# Oracle® HTTP Server

Administrator's Guide

10*g* Release 1 (10.1)

**Part No. B12255-01**

December 2003

ORACLE®

Oracle HTTP Server Administrator's Guide, 10*g* Release 1 (10.1)

Part No. B12255-01



Primary Author: Priya Darshane

Contributor: Julia Pond, Warren Briese, Kevin Clark, Priscila Darakjian, Sander Goudswaard, Pushkar Kapasi, Chuck Murray, Mark Nelson, Bert Rich, Shankar Raman, Baogang Song, Kevin Wang

# Contents

## 7 Oracle HTTP Server Modules

## 8 Managing Security



## 9 Frequently Asked Questions

# Send Us Your Comments

**Oracle HTTP Server Administrator's Guide, 10*g* Release 1 (10.1)**

**Part No. B12255-01**

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: infodev_us@oracle.com
- FAX: 650-506-7227 Attn: Server Technologies Documentation Manager
- Postal service:
  Oracle Corporation
  Server Technologies Documentation
  500 Oracle Parkway, Mailstop 4op11
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

# Preface

This guide describes how to administer the Oracle HTTP Server.

This preface contains these topics:

## Intended Audience

The *Oracle HTTP Server Administrator's Guide* is intended for database administrators and security managers.

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

**Accessibility of Links to External Web Sites in Documentation** This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

## Organization

This document contains:

### Chapter 1, "Oracle HTTP Server Overview"

This chapter describes the Oracle HTTP Server, highlighting the differences between the Oracle distribution and the open source Apache product on which it is based. It also explains how to start, stop and restart the server.

### Chapter 2, "Oracle HTTP Server Concepts"

This chapter introduces you to the Oracle HTTP Server directory structure, and configuration files, configuration file syntax, modules, and directives.

### Chapter 3, "Specifying Server and File Locations"

This chapter explains how to set Oracle HTTP Server and server administrator options, and specifies file locations.

### Chapter 4, "Managing Server Processes"

This chapter provides an overview of the Oracle HTTP Server processes, and provides information on how to regulate, and monitor these processes.

### Chapter 5, "Managing the Network Connection"

This chapter provides information about specifying IP addresses and ports, and managing server interaction, and network connection persistence.

### Chapter 6, "Configuring and Using Server Logs"

This chapter discusses Oracle Diagnostic Logging, log formats, and describes various log files and their locations.

### Chapter 7, "Oracle HTTP Server Modules"

This chapter describes the modules (mods) included in the Oracle HTTP Server. The modules extend the basic functionality of the Web server, and support integration between Oracle HTTP Server and other Oracle Database components.

### Chapter 8, "Managing Security"

This chapter provides an overview of Oracle HTTP Server security features and configuration information for setting up a secure Web site using them.

### Chapter 9, "Frequently Asked Questions"

This chapter provides answers to frequently asked questions about Oracle HTTP Server.

### Chapter A, "Oracle HTTP Server Configuration Files"

This appendix lists commonly used Oracle HTTP Server configuration files.

### Chapter B, "Third Party Licenses"

This appendix includes the Third Party License for all the third party products included with Oracle Database.

### Glossary

The glossary defines terminology used throughout this guide and the Oracle Database documentation set.

## Related Documentation

For more information, see these Oracle resources:

- Oracle Database Documentation Library
- Oracle Database Platform-Specific Documentation on Oracle Database Disk 1

Printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com/`

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

`http://otn.oracle.com/membership/`

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

`http://otn.oracle.com/documentation/`

# Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- Conventions in Text
- Conventions in Code Examples
- Conventions for Windows Operating Systems

## Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

| Convention | Meaning | Example |
|---|---|---|
| **Bold** | Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both. | When you specify this clause, you create an **index-organized table**. |
| *Italics* | Italic typeface indicates book titles or emphasis. | *Oracle9i Database Concepts*<br>Ensure that the recovery catalog and target database do *not* reside on the same disk. |
| `UPPERCASE monospace (fixed-width) font` | Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles. | You can specify this clause only for a `NUMBER` column.<br>You can back up the database by using the `BACKUP` command.<br>Query the `TABLE_NAME` column in the `USER_TABLES` data dictionary view.<br>Use the `DBMS_STATS.GENERATE_STATS` procedure. |

| Convention | Meaning | Example |
|---|---|---|
| lowercase monospace (fixed-width) font | Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values.<br>**Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown. | Enter `sqlplus` to open SQL*Plus.<br>The password is specified in the `orapwd` file.<br>Back up the datafiles and control files in the `/disk1/oracle/dbs` directory.<br>The `department_id`, `department_name`, and `location_id` columns are in the `hr.departments` table.<br>Set the `QUERY_REWRITE_ENABLED` initialization parameter to `true`.<br>Connect as `oe` user.<br>The `JRepUtil` class implements these methods. |
| *lowercase italic monospace (fixed-width) font* | Lowercase italic monospace font represents placeholders or variables. | You can specify the *parallel_clause*.<br>Run `U`*old_release*`.SQL` where *old_release* refers to the release you installed prior to upgrading. |

## Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

| Convention | Meaning | Example |
|---|---|---|
| [ ] | Brackets enclose one or more optional items. Do not enter the brackets. | `DECIMAL (`*digits* `[ ,` *precision* `])` |
| { } | Braces enclose two or more items, one of which is required. Do not enter the braces. | `{ENABLE \| DISABLE}` |
| \| | A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar. | `{ENABLE \| DISABLE}`<br>`[COMPRESS \| NOCOMPRESS]` |

| Convention | Meaning | Example |
|---|---|---|
| . . . | Horizontal ellipsis points indicate either:<br>■ That we have omitted parts of the code that are not directly related to the example<br>■ That you can repeat a portion of the code | `CREATE TABLE ... AS subquery;`<br><br>`SELECT col1, col2, ... , coln FROM employees;` |
| .<br>.<br>. | Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example. | `SQL> SELECT NAME FROM V$DATAFILE;`<br>`NAME`<br>`----------`<br>`/fsl/dbs/tbs_01.dbf`<br>`/fs1/dbs/tbs_02.dbf`<br>`.`<br>`.`<br>`.`<br>`/fsl/dbs/tbs_09.dbf`<br>`9 rows selected.` |
| Other notation | You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown. | `acctbal NUMBER(11,2);`<br>`acct CONSTANT NUMBER(4) := 3;` |
| *Italics* | Italicized text indicates placeholders or variables for which you must supply particular values. | `CONNECT SYSTEM/system_password`<br>`DB_NAME = database_name` |
| UPPERCASE | Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase. | `SELECT last_name, employee_id FROM employees;`<br>`SELECT * FROM USER_TABLES;`<br>`DROP TABLE hr.employees;` |
| lowercase | Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.<br><br>**Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown. | `SELECT last_name, employee_id FROM employees;`<br>`sqlplus hr/hr`<br>`CREATE USER mjones IDENTIFIED BY ty3MU9;` |

## Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

| Convention | Meaning | Example |
|---|---|---|
| Choose Start > | How to start a program. | To start the Database Configuration Assistant, choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant. |
| File and directory names | File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (\|), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention. | `c:\winnt"\"system32 is the same as C:\WINNT\SYSTEM32` |
| `C:\>` | Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the *command prompt* in this manual. | `C:\oracle\oradata>` |
| Special characters | The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters. | `C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\"`<br>`C:\>imp SYSTEM/`*`password`*` FROMUSER=scott TABLES=(emp, dept)` |
| *HOME_NAME* | Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore. | `C:\> net start Oracle`*`HOME_NAME`*`TNSListener` |

| Convention | Meaning | Example |
|---|---|---|
| *ORACLE_HOME* and *ORACLE_BASE* | In releases prior to Oracle8*i* release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory. For Windows NT, the default location was `C:\orant`.<br><br>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is `C:\oracle`. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is `C:\oracle\ora`*nn*, where *nn* is the latest release number. The Oracle home directory is located directly under *ORACLE_BASE*.<br><br>All directory path examples in this guide follow OFA conventions.<br><br>Refer to *Oracle9i Database Getting Starting for Windows* for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories. | Go to the *ORACLE_BASE*`\`*ORACLE_HOME*`\rdbms\admin` directory. |

# 1

# Oracle HTTP Server Overview

This chapter describes the Oracle HTTP Server, highlighting the differences between the Oracle distribution and the open source Apache product on which it is based. It also explains how to start, stop and restart the server.

Topics discussed are:

- Oracle HTTP Server Features
- Oracle HTTP Server Components
- Oracle HTTP Server Support
- Oracle HTTP Server Management
- Starting, Stopping, and Restarting Oracle HTTP Server

Documentation from the Apache Software Foundation is referenced when applicable.

> **Note:** Readers using this guide in PDF or hard copy formats will be unable to access third-party documentation, which Oracle provides in HTML format only. To access the third-party documentation referenced in this guide, use the HTML version of this guide and click the hyperlinks.

# Oracle HTTP Server Features

Oracle HTTP Server is the Web server component of Oracle Database. It is based on the **Apache HTTP** Server, version 1.3.28. It is a robust, reliable Web server, preconfigured to do the following:

- provide a high availability infrastructure integration with **Oracle Process Manager and Notification Server** (OPMN), for process management, death detection and failover for Oracle HTTP Server processes.

  **See Also:** *Oracle Application Server 10g High Availability Guide*

- provide Dynamic Monitoring Services (DMS) metrics that give runtime performance statistics for Oracle HTTP Server processes.

  **See Also:** *Oracle Application Server 10g Performance Guide*

- provide a request ID, which enhances request tracking through various components by attaching a request ID to each request. This provides more detailed information, allowing you to see how much time a particular request spends in any component or layer.
- enable securing of transactions with Secure Sockets Layer (SSL) technology.

  **See Also:**

  - *Oracle Application Server 10g Security Guide*
  - Chapter 8, "Managing Security" on page 8-1

- execute Perl scripts in the same process as the Oracle HTTP Server, or as **CGI** script.
- access database stored procedures with a PL/SQL engine.

  **See Also:** *Oracle Application Server 10g mod_plsql User's Guide*

- enable scripting of HTML pages with PL/SQL code.

# Oracle HTTP Server Components

Oracle HTTP Server consists of several components that run within the same process. These components provide the extensive list of features that Oracle HTTP Server offers when handling client requests. Following are the major components:

- **HTTP Listener**: Oracle HTTP Server is based on an Apache HTTP listener to serve client requests. An HTTP server listener handles incoming requests and routes them to the appropriate processing utility.
- **Modules (mods)**: Many of the standard Apache modules are included with Oracle HTTP Server. Oracle also includes several internal modules that are specific to Oracle Database components.

  **See Also:** "Oracle HTTP Server Modules" on page 1-3 for a complete list of modules shipped with Oracle HTTP Server.

- **Perl Interpreter**: A persistent Perl runtime environment embedded in Oracle HTTP Server through `mod_perl`.

  **See Also:** *Oracle Application Server 10g Concepts* for more information regarding Oracle Database components, and how they relate to each other.

## Oracle HTTP Server Modules

Table 1–1 identifies the modules shipped with Oracle HTTP Server. Modules extend the basic functionality of the Web server, and support integration between Oracle HTTP Server and other Oracle Database components. Note that the list differs from the Apache open source distribution (given the inclusion of Oracle modules), and that not all modules are supported by Oracle.

***Table 1–1 Oracle HTTP Server Modules***

| Module | Oracle Support | Notes |
|---|---|---|
| mod_access | **Yes** | |
| mod_actions | **Yes** | |
| mod_alias | **Yes** | |
| mod_asis | **No** | |
| mod_auth | **Yes** | |
| mod_auth_anon | **Yes** | |

*Table 1–1 Oracle HTTP Server Modules (Cont.)*

| Module | Oracle Support | Notes |
|---|---|---|
| mod_auth_db | **No** | Disabled. Not shipped by Oracle. |
| mod_auth_dbm | **No** | |
| mod_auth_digest | **No** | Disabled. Experimental MD5 authentication; not shipped by Oracle. |
| mod_autoindex | **Yes** | |
| mod_cern_meta | **No** | |
| mod_certheaders | **Yes** | |
| mod_cgi | **Yes** | |
| mod_define | **Yes** | UNIX systems only. |
| mod_digest | **Yes** | |
| mod_dir | **Yes** | |
| mod_dms | **Yes** | Oracle module. |
| mod_env | **Yes** | |
| mod_example | **No** | |
| mod_expires | **Yes** | |
| mod_fastcgi | **Yes** | |
| mod_headers | **Yes** | |
| mod_imap | **No** | |
| mod_include | **Yes** | |
| mod_info | **Yes** | |
| mod_isapi | **No** | Windows systems only. Not shipped by Oracle |
| mod_log_agent | **No** | Deprecated. |
| mod_log_config | **Yes** | |
| mod_log_referer | **Yes** | Deprecated. |
| mod_mime | **Yes** | |
| mod_mime_magic | **Yes** | |
| mod_mmap_static | **No** | |
| mod_negotiation | **Yes** | |

*Table 1–1 Oracle HTTP Server Modules (Cont.)*

| Module | Oracle Support | Notes |
|---|---|---|
| mod_onsint | **Yes** | Oracle module. |
| mod_ossl | **Yes** | Oracle module. |
| mod_perl | **Yes** | |
| mod_plsql | **Yes** | Oracle module. |
| mod_proxy | **Yes** | |
| mod_rewrite | **Yes** | |
| mod_setenvif | **Yes** | |
| mod_so | **Yes** | |
| mod_speling | **Yes** | |
| mod_status | **Yes** | |
| mod_unique_id | **Yes** | |
| mod_userdir | **Yes** | |
| mod_usertrack | **Yes** | |
| mod_vhost_alias | **Yes** | |

# Oracle HTTP Server Support

Oracle provides technical support for the following Oracle HTTP Server features and conditions:

- Modules included in the Oracle distribution, except as noted in the table in Table 1–1, "Oracle HTTP Server Modules". Modules from any other source, including the Apache Software Foundation, are not supported by Oracle.
- Problems that can be reproduced within an Apache configuration consisting only of supported Oracle Apache modules.
- Use of the included Perl interpreter within the supported Apache configuration.

# Oracle HTTP Server Management

You can manage Oracle HTTP Server using `opmnctl`. It is the command-line utility for Oracle Process Manager and Notification Server (OPMN) for process management. It is located in

- UNIX: *ORACLE_HOME*`/opmn/bin`
- Windows: *ORACLE_HOME*`\opmn\bin`

> **See Also:** *Oracle Process Manager and Notification Server Administrator's Guide* for more information on `opmnctl`.

# Starting, Stopping, and Restarting Oracle HTTP Server

Oracle HTTP Server is managed by Oracle Process Manager and Notification Server (OPMN). You must always use the `opmnctl` utility to start, stop and restart Oracle HTTP Server. Otherwise, the configuration management infrastructure cannot detect or communicate with the Oracle HTTP Server processes, and problems may occur.

> **Note:** Do not use the `apachectl` utility to manage the Oracle HTTP Server.

To determine the state of Oracle HTTP Server, use the following command:

```
opmnctl status
```

The processes are listed with their current state such as "Up" or "Down".

## Starting Oracle HTTP Server

To start Oracle HTTP Server, use the `startproc` command:

- UNIX: *ORACLE_HOME*`/opmn/bin> opmnctl [verbose] startproc ias-component=HTTP_Server`
- Windows: *ORACLE_HOME*`\opmn\bin> opmnctl [verbose] startproc ias-component=HTTP_Server`

## Stopping Oracle HTTP Server

To stop Oracle HTTP Server, use the `stopproc` command:

- UNIX: *ORACLE_HOME*`/opmn/bin> opmnctl [verbose] stopproc ias-component=HTTP_Server`
- Windows: *ORACLE_HOME*`\opmn\bin> opmnctl [verbose] stopproc ias-component=HTTP_Server`

## Restarting Oracle HTTP Server

Restarting Oracle HTTP Server performs a graceful restart, which is invisible to clients. In a graceful restart, on UNIX, a `USR1` signal is sent. When the process receives this signal, it tells the children to exit after processing the current request. (Children that are not servicing requests exit immediately.)

The parent re-reads the configuration files and re-opens the log files, replacing the children with new children in accordance with the settings it finds when re-reading the configuration files. It always observes the process creation settings (`MaxClients`, `MaxSpareServers`, `MinSpareServers`) specified, and takes the current server load into account.

To restart Oracle HTTP Server, use the `restartproc` command:

- UNIX: *ORACLE_HOME*`/opmn/bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`
- Windows: *ORACLE_HOME*`\opmn\bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`

> **See Also:** *Oracle Process Manager and Notification Server Administrator's Guide* for more information on `opmnctl` command options.

# 2

# Oracle HTTP Server Concepts

This chapter introduces you to the Oracle HTTP Server directory structure, and configuration files, configuration file syntax, modules, and directives.

Topics discussed are:

- Understanding Oracle HTTP Server Directory Structure
- Accessing Configuration Files
- Configuration Files Syntax
- Understanding Modules
- Classes of Directives
- Scope of Directives
- About .htaccess Files

Documentation from the Apache Software Foundation is referenced when applicable.

> **Note:** Readers using this guide in PDF or hard copy formats will be unable to access third-party documentation, which Oracle provides in HTML format only. To access the third-party documentation referenced in this guide, use the HTML version of this guide and click the hyperlinks.

# Understanding Oracle HTTP Server Directory Structure

Oracle HTTP Server is installed in the *ORACLE_HOME*/Apache directory on UNIX or *ORACLE_HOME*\Apache directory on Windows for configuring modules. For example, the modplsql folder contains the subdirectories necessary to configure and run PL/SQL applications.

The Apache directory is located at the top level under the *ORACLE_HOME*. It contains subdirectories for configuring modules mod_plsql. It also contains a subdirectory called Apache, which is the base directory of Oracle HTTP Server.

# Accessing Configuration Files

Oracle HTTP Server is configured by placing *directives*, which are basically instructions, into text configuration files. Most of the configuration files are located in:

- UNIX: *ORACLE_HOME*/Apache/Apache/conf
- Windows: *ORACLE_HOME*\Apache\Apache\conf

Some of these files are read only once when the server starts or is reloaded, whereas some files are read every time a related file or directory is requested.

The configuration files which are read only once are called *server-wide* configuration files.

> **See Also:** Appendix A, "Oracle HTTP Server Configuration Files" on page A-1

# Configuration Files Syntax

Oracle HTTP Server contains one directive for each line. The back-slash "\" can be used as the last character on a line to indicate that the directive continues onto the next line. There must be no other characters or white space between the back-slash and the end of the line.

Directives in the configuration files are case-insensitive, but arguments to directives are often case-sensitive. Lines which begin with the character "#" are considered comments, and are ignored. Comments may not be included on a line after a configuration directive. Blank lines and white space occurring before a directive are ignored, so you may indent directives for clarity.

## Understanding Modules

Oracle HTTP Server is a modular server. Modules extend the basic functionality of the Web server, and support integration between Oracle HTTP Server and other Oracle Database components. Oracle HTTP Server includes Apache modules as well as Oracle HTTP Server modules.

You can add modules using the `LoadModule` directive. Following is an example of `LoadModule` usage.

```
LoadModule status_module modules/mod_status.so
```

**See Also:** Chapter 7, "Oracle HTTP Server Modules" on page 7-1

## Classes of Directives

Table 2–1 classifies directives according to the context in which they can be used: global, per-server, and per-directory.

