにプロシージャのテキストを保存し、次に Oracle Server Manager を実行してファイル内の文を読み込み、実行します。

1. 次の行を入力して、このプロシージャを `current_users.sql` というファイルに保存します。`current_users` プロシージャは、`all_users` 表の内容を取り出し、HTML テーブルとして書式化します。

```
create or replace procedure current_users
AS
    ignore boolean;
BEGIN
    http.htmlopen;
    http.headopen;
    http.title('Current Users');
    http.headclose;
    http.bodyopen;
    http.header(1, 'Current Users');
    ignore := owa_util.tablePrint('all_users');
    http.bodyclose;
    http.htmlclose;
END;
/
show errors
```

このプロシージャは、`http` および `owa_util` パッケージのファンクションおよびプロシージャを使用して HTML ページを生成します。たとえば、`http.htmlopen` プロシージャは次の文字列を生成します。

```
<html>
```

また、`http.title('Current Users')` は、次の文字列を生成します。

```
<title>Current Users</title>
```

`owa_util.tablePrint` ファンクションは、指定されたデータベースの表を検索して、その内容を HTML テーブルとして書式化します。

2. `Server Manager` をコマンド・ライン・モードで実行します。`ORACLE_HOME` は、Oracle データベースのファイルが入っているディレクトリです。

```
prompt> $ORACLE_HOME/bin/svrmgrl
```

3. "scott" でデータベースに接続します。パスワードは "tiger" です。

```
SVRMGR> connect scott/tiger
```

4. `current_users.sql` ファイルから `current_users` ストアド・プロシージャをロードします。`current_users.sql` ファイルが入っているディレクトリ以外のディレクトリから `Server Manager` を起動した場合は、そのファイルの絶対パスを指定する必要があります。

```
SVRMGR> @< 絶対パス >current_users.sql
```

5. Server Manager を終了します。

```
SVRMGR> exit
```

6. PL/SQL ゲートウェイを使用して実行する PL/SQL アプリケーションが格納されているスキーマを指すように、次の表に示すパラメータを使用して DAD を設定します。

表 6-1

| パラメータ | 値 |
|---------------------------------------|------------------------------------|
| Database Access Descriptor Name | Scott |
| Schema Name | Scott |
| Oracle User Name | Scott |
| Oracle Password | Tiger |
| Oracle Connect String | htmlperf-tcp |
| Authentication Mode | Basic |
| Session Cookie Name | |
| Create a Stateful Session? | No |
| Enable Connection Pooling? | Yes |
| Maximum Number of Worker Threads | 10 |
| Default (Home) Page | Scott.home |
| Document Table | Scott.wwdoc_document |
| Document Access Path | docs |
| Document Access Procedure | Scott.wpg_testdoc.process_download |
| Extensions to be Uploaded as Long Raw | * |
| Path Alias | |
| Path Alias Procedure | |

注意: 実行ユーザーに対してアプリケーションが含まれているデータベースへのログインを必須とする場合は、「Oracle User Name」フィールドと「Oracle Password」フィールドを空白にします。

アプリケーションを実行する HTML ページの作成

current_users プロシージャを実行するには、ブラウザで次の URL を入力します。

```
http://<host>:<port>/pls/mydad/scott.current_users
```

ただし、HTML ページからプロシージャを実行する方が一般的です。たとえば、次の HTML ページには前述の URL をコールするリンクが含まれています。

```
<HTML>
<HEAD>
<title>Current Users</title>
</HEAD>

<BODY>
<H1>Current Users</H1>
<p><a href="http://hal.us.oracle.com:9999/simpleApp1/cart1/current_users">Run
current_users</a>
</BODY>
</HTML>
```

次の図に、ソース・ページ（ストアド・プロシージャを実行するリンクが含まれるページ）と、current_users ストアド・プロシージャによって生成されるページを示します。

Current Users

[Run current_users](#)

索引

A

Apache
停止, 2-8

C

CGI
owa_util PL/SQL Web Toolkit パッケージ, 5-2
CONTENT_TYPE 列, 1-13
CONTENT 列, 1-13
Cookie, 5-9
owa_cookie PL/SQL Web Toolkit パッケージ, 5-3