***Table 2–1 Classes and Directives***

| Class | Context | Where Used |
|---|---|---|
| global | server configuration | Inside server configuration files, but only outside of container directives (directives such as `VirtualHost` that have a start and end directive). |
| per-server | server configuration, virtual host | Inside server configuration files, both outside (for the main server) and inside `VirtualHost` directives. |
| per-directory | server configuration, virtual host, directory | Everywhere; particularly inside the server configuration files. |

**Note:** In Table 2–1, each class is a subset of the class preceding it. For example, directives from the per-directory class can also be used in the per-server and global contexts, and directives from the per-server class can be used in the global context.

# Scope of Directives

Directives placed in the main configuration files apply to the entire server. If you wish to change the configuration for only a part of the server, you can scope your directives by placing them in specific sections.

The following section discusses the following types of directives:

- Container Directives
- Block Directives

## Container Directives

Container directives specify the scope within which directives take effect. The following container directives are discussed in detail in subsequent sections:

- <Directory>
- <DirectoryMatch>
- <Files>
- <FilesMatch>
- <Location>
- <LocationMatch>
- <Limit>
- <LimitExcept>
- <VirtualHost>

### <Directory>

It is used to enclose a group of directives that apply only to the named directory and subdirectories of that directory. Any directory that is allowed in a directory context may be used. The directory is either the full path to a directory, or a wildcard string. In a wildcard string, ? matches any single character and * matches any sequences of characters. It is important to note that `<Directory />` operated on the whole file system, where as `<Directory dir>` refers to absolute directories. `<Directory>` containers cannot be nested inside each other, but can refer to directories in the document root that are nested.

### <DirectoryMatch>

It should be used when specifying regular expressions, instead of using the tilde form of <Directory> with wildcards in the directory specification. The following two examples have the same result, matching directories starting with web and ending with a number from 1 to 9:

```
<Directory ~/web[1-9]/>
<DirectoryMatch "/web[1-9]/">
```

### <Files>

The <Files *file*> and </Files> directives support access control by filename. It is comparable to the <Directory> and <Location> directives. The directives given within this section can be applied to any object within a base name (the last component of the filename) matching the specified file name. <Files> sections are processed in the order that they appear in the configuration file, after the <Directory> sections, and .htaccess files are read, but before <Location> sections. Note that the <Files> directives can be nested inside <Directory> sections to restrict the portion of the file system to which they apply.

### <FilesMatch>

Provides access control by filename, just as the <Files> directive does. However, it accepts regular expression.

### <Location>

Limits the application of the directives within a block to those URLs specified, rather than to the physical file location like the <Directory> directive. <Location> sections are processed in the order that they appear in the configuration file, after the <Directory> sections and .htaccess files are read, and after the <Files> sections. <Location> accepts wildcard directories and regular expressions with the tilde character.

### <LocationMatch>

Functions in an identical manner to <Location> and you should use it for specifying regular expressions instead of the tilde form of <Location> with wildcards in the location specification.

For example:

```
<LocationMatch "/(extra|special)/data">
```

matches the URLs that contained the /extra/data or /special/data sub string.

### <Limit>

<Limit *method*> defines a block according to the HTTP method of the incoming request. The following example limits the application of the directives that follow scripts that use the specified method:

```
<Limit POST PUT OPTIONS>
  order deny, allow
  deny from all
  allow from 127.0.0.192.168
</Limit>
```

Generally, <Limit> should not be used unless needed. It is useful only for restricting directives to particular methods. <Limit> is frequently used with other containers, and it is contained in any of them.

### <LimitExcept>

Restrict access controls to all HTTP methods except the named ones.

#### <VirtualHost>

Oracle HTTP Server has the capabilities to serve many different Web sites simultaneously. Directives can also be scoped by placing them inside `<VirtualHost>` sections, so that they will only apply to requests for a particular Web site.

Virtual host refers to the practice of maintaining more than one server on one machine, as differentiated by their apparent hostname. For example, it is often desirable for companies sharing a Web server to have their own domain, and Web servers accessible as, for example, `www.oracle1.com` and `www.oracle2.com`, without requiring you to know any extra path information.

Oracle HTTP Server supports both IP-based virtual hosts and name-based virtual hosts. The latter variant is sometimes also called host-based or non-IP virtual hosts.

Each virtual host can have its own name, IP address, and error and access logs. Within a `<VirtualHost>` container, you can set up a large number of individual servers run by a single invocation of the Oracle HTTP Server. With virtual hosting, you can specify a replacement set of the server-level configuration directives that define the main host, and are not allowed in any other container.

### Block Directives

Specify a condition which must be true in order for directives within to take effect.

`<IfModule>` and `<IfDefine>` are block directives rather than container directives because they do not limit the scope of the directives they contain. They define whether Oracle HTTP Server parses the directives inside the block into its configuration, and the directives are ignored once the server is running.

## About .htaccess Files

Oracle HTTP Server enables for decentralized management of configuration through special files places inside the Web tree. The special files are usually called `.htaccess`, but can be specified in the `AccessFileName` directive. Directives placed in `.htaccess` files apply to the directory where you place the file, and all subdirectories. The `.htaccess` files follow the same syntax as the main configuration files. Since `.htaccess` files are read on every request, changes made in these files take immediate effect.

The server administrator further controls what directives may be placed in `.htaccess` files by configuring the `AllowOverride` directive in the main configuration files.

# 3

# Specifying Server and File Locations

This chapter explains how to set Oracle HTTP Server and server administrator options, and specifies file locations.

Topics discussed are:

- Setting Server and Administrator Functions
- Specifying File Locations

Documentation from the Apache Software Foundation is referenced when applicable.

> **Note:** Readers using this guide in PDF or hard copy formats will be unable to access third-party documentation, which Oracle provides in HTML format only. To access the third-party documentation referenced in this guide, use the HTML version of this guide and click the hyperlinks.

# Setting Server and Administrator Functions

The following set basic Oracle HTTP Server and administrator functions. They are located in the "Main Server Configuration" portion of the `httpd.conf` file.

**See Also:** "httpd.conf File Structure" on page A-2

- ServerName
- UseCanonicalName
- ServerAdmin
- ServerSignature
- ServerTokens
- ServerAlias

## ServerName

Enables the server to set a hostname that can be used to create redirection URLs, through which you can access directories without having to use a "/" at the end.

**See Also:** "`ServerName` directive" in the Apache Server documentation.

## UseCanonicalName

Determines which hostname and port to use when redirecting the URL to the same server.

- `on`: This is the default setting. Server uses the hostname and port values set in `ServerName` and `Port`.
- `off`: Server uses the hostname and port that you specify in the request.

**See Also:** "`UseCanonicalName` directive" in the Apache Server documentation.

## ServerAdmin

Creates an email address that is included with every default error message that clients encounter. It is useful to create a separate email address for this.

> **See Also:** "`ServerAdmin` directive" in the Apache Server documentation.

## ServerSignature

Enables the server to recognize which server, among the various proxies, created the returned response, such as an error message.

- on: Server creates a footer to the returned document that includes information such as `ServerName` and server version number. This is the default.
- `email`: Server creates an additional "mailto:" reference to the `ServerAdmin` of the document.
- `off`: Footer and "mailto:" reference are not created.

> **See Also:** "`ServerSignature` directive" in the Apache Server documentation.

## ServerTokens

Controls server information which is returned to clients, such as in error messages. This information includes a description of the generic operating system-type of the server, and compiled-in modules.

- `min(imal)`: provides information such as server name and version.
- `OS`: provides information such as server name, version and operating system.
- `full`: provides information such as server name, version, operating system, and complied modules.

> **See Also:** "`ServerTokens` directive" in the Apache Server documentation.

## ServerAlias

Sets alternate names for the current virtual host.

> **See Also:** "`ServerAlias` directive" in the Apache Server documentation.

# Specifying File Locations

The following directives to control the location of various server files. They are located in the "Global Environment" of the `httpd.conf` file.

> **See Also:** "httpd.conf File Structure" on page A-2

- CoreDumpDirectory
- DocumentRoot
- ErrorLog
- LockFile
- PidFile
- ScoreBoardFile
- ServerRoot

## CoreDumpDirectory

Specifies the directory in which the server dumps core. The default is the `ServerRoot` directory. This directive is applicable to UNIX only.

> **See Also:** "`CoreDumpDirectory` directive" in the Apache Server documentation.

## DocumentRoot

Sets the directory from which httpd serves files. Unless matched by a directive like `Alias`, the server appends the path from the requested URL to the document root to make the path to the document for static content.

> **See Also:** "`DocumentRoot` directive" in the Apache Server documentation.

## ErrorLog

Sets the name of the file to which the server notes any errors it encounters. If the name of the file does not begin with a slash (/), then it is assumed to be relative to the `ServerRoot`. If the name of the file begins with a pipe (|), then it is assumed to be a command to spawn to handle the error log.

> **See Also:** "`ErrorLog` directive" in the Apache Server documentation.

## LockFile

Sets the path to the lockfile used when Oracle HTTP Server is compiled with either `USE_FCNTL_SERIALIZED_ACCEPT` or `USE_FLOCK_SERIALIZED_ACCEPT`. It is recommended that default value be used. The main reason for changing it is if the logs directory is NFS mounted, since the lockfile must be stored on a local disk.

> **See Also:** "`LockFile` directive" in the Apache Server documentation.

## PidFile

Enables you to set and change the location of the `PID` file to which the server records the process identification number. If the filename does not begin with a slash (/), then it is assumed to be relative to the `ServerRoot`.

> **See Also:** "`PidFile` directive" in the Apache Server documentation.

## ScoreBoardFile

Required in some architectures to set a file that the server uses to communicate between the parent and children processes. To verify if your architecture requires a scoreboard file, run Oracle HTTP Server and see if it creates the file named by the directive. If your architecture requires it then you must ensure that this file is not used at the same time by more than one invocation of the server.

> **See Also:** "`ScoreBoardFile` directive" in the Apache Server documentation.

## ServerRoot

Specifies the directory that contains the `conf` and `logs` subdirectories. If the server is started with the `-f` option, then you will have to specify `ServerRoot`.

> **See Also:** "`ServerRoot` directive" in the Apache Server documentation.

# 4

# Managing Server Processes

This chapter provides an overview of the Oracle HTTP Server processes, and provides information on how to regulate, and monitor these processes.

Topics discussed are:

- Oracle HTTP Server Processing Model
- Handling Server Processes
- Limiting the Number of Processes and Connections
- Getting Information about Processes

Documentation from the Apache Software Foundation is referenced when applicable.

> **Note:** Readers using this guide in PDF or hard copy formats will be unable to access third-party documentation, which Oracle provides in HTML format only. To access the third-party documentation referenced in this guide, use the HTML version of this guide and click the hyperlinks.

# Oracle HTTP Server Processing Model

Once Oracle HTTP Server is started, the system is ready to listen for and respond to http(s) requests. The request processing model is different on UNIX and Windows.

After installation, the main httpd parent process, as well as the child processes, run as the user who installed Oracle Database. The `User` and `Group` directive are used to set the privileges for the child processes. These directives are ignored if you are not running as `root`.The child processes must be able to read all the content that will be served.

## Running Oracle HTTP Server as Root

On UNIX, you will hare to run as root if you want to run on ports less than 1024.

In order to run Oracle HTTP Server as `root`, perform the following steps:

1. Shutdown Oracle HTTP Server using the following command:
   - UNIX: *ORACLE_HOME*`/opmn/bin> opmnctl [verbose] stopproc ias-component=HTTP_Server`
   - Windows: *ORACLE_HOME*`\opmn\bin> opmnctl [verbose] stopproc ias-component=HTTP_Server`
2. Change to root user. Navigate to *ORACLE_HOME*`/Apache/Apache/bin` on UNIX or *ORACLE_HOME*`\Apache\Apache\bin` on Windows and execute the following command:

   ```
   chown root .apachectl
   chmod 6750 .apachectl
   ```

3. Exit `root`.
4. Restart Oracle HTTP Server using the following command:
   - UNIX: *ORACLE_HOME*`/opmn/bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`
   - Windows: *ORACLE_HOME*`\opmn\bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`

On Windows, Oracle HTTP Server launches a single parent process and one child process. The child process creates multiple threads that listen and respond to client requests.

You must decide how you want to set Oracle HTTP Server to handle processes or threads.

## Additional Security Considerations

For additional security on UNIX, you can change the user to “nobody”. Be sure that the child processes can accomplish their tasks as the user “nobody”. Change all static content, such as the *ORACLE_HOME*/Apache/Apache/htdocs directory on UNIX or *ORACLE_HOME*\Apache\Apache\htdocs on Windows, so that all the files are readable, but ideally not writable by the user “nobody”. Also, verify that all the CGI and FastCGI programs can be run by user “nobody”.

After making manual configuration changes to DAD passwords, it is recommended that the DAD passwords are obfuscated by running the “dadTool.pl” script located in *ORACLE_HOME*/Apache/modplsql/conf.

> **See Also:** "PlsqlDatabasePassword" on page 7-36 on instructions on performing the obfuscation.

If your PL/SQL application is using the file-system caching functionality in mod_plsql, then the httpd processes should have read and write privileges to the cache directory through the parameter PlsqlCacheDirectory in *ORACLE_HOME*/Apache/modplsql/conf/cache.conf on UNIX or *ORACLE_HOME*\Apache\modplsql\conf\cache.conf on Windows. By default, this parameter points to *ORACLE_HOME*/Apache/modplsql/cache on UNIX or *ORACLE_HOME*\Apache\modplsql\cache on Windows.

Finally, given that the cached content might contain sensitive data, the final contents of the file-system cache should be protected. So, although Oracle HTTP Server might run as “nobody”, access to the system as this user should be well-protected.

> **See Also:** "mod_plsql" on page 7-19

# Handling Server Processes

Use the following directives to manage the server processes:

- ServerType
- Group
- User

## ServerType

Provides the following two options, both being applicable on UNIX only:

**`inetd`**: Starts up a new child process every time a request comes in. The program exits once the request is dealt with. This setting eliminates the option of having several child processes in waiting, making it slower and expensive, but more secure.

**`standalone`**: Enables several waiting child processes, and requires the server to be started only once. It is the default and recommended setting for a busy Web site.

You must specify the `User` and `Group` under which the servers answer requests.

> **See Also:** "`ServerType` directive" in the Apache Server documentation.

## Group

Specifies the group under which the server answers requests. In order to use this directive, the standalone server must be run initially as root. It is recommended that you create a new group for running the server. This is applicable to UNIX only.

> **See Also:** "`Group` directive" in the Apache Server documentation.

## User

Specifies the user ID to which the server answers requests. Run the standalone server as `root` to use this directive.You should have privileges to access files that are available for everyone, and should not be able to execute code which is not meant for httpd requests. It is recommended that you set up a new user for running the server. This is applicable to UNIX only.

> **See Also:** "`User` directive" in the Apache Server documentation.

# Limiting the Number of Processes and Connections

The following directives control and limit the number of child processes or simultaneous requests. They are located in the "Global Environment" of the `httpd.conf` file.

**See Also:** "httpd.conf File Structure" on page A-2

- StartServers
- ThreadsPerChild
- MaxClients
- MaxRequestsPerChild
- MaxSpareServers
- MinSpareServers

## StartServers

Sets the number of child server processes created when Oracle HTTP Server is started. The default is set at 5. This is applicable to UNIX only.

**See Also:** "`StartServers` directive" in the Apache Server documentation.

## ThreadsPerChild

Controls the maximum number of child threads handling requests. The default is set at 50. This is applicable to Windows only.

**See Also:** "`ThreadsPerChild` directive" in the Apache Server documentation.

## MaxClients

Limits the number of requests that can be dealt with at one time. The default and recommended value is 150. This is applicable to UNIX only.

**See Also:** "`MaxClients` directive" in the Apache Server documentation.

## MaxRequestsPerChild

Controls the number of requests a child process handles before it dies. This value should be specified again if the machine is rebooted. If you select the value to be 0, which is the default, then the process will never die. This is applicable to UNIX only.

> **See Also:** "`MaxRequestsPerChild` directive" in the Apache Server documentation.

## MaxSpareServers

Sets the maximum number of idle child server processes. An idle process is one which is running, but not handling a request. The parent process kills off idle child processes that exceed the value set for this directive. The default is set at 10. This is applicable to UNIX only.

> **See Also:** "`MaxSpareServers` directive" in the Apache Server documentation.

## MinSpareServers

Sets the minimum number of idle child server processes. An idle process is one which is running but not handling a request. The parent process will create new children at the maximum rate of one process for each second if there are fewer processes running. The default is set at 5. This is applicable to UNIX only.

> **See Also:** "`MinSpareServers` directive" in the Apache Server documentation.

# Getting Information about Processes

There are several ways to monitor Oracle HTTP Server processes.

1. Use the performance monitor on Windows, or the `ps` utility on UNIX.

   **See Also:** *Oracle Application Server 10g Performance Guide* and your operating system documentation for more information.

2. Use mod_status for server status. By default, it is available from localhost only.

# 5

# Managing the Network Connection

This chapter provides information about specifying IP addresses and ports, and managing server interaction, and network connection persistence.

Topics discussed are:

- Specifying Listener Ports and Addresses
- Managing Interaction Between Server and Network
- Managing Connection Persistence
- Configuring Reverse Proxies and Load Balancers

Documentation from the Apache Software Foundation is referenced when applicable.

> **Note:** Readers using this guide in PDF or hard copy formats will be unable to access third-party documentation, which Oracle provides in HTML format only. To access the third-party documentation referenced in this guide, use the HTML version of this guide and click the hyperlinks.

# Specifying Listener Ports and Addresses

When Oracle HTTP Server is started, by default, it listens for requests on port 7777 (non-SSL). If port 7777 is occupied, Oracle HTTP Server listens on the next available port number between a range of 7777-7877. Thus, if port 7777 is busy, it would listen on port 7778, and so on.

A file named `setupinfo.txt` is automatically generated in *`ORACLE_HOME`*`/Apache/Apache` on UNIX or *`ORACLE_HOME`*`\Apache\Apache` on Windows . It contains port information for Oracle HTTP Server. This file is generated at install time, and is not updated thereafter. If you restart Oracle HTTP Server, the information in this file becomes inaccurate.

You can change the Oracle HTTP Server listener port (SSL and non-SSL) after installation. If you make a port change, then you have to also update other components to use the new port number.

> **See Also:** *Oracle Application Server 10g Administrator's Guide* for complete instruction.

You can specify the server to listen to more than one port, selected addresses, or a combination. The following directives, located in the "Global Environment" of the `httpd.conf` file, specify listener ports and addresses. Note that `BindAddress` and `Port` can be used only once. Apache group recommends the use of `Listen` instead.

- BindAddress
- Listen
- Port

> **See Also:** "httpd.conf File Structure" on page A-2

## BindAddress

Restricts the server to listen to a single IP address. If the argument to this directive is *, then it listens to all IP addresses. This directives has been deprecated. `Listen` offers similar functionality.

> **See Also:** "`BindAddress` directive" in the Apache Server documentation.

## Port

Specifies the **port** of the listener, if no `Listen` or `BindAddress` are present. If `Listen` is present, the `Port` value becomes the default port value that is used when Oracle HTTP Server builds URLs, or other references to itself. Usually, the values of `Port` and `Listen` should match, unless Oracle HTTP Server is fronted by a caching, or proxy server. Then, you can set `Port` to be the port that is being used by the front end server, and `Listen` to the port that Oracle HTTP Server is actually listening to. By doing this, redirects or other URLs generated by Oracle HTTP Server point to the front-end server rather than directly to Oracle HTTP Server.

> **See Also:** "`Port` directive" in the Apache Server documentation.

## Listen

Specifies an IP port that Oracle HTTP Server should listen on. Multiple `Listen` directives can be used to listen on multiple ports. If present, this value will override the value of `Port`. Accordingly, if you have a `Port` value of 7777 and a `Listen` value of 7778, then Oracle HTTP Server only listens on one port, 7778.

> **See Also:** "`Listen` directive" in the Apache Server documentation.

# Managing Interaction Between Server and Network

The following directives are used to specify how the server interacts with the network. They are located in the "Global Environment" of the `httpd.conf` file.

- ListenBackLog
- SendBufferSize
- TimeOut

**See Also:** "httpd.conf File Structure" on page A-2

## ListenBackLog

Specifies the maximum length of the queue of pending connections. This is useful if the server is experiencing a `TCP SYN` overload, which causes numerous new connections that open up but do not complete the task.

**See Also:** "`ListenBackLog` directive" in the Apache Server documentation.

## SendBufferSize

Increases the `TCP` buffer size to the number of bytes specified, thereby improving performance.

**See Also:** "`SendBufferSize` directive" in the Apache Server documentation.

## TimeOut

Sets the maximum time, in seconds, that the server waits for the following:

- The total amount of time it takes to receive a `GET` request.
- The amount of time between receipt of `TCP` packets on a `POST` or `PUT` request.
- The amount of time between `ACKs` on transmissions of `TCP` packets in responses.

The default is set at 300 seconds.

**See Also:** "`TimeOut` directive" in the Apache Server documentation.

# Managing Connection Persistence

The following directives determine how the server handles persistent connections. They are located in the "Global Environment" of the `httpd.conf` file.

- KeepAlive
- KeepAliveTimeout
- MaxKeepAliveRequests

**See Also:**

- *Oracle Application Server 10g Performance Guide*
- "httpd.conf File Structure" on page A-2

## KeepAlive

Enables a single connection to accept multiple requests from the same client. The default is set to "On".

**See Also:** "`KeepAlive` directive" in the Apache Server documentation.

## KeepAliveTimeout

Sets the number of seconds the server waits for a subsequent request before closing a `KeepAlive` connection. Once a request has been received, the timeout value specified by the `TimeOut` directive applies. The default is set at 15 seconds.

**See Also:** "`KeepAliveTimeout` directive" in the Apache Server documentation.

## MaxKeepAliveRequests

Limits the number of requests allowed for each connection when `KeepAlive` is on. If it is set to "0", unlimited requests will be allowed. The default is set at 100.

**See Also:** "`MaxKeepAliveRequests` directive" in the Apache Server documentation.

# Configuring Reverse Proxies and Load Balancers

By default, Oracle Database installs using the local hostname as set up by `ServerName` directive in Oracle HTTP Server. Most Web sites tend to have a specific hostname or domain name for their Web server. However, this is not possible out of the box because with the `ServerName` directive, Oracle HTTP Server is instantiated with the local host.

***Example 5–1 Using Reverse Proxies and Load Balancers with Oracle HTTP Server***

**Domain Name**: www.oracle.com:80 123.456.7.8 (hosted on a reverse proxy, load balancer, or firewall)

**Host Name of Oracle Database Host**: server.oracle.com 123.456.7.9

**ServerName and Port of Oracle Database Host**: server.oracle.com:7777

Make the following changes in the `httpd.conf` file:

```
Port 80
Listen 7777
Listen 80
# Virtual Hosts
# This section is mandatory for URLs that are generated by
# the PL/SQL packages of the Oracle Portal and various other components
# These entries dictate that the server should listen on port
# 7777, but will assert that it is using port 80, so that
# self-referential URLs generated specify www.oracle.com:80
# This will create URLs that are valid for the browser since
# the browser does not directly see the host server.oracle.com.
NameVirtualHost 123.456.7.9:7777
<VirtualHost server.oracle.com:7777>
ServerName www.oracle.com
Port 80
</VirtualHost>
# Since the previous virtual host entry will cause all links
# generated by the Oracle Portal to use port 80, the server.company.com
# server needs to listen on 80 as well since the Parallel Page
# Engine will make connection requests to Port 80 to request the
# portlets.
NameVirtualHost 123.456.7.9:80
<VirtualHost server.oracle.com:80>
ServerName www.oracle.com
Port 80
<VirtualHost>
```

**See Also:** "Running Oracle HTTP Server as Root" on page 4-2 for instructions on running Oracle HTTP Server with ports lesser than 1024.

# 6

# Configuring and Using Server Logs

This chapter discusses Oracle Diagnostic Logging, log formats, and describes various log files and their locations.

Topics discussed are:

- Using Oracle Diagnostic Logging
- Specifying Log Formats
- Specifying Log Level
- Specifying Log Files

Documentation from the Apache Software Foundation is referenced when applicable.

> **Note:** Readers using this guide in PDF or hard copy formats will be unable to access third-party documentation, which Oracle provides in HTML format only. To access the third-party documentation referenced in this guide, use the HTML version of this guide and click the hyperlinks.

# Using Oracle Diagnostic Logging

Oracle offers a new method for reporting diagnostic messages. This new method, Oracle Diagnostic Logging (ODL), presents a common format for diagnostic messages and log files, and a mechanism for correlating all diagnostic messages from various components across Oracle Database. Using ODL, each component logs messages to its own private local repository. A tool called `LogLoader` collects messages from each repository and loads them into a common repository where messages can be viewed as a single log stream, or analyzed in different ways.

You can view Oracle Database diagnostic log files using a text editor.

> **See Also:** *Oracle Application Server 10g Administrator's Guide* for detailed information regarding Oracle Diagnostic Logging.

## Overview

Oracle HTTP Server enables you to choose the format in which you want to generate log messages. You can either continue to generate log messages in the legacy Apache message format, or generate log messages using ODL, which complies with the new Oracle-wide standards for generating log messages.

## Configuring Oracle HTTP Server

To enable Oracle HTTP Server to use ODL, enter the directives specified in the subsequent section in the `httpd.conf` file. Oracle recommends that you enter the directives before any modules are loaded (`LoadModule` directive) in the `httpd.conf` file so that module-specific logging severities are in effect before modules have the opportunity to perform any logging.

### OraLogMode apache | oracle

Enables you to switch between ODL and legacy Apache logging facility.

**Default**: apache

### OraLogSeverity [module_name <msg_type>[:msg_level]

Enables you to set message severity. The message severity specified with this directives is interpreted as the lowest message severity that is desired, and all messages of that severity level and higher will be logged. `OraLogSeverity` may be specified multiple times. It can be specified globally (no module_name) and once for each module for which a module-specific logging severity is desired.

#### module_name

This argument is the internal name of a module, as it appears in the module structure. The `<IfModule>` directive also makes use of this internal name. The module structure derives the module name from the value of the `_FILE_` macro, without path prefix, of the file which defines the module structure. If a module name is not supplied, the `OraLogSeverity` directive is applied globally.

If the module name is specified, then the directive overrides the global directive value of all the messages originating from the specified module. Specifying a module name for a module that does not get loaded generates an error.

#### msg_type

Message types may be specified in upper or lower case, but will appear in the message output in upper case. This parameter must be of one of the following values:

- `INTERNAL_ERROR`
- `ERROR`
- `WARNING`
- `NOTIFICATION`
- `TRACE`

#### msg_level

This parameter must be an integer in the range of 1-32.

Table 6–1 lists some examples of `OraLogSeverity`.

***Table 6–1 Examples of OraLogSeverity***

| OraLogSeverity Example | Action Taken |
|---|---|
| `OraLogSeverity INTERNAL_ ERROR:10` | Logs all messages of type "internal error" of levels 1-10 |
| `OraLogSeverity WARNING:7` | Logs all messages of type "internal error" of all levels<br>Logs all messages of type "error" of all levels<br>Logs all messages of type "warning" of levels 1-7 |
| `OraLogSeverity WARNING` | For messages from other sources:<br>■ Logs all messages of type "internal error" of all levels<br>■ Logs all messages of type "error" of all levels<br>■ Logs all messages of type "warning" of all levels |

### Default

If a message level is not specified, then the level defaults to the lowest severity. If the entire directive is omitted, then the value of the global Apache `LogLevel` directive is used and translated to the corresponding Oracle message type and the lowest level within the corresponding range, as listed in Table 6–2:

***Table 6–2 Apache Log Level and Corresponding Oracle Message Type***

| Apache Log Level | Oracle Message Type |
|---|---|
| `emerg` | `INTERNAL_ERROR:16` |
| `alert` | `INTERNAL_ERROR:32` |
| `crit` | `ERROR:16` |
| `error` | `ERROR:32` |
| `warn` | `WARNING:32` |
| `notice` | `NOTIFICATION:16` |
| `info` | `NOTIFICATION:32` |
| `debug` | `TRACE:32` |

**See Also:** "Specifying Log Level" on page 6-6

### OraLogDir <bus stop dir>

Specifies the path to the directory which contains all log files. This directory must exit.

**Default**:

- UNIX: *ORACLE_HOME*/Apache/Apache/logs/oracle
- Windows: *ORACLE_HOME*\Apache\Apache\logs\oracle

## Specifying Log Formats

LogFormat specifies the information included in the log file, and the manner in which it is written. The default format is the Common Log Format (CLF). The CLF format is: host ident authuser date request status bytes

- host: This is the client domain name or its IP number.
- ident: If IdentityCheck is enabled and the client machine runs identd, then this is the client identity information.
- authuser: This is the user ID for authorized user.
- date: This is the date and time of the request in the <day/month/year:hour:minute:second> format.
- request: This is the request line, in double quotes, from the client.
- status: This is the three-digit status code returned to the client.
- bytes: This is the number of bytes, excluding headers, returned to the client.

# Specifying Log Level

Table 6–3 lists all the different logging levels, their descriptions, and, example messages:

***Table 6–3 Logging Level***

| Logging Level | Description | Example Message |
|---|---|---|
| `Emergency` | Emergencies- system is unusable. | "Child cannot open lock file. Exiting." |
| `Alert` | Action must be taken immediately. | "getpwuid: couldn't determine user name from uid" |
| `Critical` | Critical conditions. | "socket: Failed to get a socket, exiting child" |
| `Error` | Error conditions. | "Premature end of script headers" |
| `Warning` | Warning conditions. | "child process 1234 did not exit, sending another `SIGHUP`" |
| `Notice` | Normal but significant condition. | "httpd: caught `SIGBUS`, attempting to dump core in..." |
| `Information` | Informational messages that describe possible problems and possible solutions to those problems. | "Server seems busy, (you may need to increase `StartServers`, or `Min/MaxSpareServers`)..." |
| `Debug` | Debug-level messages. | "Opening config file..." |

# Specifying Log Files

The log files are discussed in the subsequent sections:

- Access Log
- CustomLog
- Error Log
- PID File
- Piped Log
- Rewrite Log
- Script Log
- SSL Log
- Transfer Log

It is important to periodically rotate the log files by moving or deleting existing logs on a moderately busy server. For this, the server must be restarted after the log files are moved or deleted so that new log files are opened.

**See Also:** "Log Rotation" in the Apache Server documentation.

## Access Log

The server access log records all requests processed by the server. The location and content of the access log is controlled by the `CustomLog` directive. The `LogFormat` directive can be used to simplify the selection of the contents of the logs.

**See Also:** "Access Log" in the Apache Server documentation.

## CustomLog

The `CustomLog` directive is used to log requests to the server. A log format is specified, and the logging can optionally be made conditional on request characteristics using environment variables.

**See Also:** "`CustomLog` directive" in the Apache Server documentation.

## Error Log

The server sends diagnostic information and records error messages to a log file located, by default, in:

- UNIX: *ORACLE_HOME*/Apache/Apache/logs/error_log
- Windows: *ORACLE_HOME*\Apache\Apache\logs\error_log

The file name can be set using the ErrorLog directive.

> **See Also:** "ErrorLog directive" in the Apache Server documentation.

## PID File

When the server is started, it notes the process ID of the parent httpd process to the PID file located by, default, in

- UNIX: *ORACLE_HOME*/Apache/Apache/logs/httpd.pid
- Windows: *ORACLE_HOME*\Apache\Apache\logs\httpd.pid

This filename can be changed with the PidFile directive. The process ID is for use by the administrator for restarting and terminating the daemon. If the process dies (or is killed) abnormally, then it is necessary to kill the children httpd processes.

> **See Also:** "Pid File" in the Apache Server documentation.

## Piped Log

Oracle HTTP Server is capable of writing error and access log files through a pipe to another process, rather than directly to file. This increases the flexibility of logging, without adding code to the main server. In order to write logs to a pipe, replace the file name with the pipe character "|", followed by the name of the executable which should accept log entries on its standard input. Oracle HTTP Server starts the piped-log process when the server starts, and restarts it if it crashes while the server is running.

> **See Also:** "Piped Log" in the Apache Server documentation.

## Rewrite Log

Rewrite Log is necessary for debugging when `mod_rewrite` is used. This log file produces a detailed analysis of how the rewriting engine transforms requests. The level of detail is controlled by the `RewriteLogLevel` directive.

**See Also:** "Rewrite Log" in the Apache Server documentation.

## Script Log

Script Log enables you to record the input to and output from the CGI scripts. This should only be used in testing, and not for live servers.

**See Also:** "Script Log" in the Apache Server documentation.

## SSL Log

When Oracle HTTP Server starts in SSL mode, it creates `ssl_engine_log` and `ssl_request_log` in

- UNIX: *ORACLE_HOME*`/Apache/Apache/logs`
- Windows: *ORACLE_HOME*`\Apache\Apache\logs`

`ssl_engine_log` tracks SSL and protocol issues, where as `ssl_request_log` records user activity. Use the `SSLLogFile` directive to control output.

**See Also:** "Enabling SSL" on page 8-10

## Transfer Log

Transfer Log specifies the file in which to store the log of accesses to the site. If it is not explicitly included in the `conf` file, then no log is generated. The server typically logs each request to a transfer file located, by default, in

- UNIX:*ORACLE_HOME*`/Apache/Apache/logs/access_log`
- Windows: *ORACLE_HOME*`\Apache\Apache\logs\access_log`

The filename can be set using a `CustomLog` directive.

# 7

# Oracle HTTP Server Modules

This chapter describes the **modules** (mods) included in the Oracle HTTP Server. The modules extend the basic functionality of the Web server, and support integration between Oracle HTTP Server and other Oracle Database components.

Documentation from the Apache Software Foundation is referenced when applicable.

> **Note:** Readers using this guide in PDF or hard copy formats will be unable to access third-party documentation, which Oracle provides in HTML format only. To access the third-party documentation referenced in this guide, use the HTML version of this guide and click the hyperlinks.

# List of Modules

Table 7–1 lists all the Oracle HTTP Server modules discussed in this chapter.

***Table 7–1 Oracle HTTP Server Modules***

| Oracle HTTP Server Modules | | | |
|---|---|---|---|
| mod_access | mod_actions | mod_alias | mod_asis |
| mod_auth | mod_auth_anon | mod_auth_db | mod_auth_dbm |
| mod_auth_digest | mod_autoindex | mod_cern_meta | mod_certheaders |
| mod_cgi | mod_define | mod_digest | mod_dir |
| mod_dms | mod_env | mod_example | mod_expires |
| mod_fastcgi | mod_headers | mod_imap | mod_include |
| mod_info | mod_isapi | mod_log_agent | mod_log_config |
| mod_log_referer | mod_mime | mod_mime_magic | mod_mmap_static |
| mod_negotiation | mod_onsint | mod_ossl | mod_perl |
| mod_plsql | mod_proxy | mod_rewrite | mod_setenvif |
| mod_so | mod_speling | mod_status | mod_unique_id |
| mod_userdir | mod_usertrack | mod_vhost_alias | |

## mod_access

Controls access to the server based on characteristics of a request, such as hostname or IP address.

> **See Also:** Module `mod_access` in the Apache Server documentation.

## mod_actions

Enables execution of CGI scripts based on file type or request method.

> **See Also:** Module `mod_actions` in the Apache Server documentation.

## mod_alias

Enables manipulation of URLs in processing requests. It provides mapping between URLs and filesystem paths, and URL redirection capabilities.

> **See Also:** Module `mod_alias` in the Apache Server documentation.

## mod_asis

Enables sending files that contain their own HTTP headers. It is not supported by Oracle.

> **See Also:** Module `mod_asis`" in the Apache Server documentation.

## mod_auth

Enables user authentication with files based user lists.

> **See Also:** Module `mod_auth` in the Apache Server documentation.

## mod_auth_anon

Enables anonymous user access to protected areas (similar to anonymous FTP, where the email addresses can be logged).

> **See Also:** Module `mod_auth_anon` in the Apache Server documentation.

## mod_auth_db

Uses Berkeley DB files to provide user authentication.

This module is disabled in the Oracle HTTP Server and is not supported by Oracle.

## mod_auth_dbm

Uses DBM files to provide user authentication.

This module is not supported by Oracle.

## mod_auth_digest

Uses MD5 Digest Authentication to provide user authentication.

This module is not supported by Oracle.

## mod_autoindex

Generates directory indexes automatically.

> **See Also:** Module `mod_autoindex` in the Apache Server documentation.

## mod_cern_meta

Emulates CERN (Conseil Europeen pour le Recherche Nucleaire) HTTPD metafile semantics. Metafiles are additional HTTP headers that can be produced for each file the server accesses, in addition to the typical set.

This module is not supported by Oracle.

# mod_certheaders

Enables reverse proxies that terminate SSL connections in front of Oracle HTTP Server to transfer information regarding SSL connection, such as SSL client certificate information, to Oracle HTTP Server, and applications running behind Oracle HTTP Server. This information is transferred from the reverse proxy to Oracle HTTP Server using HTTP headers. The information is transferred from the headers to the standard CGI environment variable, which `mod_ossl` or `mod_ssl` populates if the SSL connection is terminated by Oracle HTTP Server. It also enables certain requests to be treated as HTTPS requests even though they are received through HTTP.

Perform the following steps to configure `mod_certheaders`:

1. Configure Oracle HTTP Server to load `mod_certheaders`. To do this, add a `LoadModule` directive to `httpd.conf` file.
   - UNIX: `LoadModule certheaders_module libexec/mod_certheaders.so`
   - Windows: `LoadModule certheaders_module modules/ApacheModuleCertHeaders.dll`
2. Specify which headers should be translated to CGI environment variables. This can be achieved by using the `AddCertHeader` directive. This directive takes a single argument, which is the CGI environment variable that should be populated from a HTTP header on incoming requests. For example, to populate the `SSL_CLIENT_CERT` CGI environment variable, add the following line to `httpd.conf`:

   ```
   AddCertHeader SSL_CLIENT_CERT
   ```

   The `AddCertHeader` directive can be a global setting if it is placed in the base virtual server section of `httpd.conf`. It can be specific to a single virtual host by placing it within a virtual host container, or it can be specific to a set of URIs by placing it within a `<Directory>` or `<Location>` container directive within `httpd.conf`. The combination of these directives are additive, so that for a given URI, all directives that are specific to that URI will be added to any that are specific to that request's virtual host, which will be added to any that is defined for that base virtual host.