D

DAD_CHARSET 列, 1-14
Database Access Descriptor (DAD)
設定, 3-1
保護, 3-6
定義, 1-2
Database Access Descriptor, 保護, 3-6
Document Table Definition
以前のスタイル, 1-14
document_path, 1-15
document_proc, 1-15
document_table, 1-14
DTD, 1-12
以前のスタイル, 1-14

G

GET 方式, 1-4

H

HTML タグ
属性, 5-8
HTML ページ
アプリケーションを実行, 6-4
http および htf PL/SQL Web Toolkit パッケージ, 5-2,
5-3
拡張, 5-9

L

LONG データ型, 5-9

M

MIME タイプ, 1-18

O

owa PL/SQL Web Toolkit パッケージ, 5-2
owa_content PL/SQL Web Toolkit パッケージ, 5-2,
5-5
owa_cookie PL/SQL Web Toolkit パッケージ, 5-3
owa_custom PL/SQL Web Toolkit パッケージ, 5-3,
5-4
owa_image PL/SQL Web Toolkit パッケージ, 5-3, 5-4
owa_opt_lock PL/SQL Web Toolkit パッケージ, 5-3,
5-4
owa_pattern PL/SQL Web Toolkit パッケージ, 5-2
owa_pattern.change ファンクションおよびプロシ
ージャ, 5-12
owa_pattern.match ファンクション, 5-11
owa_sec PL/SQL Web Toolkit パッケージ, 5-2
owa_text PL/SQL Web Toolkit パッケージ, 5-2

owa_util PL/SQL Web Toolkit パッケージ, 1-26, 5-2
owaload.sql, 2-2

P

pls.conf 設定ファイル, 2-7
PL/SQL ゲートウェイ
 WebDB とともに使用する, 4-1
 アプレット, 5-8
 実行, 1-3
 設定, 2-7, 3-1
 チュートリアル, 6-1
PL/SQL Web Toolkit, 5-1
PL/SQL ゲートウェイ
 機能, 1-5
PL/SQL プロシージャ
 データベースへのロード, 6-1
POST 方式, 1-4

R

REQUEST_CHARSET, 1-27

U

upload_as_content_type, 1-15

W

WebDB, 4-1

あ

アップロード, 1-12
アプレット, 5-8

い

イメージ
 owa_image PL/SQL Web Toolkit パッケージ, 5-3
インストール, 2-1

お

オーバーロード, 1-8, 1-9

か

環境変数
 CGI, 1-26
管理ページ
 Database Access Descriptor (DAD), 3-6
 アクセス設定, 3-2

く

クライアント・リクエスト, 1-2
グローバル設定, 3-2

し

システム要件, 2-1

せ

セキュリティ
 Database Access Descriptor (DAD) 管理, 3-6
設定
 Database Access Descriptor (DAD), 3-1
 PL/SQL ゲートウェイ, 2-7, 3-1
 WebDB, 4-1

た

ダウンロード, 1-12

ち

チュートリアル, 6-1

て

データベース
 パスワードの設定, 3-4
 ロック, 5-4

と

動的 SQL ユーティリティ
 owa_util パッケージ内, 5-2
ドキュメント・アクセス・パス, 1-15
 設定, 3-5
ドキュメント表

- 設定, 3-5
- ドキュメント表の定義, 1-12
- トランザクション・モデル, 1-8

に

- 認証, 1-5
- 認証解除, 1-6

は

- 配列, 1-9
- バスの別名
 - 設定, 3-5
- パラメータ
 - PL/SQL Web Toolkit, 5-7
 - 受渡し, 1-8, 1-10
 - 大きな, 1-11
 - オーバーロード, 1-8
 - 柔軟, 1-10

ひ

- 日付ユーティリティ
 - owa_util パッケージ内, 5-2

ふ

- ファイルのアップロード, 1-12, 1-16
 - 属性, 1-18
 - ドキュメント表, 3-5
 - 複数のファイル, 1-18
- ファイルのダウンロード, 1-18

ほ

- ホーム・ページ
 - 設定, 3-5

も

- 文字列のマッチング, 5-10