Table 7–2 lists all the supported CGI environment variables with their corresponding HTTP header names.

***Table 7–2 CGI Environment Variables with Corresponding Header Names***

| CGI Variable | Header Name | CGI Variable | Header Name |
|---|---|---|---|
| SSL_PROTOCOL | SSL-Protocol | SSL_SESSION_ID | SSL-Session_Id |
| SSL_CIPHER | SSL-Cipher | SSL_CIPHER_EXPORT | SSL-Cipher-Export |
| SSL_CIPHER_ALGKEYSIZE | SSL-Cipher-Algkeysize | SSL_VERSION_LIBRARY | SSL-Version-Library |
| SSL_CLIENT_CERT | SSL-Client-Cert | SSL_VERSION_INTERFACE | SSL-Version-Interface |
| SSL_CLIENT_CERT_ CHAIN_n | SSL-Client-Cert-Chain -n | SSL_CIPHER_USEKEYSIZE | SSL-Cipher-Usekeysize |
| SSL_CLIENT_VERIFY | SSL-Client-Verify | SSL_SERVER_CERT | SSL-Server-Cert |
| SSL_CLIENT_M_VERSION | SSL-Client-M-Version | SSL_SERVER_M_VERSION | SSL-Server-M-Version |
| SSL_CLIENT_M_SERIAL | SSL-Client-M-Serial | SSL_SERVER_M_SERIAL | SSL-Server-M-Serial |
| SSL_CLIENT_V_START | SSL-Client-V-Start | SSL_SERVER_V_END | SSL-Server-V-End |
| SSL_CLIENT_V_END | SSL-Client-V-End | SSL_SERVER_V_END | SSL-Server-V-End |
| SSL_CLIENT_S_DN | SSL-Client-S-DN | SSL_SERVER_S_DN | SSL-Server-S-DN |
| SSL_CLIENT_S_DN_C | SSL-Client-S-DN-C | SSL_SERVER_S_DN_C | SSL-Server-S-DN-C |
| SSL_CLIENT_S_DN_ST | SSL-Client-S-DN-ST | SSL_SERVER_S_DN_ST | SSL-Server-S-DN-ST |
| SSL_CLIENT_S_DN_L | SSL-Client-S-DN-L | SSL_SERVER_S_DN_L | SSL-Server-S-DN-L |
| SSL_CLIENT_S_DN_O | SSL-Client-S-DN-O | SSL_SERVER_S_DN_O | SSL-Server-S-DN-O |
| SSL_CLIENT_S_DN_OU | SSL-Client-S-DN-OU | SSL_SERVER_S_DN_OU | SSL-Server-S-DN-OU |
| SSL_CLIENT_S_DN_CN | SSL-Client-S-DN-CN | SSL_SERVER_S_DN_CN | SSL-Server-S-DN-CN |
| SSL_CLIENT_S_DN_T | SSL-Client-S-DN-T | SSL_SERVER_S_DN_T | SSL-Server-S-DN-T |
| SSL_CLIENT_S_DN_I | SSL-Client-S-DN-I | SSL_SERVER_S_DN_I | SSL-Server-S-DN-I |
| SSL_CLIENT_S_DN_G | SSL-Client-S-DN-G | SSL_SERVER_S_DN_G | SSL-Server-S-DN-G |
| SSL_CLIENT_S_DN_S | SSL-Client-S-DN-S | SSL_SERVER_S_DN_S | SSL-Server-S-DN-S |
| SSL_CLIENT_S_DN_D | SSL-Client-S-DN-D | SSL_SERVER_S_DN_D | SSL-Server-S-DN-D |
| SSL_CLIENT_S_DN_UID | SSL-Client-S-DN-Uid | SSL_SERVER_S_DN_UID | SSL-Server-S-DN-Uid |
| SSL_CLIENT_S_DN_Email | SSL-Client-S-DN-Email | SSL_SERVER_S_DN_Email | SSL-Server-S-DN-Email |
| SSL_CLIENT_I_DN | SSL-Client-I-DN | SSL_SERVER_I_DN | SSL-Server-I-DN |
| SSL_CLIENT_I_DN_C | SSL-Client-I-DN-C | SSL_SERVER_I_DN_C | SSL-Server-I-DN-C |
| SSL_CLIENT_I_DN_ST | SSL-Client-I-DN-ST | SSL_SERVER_I_DN_ST | SSL-Server-I-DN-ST |
| SSL_CLIENT_I_DN_L | SSL-Client-I-DN-L | SSL_SERVER_I_DN_L | SSL-Server-I-DN-L |

*Table 7–2 CGI Environment Variables with Corresponding Header Names*

| CGI Variable | Header Name | CGI Variable | Header Name |
|---|---|---|---|
| SSL_CLIENT_I_DN_O | SSL-Client-I-DN-O | SSL_SERVER_I_DN_O | SSL-Server-I-DN-O |
| SSL_CLIENT_I_DN_OU | SSL-Client-I-DN-OU | SSL_SERVER_I_DN_OU | SSL-Server-I-DN-OU |
| SSL_CLIENT_I_DN_CN | SSL-Client-I-DN-CN | SSL_SERVER_I_DN_CN | SSL-Server-I-DN-CN |
| SSL_CLIENT_I_DN_T | SSL-Client-I-DN-T | SSL_SERVER_I_DN_T | SSL-Server-I-DN-T |
| SSL_CLIENT_I_DN_I | SSL-Client-I-DN-I | SSL_SERVER_I_DN_I | SSL-Server-I-DN-I |
| SSL_CLIENT_I_DN_G | SSL-Client-I-DN-G | SSL_SERVER_I_DN_G | SSL-Server-I-DN-G |
| SSL_CLIENT_I_DN_S | SSL-Client-I-DN-S | SSL_SERVER_I_DN_S | SSL-Server-I-DN-S |
| SSL_CLIENT_I_DN_D | SSL-Client-I-DN-D | SSL_SERVER_I_DN_D | SSL-Server-I-DN-D |
| SSL_CLIENT_I_DN_UID | SSL-Client-I-DN-Uid | SSL_SERVER_I_DN_UID | SSL-Server-I-DN-Uid |
| SSL_CLIENT_I_DN_Email | SSL-Client-I-DN-Email | SSL_SERVER_I_DN_Email | SSL-Server-I-DN-Email |
| SSL_CLIENT_A_SIG | SSL-Client-A-Sig | SSL_SERVER_A_SIG | SSL-Server-A-Sig |
| SSL_CLIENT_A_KEY | SSL-Client-A-Key | SSL_SERVER_A_KEY | SSL-Server-A-Key |

3. `mod_certheaders` can be used to instruct Oracle HTTP Server to treat certain requests as if they were received through HTTPS even though they were received through HTTP. This is useful when Oracle HTTP Server is front-ended by a reverse proxy or load balancer, which acts as a termination point for SSL requests, and forwards the requests to Oracle HTTP Server through HTTPS.

   For load balancers, `mod_certheaders` must be explicitly configured to determine which requests should be treated as HTTPS requests. To do this, use the following directive:

   ```
   SimulateHttps on
   ```

   `SimulateHttps` can be embedded within a virtual host, such as:

   ```
   <VirtualHost localhost:7777>
       SimulateHttps on
       .
       .
       .
   </VirtualHost>
   ```

This tells `mod_certheaders` to treat every request handled by this virtual host as HTTPS, or the directive can be placed within a `<LocationMatch>`, `<Directory>`, or `<DirectoryMatch>` directive container such as:

```
<Location /foo/>
    SimulateHttps on
</Location>
```

This limits it to URLs starting with `/foo/`.

## mod_cgi

Enables the server to run CGI scripts.

**See Also:** Module `mod_cgi` in the Apache Server documentation.

## mod_define

Enables the `Define` directive, which defines a variable that can be expanded on any configuration line. The `Define` directive has the status `Extension`, which means that it is not compiled into the server by default.

This module requires the Extended API (EAPI). Oracle HTTP Server always has EAPI-enabled.

This module is available on UNIX systems only.

## mod_digest

Uses an older version of the MD5 Digest Authentication specification than that used in `mod_auth_digest` to provide user authentication. `mod_digest` probably only works with older browsers.

**See Also:** Module `mod_digest` in the Apache Server documentation.

## mod_dir

Enables the server to perform slash (/) redirects. Directories must contain a trailing slash. If a request for a URL without a trailing slash is received, `mod_dir` redirects the request to the same URL followed by a trailing slash. For example:

```
http://myserver/documents/mydirectory
```

is redirected to

```
http://myserver/documents/mydirectory/
```

**See Also:** Module `mod_dir` in the Apache Server documentation.

## mod_dms

Enables you to monitor performance of site components with Oracle's Dynamic Monitoring Service (DMS).

**See Also:** *Oracle Application Server 10g Performance Guide*

## mod_env

Enables you to control the environment for CGI scripts and SSI (Server Side Includes) pages by passing, setting, and unsetting environment variables.

**See Also:** Module `mod_env` in the Apache Server documentation.

## mod_example

Provides examples and guidance on how to write modules using the Apache API. When implemented, it demonstrates module callbacks triggered by the server.

This module is not supported by Oracle.

## mod_expires

Enables the server to generate Expires HTTP headers, which provide information to the client about document validity. Documents are served from the source if, based on the expiration criteria, the cached copy has expired.

> **See Also:** Module `mod_expires` in the Apache Server documentation.

## mod_fastcgi

Supports the FastCGI protocol, which enables you to maintain a pool of running servers for CGI applications, thereby eliminating start-up and initialization overhead.

> **See Also:** Module `mod_fastcgi` in the Apache Server documentation.

## mod_headers

Enables you to merge, replace, or remove HTTP response headers.

> **See Also:** Module `mod_headers` in the Apache Server documentation.

## mod_imap

Enables server-side image map processing.

This module is not supported by Oracle.

## mod_include

Provides a filter that processes documents for SSI (Server Side Includes) directives.

> **See Also:** Module `mod_include` in the Apache Server documentation.

## mod_info

Summarizes the entire server configuration, including all installed modules and directive settings.

> **See Also:** Module `mod_info` in the Apache Server documentation.

## mod_isapi

Enables serving of Internet Server extensions (such as `.dll` modules).

It is available on the Windows platform only, and is not supported by Oracle.

## mod_log_agent

Enables logging of client user agents. It is deprecated; you should use `mod_log_config` instead of `mod_log_agent`.

This module is not supported by Oracle.

## mod_log_config

Provides configurable, customizable logging of server activities. You can choose the log format, and select or exclude individual requests for logging, based on characteristics of the requests.

> **See Also:** Module `mod_log_config` in the Apache Server documentation.

## mod_log_referer

Enables logging of documents that reference documents on the server. It is deprecated; you should use `mod_log_config` instead of `mod_log_referer`.

> **See Also:** Module `mod_log_referer` in the Apache Server documentation.

## mod_mime

Enables the server to determine the type of a file from its filename, and associate files with handlers for processing.

> **See Also:** Module `mod_mime` in the Apache Server documentation.

## mod_mime_magic

Enables the server to determine the MIME type of a file by examining a few bytes of its content. It is used in cases when `mod_mime` cannot determine a file type. Make sure that `mod_mime` appears before `mod_mime_magic` in the configuration file, so that `mod_mime` processes the files first.

> **See Also:** Module `mod_mime_magic` in the Apache Server documentation.

## mod_mmap_static

Maps a list of files into memory, useful for frequently requested files that are not changed often.

This module is not supported by Oracle.

## mod_negotiation

Enables the server for content negotiation (selection of documents based on the client's capabilities).

> **See Also:** Module `mod_negotiation` in the Apache Server documentation.

# mod_onsint

This module provides integration support with Oracle Notification Service (ONS) and OPMN (Oracle Process Manager and Notification Server).

## Benefits of mod_onsint

`mod_onsint` provides the following functionality:

- Provides a subscription mechanism for ONS notifications within Oracle HTTP Server. This is particularly important on UNIX where Oracle HTTP Server employs a multi-process architecture. In such an architecture, it is not feasible to have an ONS subscriber in each process since there are up to 8192 processes that comprise a single Oracle HTTP Server instance. Instead, `mod_onsint` provides a single process that receives notification for all modules within an Oracle HTTP Server instance.
- Publishes `PROC_READY` ONS notifications so that other components such as OPMN are notified that the listener is up and ready. It also provides information such as DMS metrics and information about how the listener can be contacted. These notifications are sent periodically by `mod_onsint` as long as the Oracle HTTP Server instance is running.
- Provides functionality that enables Oracle HTTP Server to terminate as a single unit if the parent process fails. The parent process is responsible for starting and stopping all of the child processes for an Oracle HTTP Server instance. The failure of the parent process without first shutting down the child processes leaves Oracle HTTP Server in an inconsistent state that can only be fixed by manually killing all of the orphaned child processes. Until this is done, a new Oracle HTTP Server instance cannot be started since the orphaned child processes still occupy the ports Oracle HTTP Server wants to use. `mod_onsint` provides a monitor of the parent process. If it detects that the parent process has died, it kills all of the remaining child processes. When combined with OPMN, this provides restartability for Oracle HTTP Server in the case of a parent process failure. `mod_onsint` ensures that all of the Oracle HTTP Server child processes die, leaving the ports open for a new Oracle HTTP Server instance. OPMN ensures that a new instance is started once the failure of the original instance is detected.

## Implementation Differences for mod_onsint

Due to the difference in architecture of Oracle HTTP Server on UNIX and Windows, the implementation of `mod_onsint` varies slightly on these platforms.

On UNIX, `mod_onsint` spawns a process at module initialization time. This process is responsible for watching the parent process as well as sending and receiving ONS messages. Callback functions from other modules interested in ONS notifications are made in this process. For this information to be shared with other Oracle HTTP Server child processes, the use of an interprocess communication method such as a memory mapped file must be used. If a failure of a parent process is detected on UNIX, a signal is sent to all the other child processes, causing them to shut down.

On Windows, Oracle HTTP Server consists of only two processes, the parent and a multi-threaded child that handles all of the HTTP requests. In this model, `mod_onsint` runs as a thread within the child process. This thread watches the parent process as well as sending and receiving ONS messages. Callback functions from other modules interested in ONS notifications are made in the child process. If a failure of the parent process is detected, the `mod_onsint` terminates the child process, effectively shutting down Oracle HTTP Server.

**See Also:** "Oracle HTTP Server Processing Model" on page 4-2

There is no configuration of `mod_onsint` needed to provide functionality equivalent to that provided with Oracle HTTP Server in Oracle9*i* Application Server, Release 2 (9.0.2), other than the loading of the module. There is only an optional directive called `OpmnHostPort` that can be set. This directive enables you to specify a hostname and port that OPMN should use for pinging the Oracle HTTP Server instance that `mod_onsint` is running in. If `OpmnHostPort` is not specified, `mod_onsint` chooses an HTTP port automatically. However, in certain circumstances, you may want to choose a specific HTTP port and hostname that OPMN should use to ping the listener with.

`OpmnHostPort` takes a single argument which is a `host:port` string that specifies the values to pass to OPMN. For example, the following line would specify that OPMN should use the localhost interface and port 7778 to ping this listener:

```
OpmnHostPort localhost: 7778
```

This directive must be in the global section of the `httpd.conf` file. It cannot be embedded into any virtual host of location container. After installation, an `OpmnHostPort` directive is located in `dms.conf`. It points OPMN to the Oracle HTTP Server "diagnostic port", which is a special localhost only virtual host.

You cannot combine directives using the one-argument syntax with directives using the two-argument syntax. If you use the two-argument syntax, the default for groups without a group-specific secret key is 'disabled'.

## mod_ossl

This Oracle module enables strong cryptography for Oracle HTTP Server. It is a plug-in to Oracle HTTP Server that enables the server to use SSL. It is very similar to the OpenSSL module, `mod_ssl`. However, in contrast to the OpenSSL module, `mod_ossl` is based on the Oracle implementation of SSL, which supports SSL, version 3, and is based on Certicom and RSA Security technology.

**See Also:**

- *Oracle Application Server 10g Security Guide*
- "Using mod_ossl to Authenticate Users" on page 8-10

## mod_perl

This module embeds the Perl interpreter into the Oracle HTTP Server. This eliminates start-up overhead and enables you to write modules in Perl.

---

**Note:** The demonstration script for this module that is shipped with Oracle Database should be disabled in production environments. It is included only to verify that the installation was successful.

---

**See Also:** `mod_perl` Guide

## Database Usage Notes

This section provides information for `mod_perl` users working with databases. It explains how to test a local database connection and set character forms.

### Using Perl to Access the Database

The following section contains information about using Perl to access the database. Perl scripts access databases using the DBI/DBD driver for Oracle. The DBI/DBD driver is part of Oracle Database. It calls Oracle Callable Interface (OCI) to access the databases.

DBI must be enabled in `httpd.conf` for DBI to function. To do this, perform the following steps:

1. Edit `httpd.conf` using a text editor.
2. Search for "`PerlModule Apache: :DBI`".
3. Uncomment the line "`PerlModule Apache: :DBI`".
4. Restart Oracle HTTP Server using the following commands:

- UNIX: *`ORACLE_HOME`*`/opmn/bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`
- Windows: *`ORACLE_HOME`*`\opmn\bin> opmnctl [verbose] restartproc ias-component=HTTP_Server`

Files must be copied to *`ORACLE_HOME`*`/Apache/Apache/cgi-bin`

***Example 7–1 Using Perl to Access the Database***

```
#!<ORACLE_HOME>perl/bin/perl -w
  use DBI;
  my $dataSource = "host=<hostname.domain>;sid=<orclsid>;port=1521";
  my $userName = "scott";
  my $password = "tiger";
  my $dbhandle = DBI->connect("dbi:Oracle:$dataSource", $userName, $password)
    or die "Can't connect to the Oracle Database: $DBI::errstr\n";
  print "Content-type: text/plain\n\n";
  print "Database connection successful.\n";
  ### Now disconnect from the database
  $dbhandle->disconnect
    or warn "Database disconnect failed; $DBI::errstr\n";
  exit;
```

You can access the DBI scripts from the following locations:

```
http://<hostname.domain>:<port>/cgi-bin/<scriptname>
http://<hostname.domain>:<port>/perl/<scriptname>
```

If the script specifies "`use Apache::DBI`" instead of "`use DBI`", then it will only be able to run from
`http://<hostname.domain>:<port>/perl/<scriptname>`.

## Testing Database Connection

The following is a sample Perl script for testing the database connection of a local seed database. To use the script to test another database connection, you must replace `scott/tiger` with the user name and password for the target database.

***Example 7–2 Sample Perl Script For Testing Connection for Local Seed Database***

```
##### Perl script start ######
use DBI;
print "Content-type: text/plain\n\n";
$dbh = DBI->connect("dbi:Oracle:", "scott/tiger", "") || die $DBI::errstr;
 $stmt = $dbh->prepare("select * from emp order by empno")|| die $DBI::errstr;
$rc = $stmt->execute() || die $DBI::errstr;
while (($empno, $name) = $stmt->fetchrow()) { print "$empno $name\n"; }
warn $DBI::errstr if $DBI::err;
die "fetch error: " . $DBI::errstr if $DBI::err;
$stmt->finish() || die "can't close cursor";
$dbh->disconnect() || die "cant't log off Oracle";
##### Perl script End ######
```

## Using SQL NCHAR Datatypes

SQL NCHAR datatypes have been refined in Oracle9*i*, and are now called reliable Unicode datatypes. SQL NCHAR datatypes such as NCHAR, NVARCHAR2 and NCLOB allow you to store any Unicode characters regardless of the database character set. The character set for those datatypes is specified by the national character set, which is either AL16UTF-16 or UTF8.

> **See Also:** Oracle9*i* documentation for more about SQL NCHAR datatypes.

This release of DBD::Oracle supports SQL NCHAR datatypes and provides driver extension functions to specify the character form for data binding. The following script shows an example to access SQL NCHAR data:

***Example 7–3 Sample Script to Access SQLNCHAR Data***

```
# declare to use the constants for character forms
use DBD::Oracle qw(:ora_forms);
# connect to the database and get the database handle
$dbh = DBI->connect( ... );
# prepare the statement and get the statement handle
$sth = $dbh->prepare( 'SELECT * FROM TABLE_N WHERE NCOL1 = :nchar1' );
# bind the parameter of a NCHAR type
$sth->bind_param( ':nchar1', $param_1 );
# set the character form to NCHAR
$sth->func( { ':nchar1' => ORA_NCHAR } , 'set_form' );
$sth->execute;
```

As shown in Example 7–3, the set_form function is provided as a private function that you can invoke with the standard DBI func() method. It takes an anonymous hash that specifies which placeholder should be associated with which character form. The valid values of character form are either ORA_IMPLICIT or ORA_NCHAR. Setting the character form to ORA_IMPLICIT causes the application's bound data to be converted to the database character set, and ORA_NCHAR to the national character set. The default form is ORA_IMPLICIT.

Another function is provided to specify the default character set form as follows:

```
# specify the default form to be NCHAR
$dbh->func( ORA_NCHAR, 'set_default_form' );
```

After this call is made, the form of all parameters is ORA_NCHAR, unless otherwise specified with set_form calls. Note that unlike the set_form function, this is a function on the database handle, so every statement from the database handle with its default form specified has the form of your choice by default.

**set_form** This function sets the character form for parameter(s). Valid forms are either `ORA_IMPLICIT` (default) or `ORA_NCHAR`. The constants are available as: `ora_forms` in `DBD::Oracle`.

***Example 7–4 Sample for set_form***

```
# a declaration example for the constants ORA_IMPLICIT and ORA_NCHAR
use DBD::Oracle qw(:ora_forms);
# set the character form for the placeholder :nchar1 to NCHAR
$sth->func( { ':nchar1' => ORA_NCHAR } , 'set_form' );
# set the character form using the positional index
$sth->func( { 2 => ORA_NCHAR } , 'set_form' );
# set the character form for multiple placeholders at once
$sth->func( { 1 => ORA_NCHAR, 2 => ORA_NCHAR } , 'set_form' );
```

**set_default_form** This function sets the default character form for a database handle.

***Example 7–5 Default Character Form for a Database Handle***

```
$dbh->func( ORA_NCHAR , 'set_default_form' );
```

# mod_plsql

This Oracle module connects the Oracle HTTP Server to an Oracle database, enabling you to create Web applications using Oracle stored procedures.

In order to access a Web-enabled PL/SQL application, configure a PL/SQL Database Access Descriptor (DAD) for `mod_plsql`. A DAD is a set of values that specifies how `mod_plsql` connects to a database server to fulfill an HTTP request. Besides the connect details, a DAD contains important configuration parameters for various operations in the database and for `mod_plsql` in general. Any Web-enabled PL/SQL application which makes use of the PL/SQL Web ToolKit needs to create a DAD to invoke the application.

- Any PL/SQL Application written using the PL/SQL Web ToolKit
- Oracle Application Server Portal

## Creating a DAD

Perform the following steps to create a DAD:

1. Edit the DAD configuration file *ORACLE_HOME*/Apache/modplsql/conf/dads.conf.
2. Add a DAD where the DAD has the following format:
   a. The Oracle HTTP Server <Location> directive which defines a virtual path used to access the PL/SQL Web Application. This directive begins enclosing a group of directives that apply to the named Location.

      For example, the directive <Location /myapp> defines a virtual path called "/myapp" that will be used to invoke a PL/SQL Web Application through a URL like http://host:port/myapp/.

      **Note:** Older versions of mod_plsql were always mounted on a virtual path with a prefix of '/pls'. This restriction is removed in newer versions but might still be a restriction imposed by some of the older PL/SQL applications.

   b. The Oracle HTTP Server "SetHandler" directive which directs Oracle HTTP Server to enable mod_plsql to handle the request for the virtual path defined by the named Location

      ```
      SetHandler pls_handler
      ```

   c. Additional Oracle HTTP Server directives that are allowed in the context of a <Location> directive. Typically, the following directives are used:

      ```
      Order deny,allow
      Allow from all
      AllowOverride None
      ```

   d. One or more mod_plsql specific directives. For example:

      ```
      PlsqlDatabaseUsername        scott
      PlsqlDatabasePassword        tiger
      PlsqlDatabaseConnectString   orcl
      PlsqlAuthenticationMode      Basic
      ```

   e. An Oracle HTTP Server </Location> directive which closes the group of directives for the named Location, and defines a single DAD.

3. Save the edits.
4. Obfuscate the DAD password by running the "`dadTool.pl`" script located in *`ORACLE_HOME`*`/Apache/modplsql/conf`.

   **See Also:** "PlsqlDatabasePassword" on page 7-36 for instructions on performing the obfuscation.

5. Restart the Oracle HTTP Server for the configuration to take effect.

You can create additional DADs by defining other uniquely named `Locations` in `dads.conf`.

This section contains the following topics:



## Configuration Files

`mod_plsql` configuration reside in the following three configuration files:

- plsql.conf
- dads.conf
- cache.conf

### plsql.conf

This file contains the `LoadModule` directive to load `mod_plsql` into Oracle HTTP Server, any global setting for `mod_plsql`, and include directives for `dads.conf` and `cache.conf`. This file is included by the Oracle HTTP Server configuration file *`ORACLE_HOME`*`/Apache/Apache/conf/oracle_apache.conf` on UNIX or *`ORACLE_HOME`*`\Apache\Apache\conf\oracle_apache.conf` on Windows, which itself gets included in the primary Oracle HTTP Server configuration file `httpd.conf`.

**See Also:** "oracle_apache.conf" on page A-5

### dads.conf

This file contains the configuration parameters for the **PL/SQL database access descriptor** (DAD). A DAD is a set of values that specifies how `mod_plsql` connects to a database server to fulfill a HTTP request.

### cache.conf

This file contains the configuration settings for the file system caching functionality implemented in `mod_plsql`. This configuration file is relevant only if PL/SQL applications use the `OWA_CACHE` package to cache dynamically generated content in the file system.

> **See Also:** *Oracle HTTP Server mod_plsql User's Guide* for details on caching functionality in `mod_plsql`.

## Configuration Parameters

Table 7–3 contains a list of `mod_plsql` configuration parameters. They are discussed in detail in later sections.

While specifying a value for a configuration parameter, follow Oracle HTTP Server conventions for specifying values. For instance, if a value has white spaces in it, enclose the value with double quotes. For example: `PlsqlNLSLanguage "TRADITIONAL CHINESE_TAIWAN.UTF8"`

Also, multi-line directives enables you to specify the same directive multiple times in a DAD.

***Table 7–3 mod_plsql Configuration Files and Parameters***

| Configuration File | Parameters |
|---|---|
| plsql.conf | PlsqlDMSEnable<br>PlsqlLogEnable<br>PlsqlLogDirectory<br>PlsqlIdleSessionCleanupInterval |

*Table 7–3 mod_plsql Configuration Files and Parameters (Cont.)*

| Configuration File | Parameters |
|---|---|
| dads.conf | PlsqlAfterProcedure<br>PlsqlAlwaysDescribeProcedure<br>PlsqlAuthenticationMode<br>PlsqlBeforeProcedure<br>PlsqlBindBucketLengths<br>PlsqlBindBucketWidths<br>PlsqlCGIEnvironmentList<br>PlsqlCompatibilityMode<br>PlsqlDatabaseConnectString<br>PlsqlDatabasePassword<br>PlsqlDatabaseUserName<br>PlsqlDefaultPage<br>PlsqlDocumentPath<br>PlsqlDocumentPath<br>PlsqlDocumentProcedure<br>PlsqlDocumentTablename<br>PlsqlErrorStyle<br>PlsqlExclusionList<br>PlsqlFetchBufferSize<br>PlsqlInfoLogging<br>PlsqlMaxRequestsPerSession<br>PlsqlNLSLanguage<br>PlsqlPathAlias<br>PlsqlPathAliasProcedure<br>PlsqlSessionCookieName<br>PlsqlSesssionStateManagement<br>PlsqlTransferMode<br>PlsqlUploadAsLongRaw |

*Table 7–3 mod_plsql Configuration Files and Parameters (Cont.)*

| Configuration File | Parameters |
|---|---|
| cache.conf | PlsqlCacheCleanupTime<br>PlsqlCacheDirectory<br>PlsqlCacheEnable<br>PlsqlCacheMaxAge<br>PlsqlCacheMaxSize<br>PlsqlCacheTotalSize |

### plsql.conf

This file contains the `LoadModule` directive to load `mod_plsql` into the Oracle HTTP Server, global settings for `mod_plsql`, and include directives for `dads.conf` and `cache.conf`.

> **Note:** Refer to `plsql.README` located in *`ORACLE_HOME`*`/Apache/modplsql/conf` for detailed description of `plsql.conf`.

The following section discusses the following parameters that can be specified in `plsql.conf`:

- `PlsqlDMSEnable`
- `PlsqlLogEnable`
- `PlsqlLogDirectory`
- `PlsqlIdleSessionCleanupInterval`

**PlsqlDMSEnable** Enables Dynamic Monitoring Service (DMS) for `mod_plsql`.

| Category | Value |
|---|---|
| Syntax | `PlsqlDMSEnable` *`On/Off`* |
| Default | On |
| Example | `PlsqlDMSEnable On` |

**PlsqlLogEnable** Enables debug level logging for `mod_plsql`.

Debug level logging is meant to be used for debugging purposes only. When logging is enabled, log files are generated at:

- UNIX: *`ORACLE_HOME`*`/Apache/modplsql/logs`
- Windows: *`ORACLE_HOME`*`\Apache\modplsql\logs`

as configured by `PlsqlLogDirectory`. This parameter should be set to "Off" unless recommended by Oracle support to debug problems with `mod_plsql`.

To view more details about the internal processing of `mod_plsql`, set this directive to "On". This causes `mod_plsql` to start logging for every request that is processed. The log files are generated as specified by the `PlsqlLogDirectory` directive.

| Category | Value |
|---|---|
| Syntax | PlsqlLogEnable *`On/Off`* |
| Default | Off |
| Example | PlsqlLogEnable Off |

**PlsqlLogDirectory** Specifies the directory where debug level logs are written out.

Set the directory name of the location where log files should be generated when logging is enabled. To avoid possible confusion about the location of this directory, an absolute path is recommended.

On UNIX, this directory must have write permissions by the owner of the child httpd processes.

| Category | Value |
|---|---|
| Syntax | `PlsqlLogDirectory` *`directory`* |
| Default | None |
| Example | `PlsqlLogDirectory` *`ORACLE_HOME`*`/Apache/modplsql/logs` |

**PlsqlIdleSessionCleanupInterval** Specifies the time (in minutes) in which the idle database sessions should be closed and cleaned by `mod_plsql`.

This directive is used in conjunction with connection pooling of database connections and sessions in `mod_plsql`. When a session is not used for the specified amount of time, it is closed, and freed. This is done so that unused sessions can be cleaned, and the memory is freed on the database side.

Setting this time to a low number helps in faster cleanup of unused database sessions. Be aware that if this number is too low, then this may adversely affect the performance benefits of connection pooling in `mod_plsql`.

If the number of open database sessions is not a concern, you can increase the value of this parameter for best performance. In such a case, if the site is accessed frequently enough that the idle session cleanup interval is never reached for a session, then the DAD configuration parameter `PlsqlMaxRequestsPerSession` can be modified so that it is guaranteed that a pooled database session gets recycled on a regular basis.

For most installations, the default parameter value should suffice.

| Category | Value |
|---|---|
| Syntax | `PlsqlIdleSessionCleanupInterval` *number* |
| Default | 15 (minutes) |
| Example | `PlsqlIdleSessionCleanupInterval 15` |

## dads.conf

This file contains the configuration parameters for the PL/SQL Database Access Descriptor (DAD).

**DAD Parameters** This section describes all the DAD level parameters that can be specified in the `dads.conf` file. Besides these directives, you can also specify additional Oracle HTTP Server directives that can be specified in the context of a `<Location>` directive, such as:

```
Order deny,allow
AllowOverride None
```

The following parameters are discussed in detail in the subsequent sections:

- PlsqlAfterProcedure
- PlsqlAlwaysDescribeProcedure
- PlsqlAuthenticationMode
- PlsqlBeforeProcedure
- PlsqlBindBucketLengths
- PlsqlBindBucketWidths
- PlsqlCGIEnvironmentList
- PlsqlCompatibilityMode
- PlsqlDatabaseConnectString
- PlsqlDatabasePassword
- PlsqlDatabaseUserName
- PlsqlDefaultPage
- PlsqlDocumentPath
- PlsqlDocumentProcedure
- PlsqlDocumentTablename
- PlsqlErrorStyle
- PlsqlExclusionList
- PlsqlFetchBufferSize
- PlsqlInfoLogging
- PlsqlMaxRequestsPerSession
- PlsqlNLSLanguage
- PlsqlPathAlias
- PlsqlPathAliasProcedure
- PlsqlSessionCookieName
- PlsqlSesssionStateManagement
- PlsqlTransferMode
- PlsqlUploadAsLongRaw

**PlsqlAfterProcedure** Specifies the procedure to be invoked after calling the requested procedure. This enables you to put a hook point after the requested procedure is called. This is useful in doing SQL*Traces/SQL Profiles while debugging a problem with the requested procedure. This is also useful when you want to ensure that a specific call be made after running every procedure.

| Category | Value |
|---|---|
| Syntax | `PlsqlAfterProcedure` *`string`* |
| Default | None |
| Example | `PlsqlAfterProcedure portal.mypkg.myafterproc` |

Notes:

- For all purposes, except for debugging, this parameter should be omitted. You could use this parameter to stop SQL Trace/SQL Profiling.
- In older versions of the product, this parameter was called `after_proc`.

**PlsqlAlwaysDescribeProcedure** Specifies whether `mod_plsql` should describe a procedure before trying to execute it. If this is set to "On", then `mod_plsql` will always describe a procedure before invoking it. Otherwise, `mod_plsql` will only describe a procedure when its internal heuristics have interpreted a parameter type incorrectly.

| Category | Value |
|---|---|
| Syntax | `PlsqlAlwaysDescribeProcedure` *`On/Off`* |
| Default | Off |
| Example | `PlsqlAlwaysDescribeProcedure Off` |

Notes:

- For all purposes, except for debugging, you should leave this parameter set to "Off".
- In older versions of the product, this parameter was called `always_desc`.

**PlsqlAuthenticationMode** Specifies the authentication mode to use for allow access through this DAD.

| Category | Value |
|---|---|
| Syntax | `PlsqlAuthenticationMode` *`Basic/SingleSignOn/GlobalOwa/CustomOwa/PerPackageOwa`* |
| Default | Basic |
| Example | `PlsqlAuthenticationMode Basic` |

Notes:

- Most customer applications use Basic Authentication. Custom Authentication modes (`GlobalOwa`, `CustomOwa`, `PerPackageOwa`) are used by very few PL/SQL applications. The `SingleSignOn` mode is supported only for Oracle Application Server releases, and is used by Oracle Application Server Portal and Oracle Application Server Single Sign-On.
- If the DAD is not using the `Basic` authentication, then you must include a valid username/password in the DAD configuration. For the `Basic` mode, if you wish to perform dynamic authentication, the DAD username/password parameters must be omitted.
- In older versions of the product, this configuration parameter was derived from a combination of `enablesso` and `custom_auth`.
  - `enablesso = Yes` translates to `PlsqlAuthenticationMode SingleSignOn`
  - `custom_auth = Global` translates to `PlsqlAuthenticationMode GlobalOwa`
  - `custom_auth = Custom` translates to `PlsqlAuthenticationMode CustomOwa`
  - `custom_auth = PerPackage` translates to `PlsqlAuthenticationMode PerPackageOwa`

  All other combinations translate to `Basic`.

  **See Also:** "Securing Application Database Access through `mod_plsql`" chapter in the *Oracle HTTP Server mod_plsql User's Guide* for more information regarding different authentication modes.

**PlsqlBeforeProcedure** Specifies the procedure to be invoked before calling the requested procedure. This enables you to put a hook point before the requested procedure is called. This is useful in doing SQL*Traces/SQL Profiles while debugging a problem with the requested procedure. This is also useful when you want to ensure that a specific call be made before running every procedure.

| Category | Value |
|---|---|
| Syntax | `PlsqlBeforeProcedure` *`string`* |
| Default | None |
| Example | `PlsqlBeforeProcedure portal.mypkg.mybeforeproc` |

Notes:

- For all purposes, except for debugging purposes, this parameter should be omitted. You could use this parameter to start SQL Trace/SQL Profiling.
- In older versions of the product, this parameter was called `before_proc`.

**PlsqlBindBucketLengths** Specifies the rounding size to use while binding the number of elements in a collection bind. While executing PL/SQL statements, the Oracle database maintains a cache of PL/SQL statements in the shared SQL area, and attempts to reuse the cached statement if the same statement is executed again. Oracle's matching criteria requires that the statement texts be identical, and that the bind variable data types match. Unfortunately, the type match for strings is sensitive to the exact byte size specified, and for collection bindings is also sensitive to the number of elements in the collection. Since `mod_plsql` binds statements dynamically, the odds of hitting the shared cache are low, and it may fill up with near-duplicates and lead to contention for the latch on the shared area. This parameter reduces that effect by bucketing bind lengths to the nearest level.

All numbers specified should be in ascending order. After the last specified size, subsequent bucket sizes will be assumed to be twice the last one.

| Category | Value |
|---|---|
| Syntax | `PlsqlBindBucketLengths` *`number multiline`* |
| Default | 4,20,100,400 |
| Example | `PlsqlBindBucketLengths 4`<br>`PlsqlBindBucketLengths 25`<br>`PlsqlBindBucketLengths 125` |

Notes:

- This parameter is relevant only if you are using procedures with array parameters, and passing varying number of parameters to the procedure.
- The default should be sufficient for most PL/SQL applications.
- To see if this parameter needs to be changed, check the number of versions of a SQL statement in the SQL area.
- Consider using flexible parameter passing to reduce the problem.
- In older versions of the product, this parameter was called `bind_bucket_lengths`.

**PlsqlBindBucketWidths** Specifies the rounding size to use while binding the number of elements in a collection bind. While executing PL/SQL statements, the Oracle database maintains a cache of PL/SQL statements in the shared SQL area, and attempts to reuse the cached statement if the same statement is executed again. Oracle's matching criteria requires that the statement texts be identical, and that the bind variable data types match. Unfortunately, the type match for strings is sensitive to the exact byte size specified, and for collection bindings is also sensitive to the number of elements in the collection. Since `mod_plsql` binds statements dynamically, the odds of hitting the shared cache are low, and it may fill up with near-duplicates and lead to contention for the latch on the shared area. This parameter reduces that effect by bucketing bind widths to the nearest level.

All numbers specified should be in ascending order. After the last specified size, subsequent bucket sizes will be assumed to be twice the last one.

The last bucket width must be equal to or less than 4000. This is due to the restriction imposed by OCI where array bind widths cannot be greater than 4000.

| Category | Value |
|---|---|
| Syntax | `PlsqlBindBucketWidths` *`number multiline`* |
| Default | `32,128,1450,2048,4000` |
| Example | `PlsqlBindBucketWidths 40`<br>`PlsqlBindBucketWidths 400`<br>`PlsqlBindBucketWidths 2000` |

Notes:

- This parameter is relevant only of you are using procedures with array parameters, and passing varying number of parameters to the procedure.
- The default should be sufficient for most PL/SQL applications.
- To see if this parameter needs to be changed, check the number of versions of a SQL statement in the SQL area.
- Consider using flexible parameter passing to reduce the problem.
- In older versions of the product, this parameter was called `bind_bucket_widths`.

**PlsqlCGIEnvironmentList** Specifies overrides and/or additions of CGI environment variables to the default set of environment variables passed down to a PL/SQL procedure. This is a multi-line directive of name-value pairs to be added, overridden or removed. You can only specify one environment variable for each directive.

You can add CGI environment variables from the Oracle HTTP Server environment by specifying the variable name. To remove a CGI environment variable, set it equal to nothing. To add your own name-value pair, use the syntax `myname=myvalue`.

| Category | Value |
|---|---|
| Syntax | `PlsqlCGIEnvironmentList string` *`multiline`* |
| Default | None |
| Example | ■ To add a new environment variable from the Oracle HTTP Server environment:<br>`PlsqlCGIEnvironmentList DOCUMENT_ROOT`<br>■ To remove an environment variable:<br>`PlsqlCGIEnvironmentList MYENVAR2=`<br>■ To override from the Oracle HTTP Server environment:<br>`PlsqlCGIEnvironmentList REQUEST_PROTOCOL=HTTPS`<br>■ To add your own environment variable:<br>`PlsqlCGIEnvironmentList MY_VARNAME=MY_VALUE` |

Notes:

- Environment variables added here are available in the PL/SQL application through the function `owa_util.get_cgi_env`.
- In older versions of the product, this parameter was called `cgi_env_list`.

**PlsqlCompatibilityMode** Specifies the compatibility mode for running `mod_plsql`. This parameter is supported only for Oracle Application Server releases, and is used when you are using `mod_plsql` with an older version of Oracle Application Server Portal. In such situations, if you are running `mod_plsql` against a pre-9.0.2 version of Oracle Application Server Portal, this should be set to 1.

| Category | Value |
|---|---|
| Syntax | `PlsqlCompatibilityMode` *`BitFlag`* |
| Default | 0 |
| Example | `PlsqlCompatibilityMode 1` |

Notes:

- This parameter enables an old bug in `mod_plsql` in which `mod_plsql` incorrectly converted the plus symbol (+) to space characters for document downloads. Enabling the first bit in this flag will make it impossible to download documents that have a plus symbol `(+)` in the document name.

**PlsqlDatabaseConnectString** Specifies the connection to an Oracle database.

| Category | Value |
|---|---|
| Syntax | `PlsqlDatabaseConnectString`<br>*`stringServiceNameFormat/SIDFormat/TNSFormat/NetServiceNameFormat`*, where string can be one of the following based on the second argument:<br>■ `ServiceNameFormat`: `HOST:PORT:SERVICE_NAME` format where `HOST` is the hostname running the database, `PORT` is the port number the TNS listener is listening on, `SERVICE_NAME` is the database service name.<br>■ `SIDFormat`: `HOST:PORT:SID` format where `HOST` is the hostname running the database, `PORT` is the port number the TNS listener is listening on, `SID` is the database SID.<br>■ `TNSFormat`: A valid TNS alias which resolves using Net8 utilities like `tnsping` and SQL*Plus.<br>■ `NetServiceNameFormat`: A valid net service name which resolves to a connect descriptor. A connect descriptor is a specially formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information.<br>If the format argument is not specified, then `mod_plsql` assumes that 'string' is either in the `HOST:PORT:SID` format, or resolvable by Net8. The differentiation between the two is made by the presence of the colon in the specified string.<br>It is recommended that newer DADs do not use the `SIDFormat` syntax. This exists only for backward compatibility reasons. Use the new two argument format for newly created DADs. |
| Default | None |
| Example | ■ `PlsqlDatabaseConnectString` *`myhost.com`*`:1521:`*`myhost`*`.iasdb.inst ServiceNameFormat`<br>■ `PlsqlDatabaseConnectString` *`myhost.com`*`:1521:iasdb SIDFormat`<br>■ `PlsqlDatabaseConnectString` *`myhost_tns`* `TNSFormat`<br>■ `PlsqlDatabaseConnectString cn=oracle,cn=iasdb NetServiceNameFormat`<br>■ `PlsqlDatabaseConnectString (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=`*`myhost.com`*`)( Port= 1521))(CONNECT_DATA=(SID=iasdb))) TNSFormat`<br>■ `PlsqlDatabaseConnectString` *`myhost_tns`*<br>■ `PlsqlDatabaseConnectString` *`myhost.com`*`:1521:iasdb` |

Notes:

- If the database is running in the same Oracle home, or the environment variable "TWO_TASK" is set (called "LOCAL" on Windows NT), this parameter need not be specified.
- If the database is running in a separate Oracle home, then this parameter is mandatory.
- If you have problems connecting to the database:
  - Check the username and password information in the DAD.
  - Make sure that you run "tnsping <*string*>" and execute commands such as:

    sqlplus *DADUsername/DADPassword@<string>*

  - Ensure that TNS_ADMIN is configured properly.
  - Verify that the HOST:PORT:SERVICE_NAME format makes the connection go through.
  - Ensure that the TNS listener and database are up and running.
  - Ensure that you can ping the host from this machine.
- From a mod_plsql perspective, TNSFormat and NetServiceNameFormat are synonymous and denote connect descriptors that are resolved by Net. The TNSFormat is provided as a convenience so that end-users use this to signify that the name resolution happens through the local tnsnames.ora. For situations where the resolution is through an LDAP lookup as configured in sqlnet.ora, it is recommended that the format specifier of NetServiceNameFormat be used.

  If your database supports high availability, for example, RAC database, it is highly recommended that you use the NetServiceNameFormat such that the resolution for the net service name is through LDAP. This enables you to add or remove RAC nodes accessible through mod_plsql by just changing Oracle Internet Directory with the new/deleted node information. In such situations, hard-coding database listener HOST:PORT information in dads.conf or in the local tnsnames.ora is not recommended.
- In older versions of the product, this configuration parameter was called connect_string.

**PlsqlDatabasePassword** Specifies the password to use to log in to the database.

| Category | Value |
|---|---|
| Syntax | `PlsqlDatabasePassword` *`string`* |
| Default | None |
| Example | `PlsqlDatabasePassword tiger` |

After making manual configuration changes to DAD passwords, it is recommended that the DAD passwords are obfuscated by running the "`dadTool.pl`" script located in *`ORACLE_HOME`*`/Apache/modplsql/conf`.

Following are the steps to obfuscate DAD passwords:

1. If necessary, switch user to the Oracle software owner user, typically `oracle` using the following command:

   ```
   $su - oracle
   ```

2. Set the *`ORACLE_HOME`* environment variable to specify the path to the Oracle home directory for the current release and set the `PATH` environment variable to include the directory containing the Perl executable and the location of the `dadTool.pl` script.

   On Bourne, Bash, or Korn Shell:

   ```
   ORACLE_HOME=new_ORACLE_HOME_path;export ORACLE_HOME
   PATH=ORACLE_HOME/Apache/modplsql/conf:ORACLE_HOME/perl/bin:PATH;export PATH
   ```

   On C or tcsh Shell:

   ```
   setenv ORACLE_HOME new_ORACLE_HOME_PATH
   setenv PATH ORACLE_HOME/Apache/modplsql/conf:ORACLE_HOME/perl/bin:PATH
   ```

   On Windows:

   ```
   set PATH=ORACLE_HOME\Apache\modplsql\conf;ORACLE_
   HOME\perl\5.6.1\bin\MSWin32-x86;%PATH%
   ```

   **Note:** The preceding command for Windows should be issued in one line.

3. Set the appropriate shared library path environment variable for your platform.
   - On UNIX platforms, include the *ORACLE_HOME*/lib directory in your shared library path. Table 7–4 shows the appropriate environment variable for each platform.

***Table 7–4 Platform Type and Corresponding Shared Library Path Environment Variable***

| Platform | Environment Variable |
|---|---|
| AIX | LIBPATH |
| HP-UX | SHLIB_PATH |
| Linux, Solaris, and Tru64 UNIX | LD_LIBRARY_PATH |

   For example, to set the SHLIB_PATH environment in the Bourne shell on HP-UX systems, enter the following command:

   ```
   $SHLIB_PATH=$ORACLE_HOME/lib:$SHLIB_PATH;export SHLIB_PATH
   ```

   - On Windows, include $*ORACLE_HOME*/bin in your PATH, for example:

   ```
   set PATH=%ORACLE_HOME%\bin;%PATH%
   ```

4. Change directory to the mod_plsql configuration directory for the current release of Oracle HTTP Server:

   ```
   cd $ORACLE_HOME/Apache/modplsql/conf
   ```

5. Invoke the following Perl script to obfuscate DAD password:

   ```
   perl dadTool.pl -o
   ```

Notes:

- This is a mandatory parameter, except for a DAD that sets PlsqlAuthenticationMode to Basic and uses dynamic authentication.
- For DADs using SingleSignOn authentication, this parameter is the name of the schema owner.
- In older versions of the product, this configuration parameter was called password.

**PlsqlDatabaseUserName** Specifies the username to use to logon to the database.

| Category | Value |
|---|---|
| Syntax | `PlsqlDatabaseUsername` *`string`* |
| Default | None |
| Example | `PlsqlDatabaseUsername scott` |

Notes:

- This is a mandatory parameter, except for a DAD that sets `PlsqlAuthenticationMode` to `Basic` and uses dynamic authentication.
- For DADs using `SingleSignOn` authentication, this parameter is the name of the schema owner.
- In older versions of the product, this configuration parameter was called `username`.

**PlsqlDefaultPage** Specifies the default procedure to call if none is specified in the URL.

| Category | Value |
|---|---|
| Syntax | `PlsqlDefaultPage` *`string`* |
| Default | None |
| Example | `PlsqlDefaultPage myschema.mypackage.home` |

Notes:

- You can also use Oracle HTTP Server Rewrite rules to achieve the same effect as you get by setting this configuration parameter.
- In older versions of the product, this parameter was called `default_page`.

**PlsqlDocumentPath** Specifies a virtual path in the URL that initiates document download form the document table. For example, if this parameter is set to `docs`, then the following URLs will start the document downloading process for URLs of the format:

```
/pls/dad/docs
/pls/plsqlapp/docs
```

| Category | Value |
|---|---|
| Syntax | `PlsqlDocumentPath` *`string`* |
| Default | `docs` |
| Example | `PlsqlDocumentPath docs` |

Notes:

- Omit this parameter for applications that do not perform document uploads or downloads.

  **See Also:** *Oracle HTTP Server mod_plsql User's Guide*

- In older versions of the product, this parameter was called `document_path`.

**PlsqlDocumentProcedure** Specifies the procedure to call when a document download is initiated. This procedure is called to process the download.

| Category | Value |
|---|---|
| Syntax | `PlsqlDocumentProcedure` *`string`* |
| Default | None |
| Example | `PlsqlDocumentProcedure portal.wwdoc_process.process_download` |

Notes:

- Omit this parameter for applications that do not perform document uploads or downloads.

  **See Also:** *Oracle HTTP Server mod_plsql User's Guide*

- In older versions of the product, this parameter was called `document_proc`.

**PlsqlDocumentTablename** Specifies the table in the database to which all documents are uploaded.

| Category | Value |
|---|---|
| Syntax | `PlsqlDocumentTablename` *`string`* |
| Default | None |
| Example | `PlsqlDocumentTablename myschema.document_table` |

Notes:

- Omit this parameter for applications that do not perform document uploads or downloads.

  **See Also:** *Oracle HTTP Server mod_plsql User's Guide*

- In older versions of the product, this parameter was called `document_table`.

**PlsqlErrorStyle** Specifies the Error Reporting Mode for `mod_plsql` errors. This parameter accepts the following values:

- **ApacheStyle**: This is the default mode. In this mode, `mod_plsql` indicates to Oracle HTTP Server the HTTP error that was encountered. Oracle HTTP Server then generates the error page. This can be used with the Oracle HTTP Server `ErrorDocument` directive to produce customized error messages.
- **ModplsqlStyle**: `mod_plsql` generates the error pages, usually a short message indicating the PL/SQL error that was encountered and PL/SQL exception stack, if any. For example:

  ```
  scott.foo PROCEDURE NOT FOUND
  ```

- **DebugStyle**: This mode provides more details than `ModplsqlStyle`. `mod_plsql` provides more details about the URL, parameters and also produces server configuration information. This mode is for debugging purposes only. Do not use this in a production system, since displaying internal server variables could be a security risk.

| Category | Value |
|---|---|
| Syntax | `PlsqlErrorStyle ApacheStyle/ModplsqlStyle/DebugStyle` |
| Default | `ApacheStyle` |
| Example | `PlsqlErrorStyle ModplsqlStyle` |

In older versions of the product, this parameter was called `error_style`.

**PlsqlExclusionList** Specifies a pattern for excluding certain procedures, packages, or schema names from being directly executed from a browser. This is a multi-line directive in which each pattern occupies one line. The pattern is case-insensitive and can accept simple wildcards such as *, ? and [a-z]. The default patterns excluded from direct URL access are: `sys.*`, `dbms_*`, `utl_*`, `owa_*`, `owa.*`, `htp.*`, `htf.*`.

Setting this directive to "`#NONE#`" will disable all protection. This is not recommended for a live site, however, it is sometimes used for debugging purposes.

If this parameter is overridden, the defaults are no longer in effect. In that case, you must explicitly add the default list to the list of excluded patterns.

| Category | Value |
|---|---|
| Syntax | `PlsqlExclusionList string multiline/#NONE#` |
| Default | `dbms_*`<br>`utl_*`<br>`owa_*`<br>`owa.*`<br>`htp.*`<br>`htf.*` |

| Category | Value |
|---|---|
| Example | PlsqlExclusionList sys.*<br>PlsqlExclusionList dbms_*<br>PlsqlExclusionList utl_*<br>PlsqlExclusionList owa_*<br>PlsqlExclusionList owa.*<br>PlsqlExclusionList htp.*<br>PlsqlExclusionList htf.*<br>PlsqlExclusionList myschema.private.*<br>The preceding configuration excludes access to URLs containing sys.*, dbms_*, utl_*, owa_*, owa.*, htp.*, htf.*, myschema.private.* |

Notes:

- Besides the patterns specified with this parameter, `mod_plsql` also disallows any fully qualified procedure names which contain special characters like tabs, newlines, carriage-returns, single-quotes, the reverse slash, the form feed, the open parenthesis, close parenthesis, and space. This cannot be changed.
- To add a pattern to the defaults, you must specify the default list with the pattern you have added (as in the example in the table).
- In older versions of the product, this parameter was called `exclusion_list`.

> **See Also:** *Oracle HTTP Server mod_plsql User's Guide* for more information regarding security.

**PlsqlFetchBufferSize** Specifies the number of rows of content to fetch from the database for each trip, using either `owa_util.get_page` or `owa_util.get_page_raw`.

By default, `mod_plsql` attempts to fetch 200 response lines of output where each line is of 255 bytes. In situations where the response bytes are single-bytes, the response buffer is populated to the maximum and can pack 255*200=51000 bytes for each round trip. However, for responses containing multi-byte data, the byte packing for each row could be less than ideal resulting in lesser bytes getting transferred for each round trip. If your application generates large pages frequently and the response does not fit in one round trip, then consider setting this parameter higher. However, the memory usage for `mod_plsql` will increase.

| Category | Value |
|---|---|
| Syntax | `PlsqlFetchBufferSize` *`number`* |
| Default | 200 |
| Example | `PlsqlFetchBufferSize` 256 |

Notes:

- This parameter is changed only for performance reasons. The minimum value for this parameter is 28, but it is seldom reduced.
- Change this parameter only under the following circumstances:
  - The average response page is large and you want to reduce the number of round-trips `mod_plsql` makes to the database to fetch the response.
  - The character set in use is multi-byte, and you want to compensate for the problem of `get_page` or `get_page_raw` fetching fewer bytes for each row (calculations in the OWA Web ToolKit are character-based and in the case of multi-byte characters, OWA packages assume a worst-case character byte size and do not attempt to pack each row to its maximum).
- In older versions of the product, this parameter was called `response_array_size`.
- In older versions of the product, the default for this parameter was 128.

**PlsqlInfoLogging** Specifies what mode `mod_plsql` should use to do extra performance logging.

The mode is:

**InfoDebug**: This logs more information to the Apache's `error_log`. This is used in conjunction with Apache's "info" logging level. If the Apache's logging level is not at least set to this high, this setting will be ignored.

| Category | Value |
|---|---|
| Syntax | `PlsqlInfoLogging` *`InfoDebug`* |
| Default | Empty |
| Example | `PlsqlInfoLogging InfoDebug` |

This logging setting is useful for debugging problems in your PL/SQL application.

**PlsqlMaxRequestsPerSession** Specifies the maximum number of requests a pooled database connection should service before it is closed and re-opened.

| Category | Value |
|---|---|
| Syntax | `PlsqlMaxRequestsPerSession` *number* |
| Default | 1000 |
| Example | `PlsqlMaxRequestsPerSession 1000` |

Notes:

- This parameter helps relieve memory and resource problems that may occur due to prolonged session reuse by a PL/SQL application.
- This parameter should not need to be changed; the default is sufficient in most cases.
- Setting this parameter to a low number can degrade performance. A case for a lower value might be an infrequently used DAD whose performance is not a concern, and for which limiting the number of requests provides some benefit.
- In older versions of the product, the equivalent to this parameter is `reuse`. Instead of taking a value of "Yes" or "No", the new parameter enables you to have finer control over the connection pool reuse in `mod_plsql`.

**PlsqlNLSLanguage** Specifies the `NLS_LANG` variable for this DAD. This parameter overrides the `NLS_LANG` environment variable. When this parameter is set, the PL/SQL Gateway uses the specified `NLS_LANG` to connect to the database. Once connected, an alter session command is issued to switch to the specified language and territory. If the middle tier character set matches that of the database, then no alter session call is issued by `mod_plsql`.

| Category | Value |
|---|---|
| Syntax | `PlsqlNLSLanguage` *string* |
| Default | None |
| Example | `PlsqlNLSLanguage America_America.UTF8` |

Notes:

- Most applications have PlsqlTransferMode set to CHAR which means that the character set in PlsqlNLSLanguage needs to match the character set of the database. In one special case, where the database and mod_plsql are both using fixed-size character sets, and the character set width matches, the character set can be different. The response character set is always the mod_plsql character set.
- If PlsqlTransferMode is set to RAW, then this parameter can be ignored.
- In older versions of the product, this parameter was called nls_lang.

**PlsqlPathAlias** Specifies a virtual path alias to map to a procedure call. This is application specific.

| Category | Value |
|---|---|
| Syntax | PlsqlPathAlias *string* |
| Default | None |
| Example | PlsqlPathAlias url |

Notes:

- For applications that do not use path aliasing,this parameter may be omitted.

  **See Also:** *Oracle HTTP Server mod_plsql User's Guide* for more details about path aliasing functionality.

- In older versions of the product, this parameter was called pathalias.

**PlsqlPathAliasProcedure** Specifies the procedure to call when the virtual path in the URL matches the path alias as configured by PlsqlPathAlias.

| Category | Value |
|---|---|
| Syntax | PlsqlPathAliasProcedure *string* |
| Default | None |
| Example | PlsqlPathAliasProcedure portal.wwpth_api_alias.process_download |

Notes:

- For applications that do not use path aliasing, this parameter may be omitted.

  **See Also:** *Oracle HTTP Server mod_plsql User's Guide* for more details about path aliasing functionality.

- In older versions of the product, this parameter was called `pathaliasproc`.

**PlsqlSessionCookieName** Specifies the cookie name whenPlsqlAuthenticationMode is set to `SingleSignOn`. This parameter is supported only for Oracle Application Server releases, and is used by the Oracle Application Server Portal and Oracle Application Server Single Sign-On.

| Category | Value |
|---|---|
| Syntax | `PlsqlSessionCookieName` *`cookie_name`* |
| Default | Same as DAD name |
| Example | `PlsqlSessionCookieName mycookie` |

Notes:

- For DADs not using `SingleSignOn` authentication, this parameter can be omitted. In most other cases, the session cookie name should be omitted (and this parameter automatically defaults to the DAD name).
- A session cookie name must be specified only for Oracle Application Server Portal instances that need to participate in a distributed Oracle Application Server Portal environment. For those Oracle Application Server Portal nodes you want to seamlessly participate as a federated cluster, ensure that the session cookie name for all of the participating nodes is the same.
- Independent Oracle Application Server Portal nodes need to use distinct session cookie names.
- In older versions of the product, this configuration parameter was called `sncookiename`.

**PlsqlSesssionStateManagement** Specifies how package and session state should be cleaned up at the end of each `mod_plsql` request.

- Setting this parameter to `StatelessWithResetPackageState` causes `mod_plsql` to call `dbms_session.reset_package_state` at the end of each `mod_plsql` request.
- Setting this parameter to `StatelessWithPreservePackageState` causes `mod_plsql` to call `htp.init` at the end of each `mod_plsql` request. This cleans up the state of session variables in the OWA Web ToolKit. The PL/SQL application is responsible for cleaning up its own session state. Failure to do so causes erratic behavior, in which a request starts recognizing or manipulating state modified in previous requests.
- Setting this parameter to `StatelessWithFastResetPackageState` causes `mod_plsql` to call `dbms_session.modify_package_state(dbms_session.reinitialize)` at the end of each `mod_plsql` request. This API is a lot faster than the mode of `StatelessWithResetPackageState`, and avoids some latch contention issues, but exists only in database versions 8.1.7.2 and higher. This mode uses up slightly more memory than the default mode.

| Category | Value |
|---|---|
| Syntax | `PlsqlSessionStateManagement`<br>`StatelessWithResetPackageState/StatelessWithFastResetPackageState/StatelessWithPreservePackageState` |
| Default | `StatelessWithResetPackageState` |
| Example | `PlsqlSessionStateManagement`<br>`StatelessWithResetPackageState` |

Notes:

- In older versions of the product, this configuration parameter was called `stateful`.
- An older value of `stateful=no` or `stateful=STATELESS_RESET` corresponds to `PlsqlSessionStateManagement StatelessWithResetPackageState`.
- An older value of `stateful=STATELESS_FAST_RESET` corresponds to `PlsqlSessionStateManagement StatelessWithFastResetPackageState`.

- An older value of `stateful=STATELESS_PRESERVE` corresponds to `PlsqlSessionStateManagement StatelessWithPreservePackageState`.

`mod_plsql` does not support stateful mode of operation. To equip PL/SQL applications with stateful behavior, save state in cookies and/or in the database.

**PlsqlTransferMode** Specifies the transfer mode for data from the database back to `mod_plsql`. Most applications use the default value of `CHAR`.

| Category | Value |
|---|---|
| Syntax | `PlsqlTransferMode` *`CHAR/RAW`* |
| Default | `CHAR` |
| Example | `PlsqlTransferMode CHAR` |

Notes:

- This parameter only needs to be changed to enable sending back responses in different character sets from the same DAD. In such a case, the `CHAR` mode is useless, since it always converts the response data from the database character set to the `mod_plsql` character set.
- In older versions of the product, `RAW` transfer mode was not supported.

**PlsqlUploadAsLongRaw** Specifies the extensions to be uploaded as `LONGRAW` data type, as opposed to using the default `BLOB` data type. The default can be overridden by specifying multi-line directives of file extensions for field. A value of '*' in this field causes all documents to be uploaded as `LONGRAW`.

| Category | Value |
|---|---|
| Syntax | `PlsqlUploadAsLongRaw` *`string multiline`* |
| Default | None |
| Example | `PlsqlUploadAsLongRaw jpg`, `PlsqlUploadAsLongRaw gif` |

Notes:

- For applications that do not do document uploads or downloads, this parameter may be omitted.

  **See Also:** *Oracle HTTP Server mod_plsql User's Guide* for more details about upload and download processes and the structure of the restrictions on the document table format.

- In older versions of the product, this parameter was called `upload_as_log_raw`.

### cache.conf

cache.conf file contains the cache settings for `mod_plsql`. This file contains parameters which specify the characteristics of the `mod_plsql` cache system.

**See Also:** This file is relevant only if the PL/SQL Application uses the `OWA_CACHE` packages to cache content in the file system. Extremely few customer applications make use of the `OWA_CACHE` packages.

The following parameters are specified in `cache.conf` file:

- `PlsqlCacheCleanupTime`
- `PlsqlCacheDirectory`
- `PlsqlCacheEnable`
- `PlsqlCacheMaxAge`
- `PlsqlCacheMaxSize`
- `PlsqlCacheTotalSize`

**PlsqlCacheCleanupTime** Specifies the time to start the cleanup of the cache storage.

This setting defines the exact day and time in which cleanup should occur. The frequency can be set as daily, weekly, and monthly.

- To define daily frequency, the keyword "`Everyday`" is used. The cleanup starts everyday at the time defined. For example, `Everyday 2:00`. This causes the cleanup to happen everyday at 2 AM (local time) in the morning.
- To define weekly frequency, the days of the week such as "`Sunday`", "`Monday`", "`Tuesday`", and so on are used. For example, `Wednesday 15:30`. This causes the cleanup to happen every Wednesday at 3:30 PM (local time) in the afternoon.
- To define monthly frequency, the keyword "`Everymonth`" is used. The cleanup starts at the Saturday of the month at the time defined. For example, `Everymonth 23:00`. This causes the cleanup to happen the first Saturday of every month at 11:00 PM (local time) at night.

| Category | Value |
|---|---|
| Syntax | `PlsqlCacheCleanupTime <Sunday-Saturday, Everyday, Everymonth> <hh:mm>` |
| Default | Saturday 23:00 |
| Example | `PlsqlCacheCleanupTime Saturday 23:00` |

**PlsqlCacheDirectory** Specifies the directory where cache files are written out by `mod_plsql`. This directory must exist or else Oracle HTTP Server will not start.

On UNIX, this directory must have write permissions by the owner of the child httpd processes.

| Category | Value |
|---|---|
| Syntax | `PlsqlCacheDirectory <directory>` |
| Default | none |
| Example | `PlsqlCacheDirectory ORACLE_HOME/Apache/modplsql/cache` |

In older versions, this parameter was called "`cache_dir`" and resides in the "[PLSQL Cache]" section of *`ORACLE_HOME`*`/Apache/modplsql/cfg/cache.cfg`.

**PlsqlCacheEnable** Enables `mod_plsql` caching.

| Category | Value |
|---|---|
| Syntax | `PlsqlCacheEnable` *`On/Off`* |
| Default | Off |
| Example | PlsqlCacheEnable `On` |

Notes:

- If you are sure that your application does not make use of the `OWA_CACHE` packages, in the PL/SQL Web Toolkit, then you can choose to disable caching. In such situations, there will be a very minor performance benefit.
- In older versions, this parameter is called "enabled" and resided in the "[PLSQL Cache]" section of *`ORACLE_HOME`*`/Apache/modplsql/cfg/cache.cfg`.

**PlsqlCacheMaxAge** Specifies the maximum time, in days, a cache file can be allowed to reside in a file system cache, after which the cached file will be removed for cache maintenance.

This setting is to ensure that the cache system does not contain old content. This setting removes old cache files and makes space for new ones.

| Category | Value |
|---|---|
| Syntax | `PlsqlCacheMaxAge` *`<number>`* |
| Default | 30 (30 days) |
| Example | `PlsqlCacheMaxAge 30` |

**PlsqlCacheMaxSize** Specifies the maximum possible size of a cache file.

This setting is to prevent the case in which one file can fill up the entire cache. In general, it is recommended that this be set to about 1-3 percent of the total cache size.

| Category | Value |
|---|---|
| Syntax | `PlsqlCacheMaxSize <number>` |
| Default | 1048576 (1 MB) |
| Example | `PlsqlCacheMaxSize 1048576` |

In older versions, this parameter was called "`max_size`" and resided in the "[PLSQL Cache]" section of *`ORACLE_HOME`*`/Apache/modplsql/cfg/cache/cfg`.

**PlsqlCacheTotalSize** Specifies the total size of the cache directory.

This setting limits the amount of space the cache is allowed to use. Both PLSQL cache and Session Cookie cache share this cache space. Note that this setting is not a hard limit. It might exceed the limit temporarily during normal processing. This is normal behavior.

The cleanup algorithm uses this setting to determine how much to reduce the cache files. Therefore, the real space limit is the physical storage's available size.

This parameter takes bytes as values;

- 1 megabytes = 1048576 bytes
- 10 megabytes = 10485760 bytes

| Category | Value |
|---|---|
| Syntax | `PlsqlCacheTotalSize <number>` |
| Default | 20971520 (20 MB) |
| Example | `PlsqlCacheTotalSize 20971520` |

In older versions, this parameter was called "`total_size`" and resided in the "[PLSQL Cache]" section of *`ORACLE_HOME`*`/Apache/modplsql/cfg/cache/cfg`.

# mod_proxy

This module provides proxy capability for `FTP`, `CONNECT` (for SSL), HTTP/0.9, HTTP/1.0, and HTTP/1.1.

**See Also:**

- Module `mod_proxy` in the Apache Server documentation.
- "Using mod_proxy Directives" on page 8-30

# mod_rewrite

Oracle HTTP Server provides `mod_rewrite` as a tool for URL manipulation. A rewriting engine based on a regular-expression parser is used by `mod_rewrite` to rewrite requested URLs. The granularity of URL manipulations can be affected by the formats of server variables, environment variables, HTTP headers, and time stamps.

This module operates on the full URLs (including the path-info part) both in per-server context (`httpd.conf`) and per-directory context (`.htaccess`) and can generate query-string parts on result.

The following topics are discussed in the subsequent sections:

- mod_rewrite Rules Processing
- mod_rewrite Directives
- Rewrite Rules Hints
- Redirection Examples

## mod_rewrite Rules Processing

Apache processes HTTP in phases. A hook for each of these phases is provided by the Apache API. `mod_rewrite` uses two of these hooks- the URL-to-filename translation hook which is used after the HTTP request has been read but before any authorization starts, and the Fixup hook which is triggered after the authorization phases and after the per-directory configuration files (`.htaccess`) have been read, but before the content handler is activated.

`mod_rewrite` reads the configured rulesets from its configuration structure. Server level rulesets are best configured at startup, while directory level rulesets are configured during the directory access of the kernel.

mod_rewrite loops through the ruleset rule by rule (RewriteRule directive) and when a particular rule matches, it loops through corresponding conditions (RewriteCond directives). First the URL is matched against the Pattern of each rule. When it fails, mod_rewrite looks for corresponding rule conditions. If none are present, it just substitutes the URL with a new value which is constructed from the string Substitution and goes on with its rule-looping. But if conditions exist, it starts an inner loop for processing them in the order that they are listed.

For conditions, a string TestString is created by expanding variables, back-references map lookups, and then CondPattern is matched against the expanded TestString. If the pattern does not match, the complete set of conditions and the corresponding rule fails. If the pattern matches, then the next condition is processed until no more conditions are available. If all conditions match, processing is continued with substituting the URL using Substitution.

When request seeks a URL with more than one slash (/), for example, http://*yourserver*//oldpath/rqstdrsrc, the "//oldpath" may bypass RewriteCond and RewriteRule directives if they are not correctly written.

For example, consider the following rule:

```
RewriteRule ^/oldpath(.*) /newpath$1 [R]
```

Requesting http://*yourserver*/oldpath/files will redirect and return the page http://*yourserver*/newpath/files as expected.

However, requesting http://*yourserver*//oldpath/files will bypass this particular rule, potentially serving a page that you were not expecting it to. You can work around the problem by making sure that rules will capture more than one slash (/). To fix the earlier example, you should use this replacement:

```
RewriteRule ^/+somepath(.*) /otherpath$1 [R]
```

## mod_rewrite Directives

This section discusses the following `mod_rewrite` directives:

- RewriteEngine
- RewriteOptions
- RewriteLog
- RewriteLogLevel
- RewriteBase

**RewriteEngine** Enables or disables the runtime rewriting engine. If it is set to "Off", this module does no runtime processing at all. Use this directive to disable the module instead of commenting out all the `RewriteRule` directives.

Rewrite configurations are not inherited by default. This means that you need to have `ReWriteEngine On` directive for each virtual host in which you want to use it.

**RewriteOptions** By specifying `RewriteOptions` 'inherit', you can force the configuration of the parent by the children. In virtual-server context this means that the maps, conditions and rules of the main server are inherited. In directory context this means that conditions and rules of the `.htaccess` configuration of the parent directory are inherited.

**RewriteLog** Sets the name of the file to which the server logs any rewriting action that it performs. If the name does not begin with a slash (/), then it is assumed to be relative to the `Server Root`. To disable logging, either remove or comment out the `RewriteLog` directive or use `RewriteLogLevel 0`. Avoid setting the filename to `/dev/null` to prevent logging. This can slow down the server with no advantage.

**RewriteLogLevel** Sets the verbosity level of the rewriting log file. The default level 0 means no logging, while 9 or more means that practically all actions are logged.

**RewriteBase** Explicitly sets the base URL for pre-directory rewrites. Rewrite rule can be used in per-directory configuration (`.htaccess`) files. When a substitution occurs for a new URL, the base URL should be added into the server processing. To be able to do this, the module needs to know what the corresponding URL-prefix or URL-base is. By default, this prefix is the corresponding file path itself. However, at most Web sites, URLs are not directly related to physical filename paths. In such cases, you have to use the `RewriteBase` directives to specify the correct URL-prefix.

If the URLs of your Web server are not directly related to physical file paths, you have to use `RewriteBase` in every `.htaccess` files where you want to use `RewriteRule` directives.

***Example 7–6 RewriteBase Directive***

Assume the following per-directory configuration file:

```
## /abc/def/.htaccess - - per-dir config file for directory /abc/def
 # /abc/def is the physical path of /xyz,
RewriteEngine On
RewriteBase /xyz
RewriteRule ^oldstuff\.html$ newstuff.html
```

In Example 7–6, a request to `/xyz/oldstuff.html` gets correctly rewritten to the physical file `/abc/def/newstff.html`.

## Rewrite Rules Hints

Table 7–5 provide hints for using rewrite rules.

***Table 7–5 Rewrite Rules Hints***

| Value | Definition |
|---|---|
| . | Any single character |
| [char] | Any character listed within a square bracket |
| b* | Any character b any number of times |
| .* | Any character any number of times |

For example, if you want to redirect requests from `/demo1`, `/demo2`, and `/demo3` to `/alldemos`, write the rewrite rule as one of the following:

```
RewriteRule /demo. /alldemos [R]
```

or

```
RewriteRule /demo [123] /alldemos [R]
```

If you intend that `/DemoA`, `/DemoB`, and `/DemoC` to be redirected to `/alldemos`, add NC (no case) to the preceding rewrite rules, such as:

```
RewriteRule /demo [123] /alldemos [R, NC]
```

This rewrite rule will not work to redirect from `/demonstration1` to `/demos`, because "." works form one character only. To enable redirection of all URLs beginning with "demo", irrespective of subsequent characters, use the rewrite rule as follows:

```
RewriteRule ^/demo* /alldemos [R, NC]
```

In the preceding example, ^ means the beginning, * means any character after demo.

If there was a request for `/demo1/not_just_index.html`, all the preceding rewrite rules would have redirected the request the request to `/alldemos/index.html`, that may not be what you want. It is quite possible that you may want to redirect to the corresponding files in `/alldemos`, as listed in Table 7–6.

***Table 7–6 Request Redirection***

| Request for | Redirected to |
|---|---|
| `/demo1/happy.html` | `/alldemos/happy.html` |
| `/demo1/go.jpg` | `/alldemos/go.jpg` |
| `/demos1/lucky.jpg` | `/alldemos/lucky.jpg` |

Then you have to use substitution in your rewrite rule as follows:

```
RewriteRule ^/demos1(.*)$ //alldemos/$1 [R NC]
```

The explanation for this rule is:

Take the value of the expression, such as `happy.html`, `go.jpg`, and `lucky.jpg`, that appears after `demo1` as variables ($1) and substitute it after `/alldemos/`.

> **See Also:** Module `mod_rewrite` in the Apache Server documentation.

## Redirection Examples

For redirecting requests from the `DocumentRoot` to a directory called `newroot`, set the following `mod_rewrite` directives:

```
RewriteEngine On
RewriteRule ^/(.*)$ /newroot/$1 [R]
```

For directing requested for files from one directory (`olddir`) to another (`newdir`), set the following directives:

```
RewriteEngine On
RewriteRule ^/olddir(.*)$ /newdir/$1 [R]
```

In each of these cases, you should ensure that the requested resources are indeed available in the redirected location. The `mod_rewrite` module does not ensure the existence of the requested resource in the new location.

For disabling all requests using the HTTP TRACE method, set the following mod_rewrite directives:

```
RewriteEngine On
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
```

## mod_setenvif

This module enables you to set environment variables based on characteristics of a request.

> **See Also:** Module mod_setenvif in the Apache Server documentation.

## mod_so

This module loads executable code and modules into the server at start-up time.

> **See Also:** Module mod_so in the Apache Server documentation.

## mod_speling

This module attempts to correct misspelled or miscapitalized URLs.

> **See Also:** Module mod_speling in the Apache Server documentation.

## mod_status

This module displays an HTML page of server activity and performance.

> **See Also:** Module mod_status in the Apache Server documentation.

## mod_unique_id

This module creates a unique ID for each request.

> **See Also:** Module `mod_unique_id` in the Apache Server documentation.

This module is available on UNIX systems only.

## mod_userdir

This module maps requests to user-specific directories.

> **See Also:** Module `mod_userdir` in the Apache Server documentation.

## mod_usertrack

This module tracks user activity by creating a log.

> **See Also:** Module `mod_usertrack` in the Apache Server documentation.

## mod_vhost_alias

This module enables dynamically configured mass virtual hosting.

> **See Also:** Module `mod_vhost_alias` in the Apache Server documentation.

# 8

# Managing Security

This chapter provides an overview of Oracle HTTP Server security features and configuration information for setting up a secure Web site using them.

Topics discussed are:

- About Oracle HTTP Server Security
- Classes of Users and Their Privileges
- Resources Protected
- Authentication and Authorization Enforcement
- Security Services Implemented Within Oracle HTTP Server

> **See Also:** For additional information about security, refer to *Oracle Application Server 10g Security Guide* provides an overview of Oracle Database security and its core functionality.

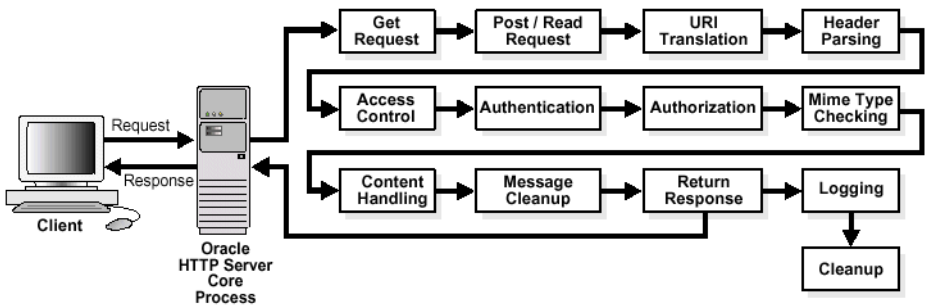
# About Oracle HTTP Server Security

Security can be organized into the three categories of authentication, authorization, and confidentiality. Oracle HTTP Server provides support for all three of these categories. It is based on the Apache Web server, and its security infrastructure is primarily provided by the Apache modules, `mod_auth` and `mod_access`, and the Oracle modules, `mod_ossl`. `mod_auth` provides authentication based on user name and password pairs, `mod_access` controls access to the server based on the characteristics of a request, such as hostname or IP address, `mod_ossl` provides confidentiality and authentication with **X.509** client certificates over SSL.

Based on the Apache model, Oracle HTTP Server provides access control, authentication, and authorization methods that can be configured with access control directives in the `httpd.conf` file. When URL requests arrive at Oracle HTTP Server, they are processed in a sequence of steps determined by server defaults and configuration parameters. The steps for handling URL requests are implemented through a module or plug-in architecture that is common to many Web listeners.

Figure 8–1 shows how URL requests are handled by the server. Each step in this process is handled by a server module depending on how the server is configured. For example, if basic authentication is used, then the steps labeled "Authentication" and "Authorization" in Figure 8–1 represent the processing of the `mod_auth` module.

***Figure 8–1 Steps for Handling URL Requests in Oracle HTTP Server***



## Classes of Users and Their Privileges

Oracle HTTP Server authorizes and authenticates users before allowing them to access, or modify resources on the server. Following are two classes of users that access the server using Oracle HTTP Server, and their privileges.

- Users that access the server without providing any authentication. They have access to unprotected resources only.
- Users that have been authenticated and potentially authorized by modules within Oracle HTTP Server. This includes users authenticated by `mod_auth` and `mod_ossl`. Such users have access to URLs defined in `http.conf` file.

**See Also:** "Authentication and Authorization Enforcement" on page 8-4

## Resources Protected

Oracle HTTP Server is configured to protect resources such as:

- Static content such as static HTML pages, graphics interchange format, `.gif`, files, and other static files that Oracle HTTP Server provides directly.
- CGI/FastCGI scripts, simple scripts or programs that Oracle HTTP Server invokes directly.
- Content generated by modules within Oracle HTTP Server. Modules such as `mod_perl`, `mod_dms` generate responses that are returned to the client.

# Authentication and Authorization Enforcement

Oracle HTTP Server provides user authentication and authorization at two stages:

- **Host-based Access Control (stage one)**: This is based on the details of the incoming HTTP request and its headers, such as IP addresses or host names.
- **User Authentication and Authorization (stage two)**: This is based on different criteria depending on the HTTP server configuration. The server can be configured to authenticate users with user name and password pairs that are checked against a list of known users and passwords.

## Host-based Access Control

Early in the request processing cycle, access control is applied, which can inhibit further processing based on the host name, IP address, or other characteristics such as browser type. You use the `deny`, `allow`, and `order` directives to set this type of access control. These restrictions are configured with Oracle HTTP Server configuration directives and can be based on particular files, directories, or URL formats using the `<Files>`, `<Directory>`, and `<Location>` container directives as shown in the Example 8–1:

***Example 8–1 Host-based Access Control***

```
<Directory /internalonly/>
  order deny, allow
  deny from all
  allow from 192.168.1 us.oracle.com
</Directory>
```

In Example 8–1, the `order` directive determines the order in which Oracle HTTP Server reads the conditions of the `deny` and `allow` directives. The `deny` directive ensures that all requests are denied access. Then, using the `allow` directive, requests originating from any IP address in the 192.168.1.* range, or with the domain name `us.oracle.com` are allowed access to files in the directory `/internalonly/`. It is common practice to specify both `allow` and `deny` in host-based authentication to make the access policy explicit.

If you want to match objects at the file system level, then you must use <Directory> or <Files>. If you want to match objects at the URL level, then you must use <Location>.

> **Note:** Allowing or restricting access based on a host name for Internet access is not considered a very good method of providing security, because host names are easy to spoof. While the same is true of IP addresses, sabotage is more difficult. However, setting access control with intranet IP address ranges is reasonable because the same risks do not apply. This assumes that your firewalls have been properly configured.

## Access Control for Virtual Hosts

To set up access control for virtual hosts, place the AccessConfig directive inside a virtual host container in the server configuration file, httpd.conf. When used in a virtual host container, the AccessConfig directive specifies an access control policy contained in a file. Example 8–2 shows an excerpt from an httpd.conf file which provides the syntax for using AccessConfig this way:

***Example 8–2 Using AccessConfig to Set Up Access Control***

```
...
<VirtualHost ip_address_of_host.some_domain.com>
  ... virtual host directives ...
  AccessConfig conf/access.conf
</VirtualHost>
```

### Using mod_access and mod_setenvif for Host-based Access Control

Using host-based access control schemes, you can control access to restricted areas based on where HTTP requests originate. Oracle HTTP Server uses `mod_access` and `mod_setenvif` to perform host-based access control. `mod_access` provides access control based on client hostname, IP address, or other characteristics of the client request, and `mod_setenvif` provides the ability to set environment variables based upon attributes of the request. When you enter configuration directives into the `httpd.conf` file that use these modules, the server fulfills or denies requests based on the address or name of the host, or based on the HTTP request header contents.

You can use host-based access control to protect static HTML pages, applications, or components.

Oracle HTTP Server supports four host-based access control schemes:

- Controlling Access by IP Address
- Controlling Access by Domain Name
- Controlling Access by Network or Netmask
- Controlling Access with Environment Variables

All of these allow you to specify the machines from which access to protected areas is granted or denied. Your decision to choose one or more of the host-based access control schemes is determined by which scheme most efficiently protects your restricted content and applications, or which scheme is easiest to maintain.

**Controlling Access by IP Address** Controlling access with IP addresses is a preferred method of host-based access control. It does not require DNS lookups that consume time, system resources, and make your server vulnerable to DNS spoofing attacks.

***Example 8–3 Controlling Access by IP Address***

```
<Directory /secure_only/>
  order deny,allow
  deny from all
  allow from 207.175.42.*
</Directory>
```

In Example 8–3, requests originating from all IP addresses except 207.175.42.* range are denied access to the `/secure_only/` directory.

**Controlling Access by Domain Name** Domain name-based access control can be used with IP address-based access control to solve the problem of IP addresses changing without warning. When you combine these methods, if an IP address changes, then the secure areas of your site are still protected because the domain names you want to keep out will still be denied access.

To combine domain name-based with IP address-based access control, use the syntax shown in Example 8–4:

***Example 8–4 controlling Access by Domain Name***

```
<Directory /co_backgr/>
  order allow,deny
  allow from all
  # 141.217.24.* is the IP for malicious.cracker.com
  deny from malicious.cracker.com 141.217.24.*
</Directory>
```

In Example 8–4, all requests for directory `/co_backgr/` are accepted except those that originate from the domain name `malicious.cracker.com` or the IP address 141.217.24.* range. Although this is not a fool proof precaution against domain name or IP address spoofing, it protects your site from `malicious.cracker.com` even if they change their IP address.

**Controlling Access by Network or Netmask** You can control access based on subsets of networks, specified by IP address. The syntax is shown in Example 8–5:

***Example 8–5 Controlling Access by Network or Netmask***

```
<Directory /payroll/>
  order deny,allow
  deny from all
  allow from 10.1.0.0/255.255.0.0
</Directory>
```

In Example 8–5, access is allowed from a network/netmask pair. A netmask shows how an IP address is to be divided into network, subnet, and host identifiers. Netmasks enable you to refer to only the host ID portion of an IP address.

The netmask in Example 8–5, 255.255.0.0, is the default netmask setting for a Class B address. The binary ones (decimal 255) mask the network ID and the binary zeroes (decimal 0) retain the host ID of a given IP address.

**Controlling Access with Environment Variables** You can use arbitrary environment variables for access control, instead of using IP addresses or domain names. Use `BrowserMatch` and `SetEnvIf` directives for this type of access control.

> **Note:** Typically, `BrowserMatch` and `SetEnvIf` are not used to implement security policies. Instead they are used to provide different handling of requests based on browser types and versions.

Use `BrowserMatch` when you want to base access on the type of browser used to send a request. For instance, if you want to allow access only to requests that come from a Netscape browser, then use the syntax shown in Example 8–6:

***Example 8–6 Controlling Access with Environment Variables***

```
BrowserMatch ^Mozilla netscape_browser
<Directory /mozilla-area/>
  order deny,allow
  deny from all
  allow from env=netscape_browser
</Directory>
```

Use `SetEnvIf` when you want to base access on header information contained in the HTTP request. For instance, if you want to deny access from any browsers using HTTP version 1.0 or earlier, then use the syntax shown in Example 8–7:

***Example 8–7 Controlling Access with SetEnv***

```
SetEnvIf Request_Protocol ^HTTP/1.1 http_11_ok
<Directory /http1.1only/>
  order deny,allow
  deny from all
  allow from env=http_11_ok
</Directory>
```

**See Also:** "Scope of Directives" on page 2-4

## User Authentication and Authorization

Basic authentication prompts for a user name and password before serving an HTTP request. When a browser requests a page from a protected area, Oracle HTTP Server responds with an unauthorized message (status code 401) containing a `WWW-Authenticate:` header and the name of the realm configured by the configuration directive, `AuthName`. When the browser receives this response, it prompts for a user name and password. After the user enters a user name and password combination, the browser sends this information back to the server in an Authorization header. In the authorization header message, the user name and password are encoded as a base 64 encoded string.

User authorization involves checking the authenticated user against an access control list that is associated with a specific server resource such as a file or directory. To configure user authorization, place the `require` directive in the `httpd.conf` file, usually within a virtual host container. User authorization is commonly used in combination with user authentication. After the server has authenticated a user's user name and password, then the server compares the user to an access control list associated with the requested server resource. If Oracle HTTP Server finds the user or the user's group on the list, then the resource is made available to that user.

### Using mod_auth to Authenticate Users

User authentication is based on user names and passwords that are checked against a list of known users and passwords. These user name and password pairs may be stored in a variety of forms, such as a text file, database, or directory service. Then configuration directives are used in `httpd.conf` to configure this type of user authentication on the server. `mod_auth` uses the `AuthUserFile` directive to set up basic authentication. It supports only files.

Any authentication scheme that you devise requires that you use a combination of the configuration directives listed in Table 8–1.

***Table 8–1 Directives Descriptions***

| Directive Name | Description |
|---|---|
| `AuthName` | Defines the name of the realm in which the user names and passwords are valid. Use quotation marks if the name includes spaces. |
| `AuthType` | Specifies the authentication type. Most authentication modules use basic authentication, which transmits user names and passwords in clear text. This is not recommended. |

*Table 8–1 Directives Descriptions (Cont.)*

| Directive Name | Description |
|---|---|
| `AuthUserFile` | Specifies the path to a file that contains user names and passwords. |
| `AuthGroupFile` | Specifies the path to a file that contains group names and their members. |

## Using mod_ossl to Authenticate Users

Secure Sockets Layer (SSL) is an encrypted communication protocol that is designed to securely send messages across the Internet. It resides between Oracle HTTP Server on the application layer and the TCP/IP layer, transparently handling encryption and decryption when a secure connection is made by a client.

One common use of SSL is to secure Web HTTP communication between a browser and a Web server. This case does not preclude the use of non-secured HTTP. The secure version is simply HTTP over SSL (named HTTPS). The differences are that HTTPS uses the URL scheme `https://` rather than `http://`, and its default communication port is 4443.

`mod_ossl` is a plug-in to Oracle HTTP Server that enables the server to use SSL. `mod_ossl` replaces `mod_ssl` in the Oracle HTTP Server distribution. Oracle no longer supports `mod_ssl`.

> **See Also:** "Using mod_ossl" on page 8-12 for detailed information regarding `mod_ossl`.

## Enabling SSL

By default, SSL is disabled when you install Oracle Database. If you want to enable SSL after installation, perform the following steps:

1. Open `opmn.xml` in a text editor.
2. In the `<ias-component id=HTTP_Server>` entry, change the start mode from "ssl-disabled" to "ssl-enabled". After modification is made, the entry should look like the following:

   ```
   <data id="start-mode" value="ssl-enabled"/>
   ```

3. Save and close `opmn.xml`.
4. Reload OPMN using the following command:

   ```
   opmnctl reload
   ```

5. Stop Oracle HTTP Server using the following command:
   - UNIX: *ORACLE_HOME*/opmn/bin> opmnctl [verbose] stopproc ias-component=HTTP_Server
   - Windows: *ORACLE_HOME*\opmn\bin> opmnctl [verbose] stopproc ias-component=HTTP_Server
6. Start Oracle HTTP Server using the following command:
   - UNIX: *ORACLE_HOME*/opmn/bin> opmnctl [verbose] startproc ias-component=HTTP_Server
   - Windows: *ORACLE_HOME*\opmn\bin> opmnctl [verbose] startproc ias-component=HTTP_Server
7. You can verify if SSL was enabled successfully by navigating to the SSL port, for example:

   ```
   HTTPS://hostname:4443
   ```

---

**Note:** The preceding steps enable SSL for Oracle HTTP Server using a default insecure certificate. To achieve completely secure SSL communication with Oracle HTTP Server, obtain and configure a real certificate within `mod_ossl`.

---

# Security Services Implemented Within Oracle HTTP Server

Oracle HTTP Server provides security services that enable you to protect your server from unwanted users and malicious attacks. These security services ensure secure data exchanged between client and the server.

`mod_ossl` enables secure connections between Oracle HTTP Server and a browser client by using an Oracle-provided encryption mechanism over SSL. It also provides data integrity and strong authentication for users and HTTP servers.

## Using mod_ossl

`mod_ossl` provides standard support for HTTPS protocol connections to Oracle Database. It enables secure connections between Oracle HTTP Server and a browser client by using an Oracle-provided encryption mechanism over SSL. It may also be used for authentication over the Internet through the use of digital certificate technology. It supports SSL v. 3.0, and provides:

- Encrypted communication between client and server, using **RSA** or **DES** encryption standards.
- Integrity checking of client-server communication using **MD5** or **SHA** checksum algorithms.
- Certificate management with Oracle **wallet**s.

Table 8–2 identifies the differences between `mod_ossl`, and `mod_ssl`.

***Table 8–2 Differences between mod_ossl and mod_ssl***

| Feature | mod_ossl | mod_ssl |
|---|---|---|
| SSL versions supported | 3.0 | 2.0, 3.0, TLS 1.0 |
| Certificate management | Oracle Wallet[1, 2] | Text file |

[1] Oracle Wallet Manager is a tool that manages certificates for mod_ossl.

[2] Supports obfuscated passwords.

The following mod_ssl directives listed are not supported by mod_ossl.

- SSLRandomSeed
- SSLCertificateFile
- SSLCertificateKeyFile
- SSLCertificateChainFile
- SSLCACertificateFile
- SSLCACertificatePath
- SSLVerifyDepth

**Caution:** The server will not start if these directives are used.

### Using mod_ossl Directives

To configure SSL for your Oracle HTTP Server, enter the mod_ossl directives you want to use in the httpd.conf file.

The following directive are described in subsequent sections:

- SSLAccelerator
- SSLCARevocationFile
- SSLCARevocationPath
- SSLCipherSuite
- SSLEngine
- SSLLog
- SSLLogLevel
- SSLMutex
- SSLOptions
- SSLPassPhraseDialog
- SSLProtocol
- SSLRequire
- SSLRequireSSL
- SSLSessionCache

- SSLSessionCacheTimeout
- SSLVerifyClient
- SSLWallet
- SSLWalletPassword

**SSLAccelerator** Specifies if SSL accelerator is used. Currently only nFast card is supported.

| Category | Value |
|---|---|
| Valid Values | `yes/no` |
| Syntax | `SSLAccelerator yes\|no` |
| Default | `SSLAccelerator no` |
| Context | server configuration |

**SSLCARevocationFile** Specifies the file where you can assemble the Certificate Revocation Lists (CRLs) from **CA**s (Certificate Authorities) that you accept certificates from. These are used for client authentication. Such a file is the concatenation of various **PEM**-encoded CRL files in order of preference. This directive can be used alternatively or additionally to SSLCARevocationPath.

| Category | Value |
|---|---|
| Syntax | `SSLCARevocationFile` *`file_name`* |
| Example | `SSLCARevocationFile /`*`ORACLE_HOME`*`/Apache/conf/ssl.crl/ca_bundle.crl` |
| Default | None |
| Context | server configuration, virtual host |

**SSLCARevocationPath** Specifies the directory where **PEM**-encoded Certificate Revocation Lists (CRLs) are stored. These CRLs come from the **CA**s (Certificate Authorities) that you accept certificates from. If a client attempts to authenticate itself with a certificate that is on one of these CRLs, then the certificate is revoked and the client cannot authenticate itself with your server.

| Category | Value |
|---|---|
| Syntax | `SSLCARevocationPath` *`path/to/CRL_directory/`* |
| Example | `SSLCARevocationPath` */`ORACLE_HOME`*`/Apache/conf/ssl.crl/` |
| Default | None |
| Context | server configuration, virtual host |

**SSLCipherSuite** Specifies the SSL **cipher suite** that the client can use during the SSL handshake. This directive uses a colon-separated cipher specification string to identify the cipher suite. Table 8–3 shows the tags you can use in the string to describe the cipher suite you want.

Tags are joined together with prefixes to form cipher specification string.

| Category | Value |
|---|---|
| Valid Values | `none`: Adds the cipher to the list<br>`+` : Adds the cipher to the list and place them in the correct location in the list<br>`-` : Remove the cipher from the list (can be added later)<br>`!` : Remove the cipher from the list permanently |
| Example | `SSLCipherSuite` *`ALL:!LOW:!DH`*<br>In this example, all ciphers are specified except low strength ciphers and those using the **Diffie-Hellman key negotiation algorithm**. |
| Syntax | `SSLCipherSuite` *`cipher-spec`* |
| Default | None |
| Context | server configuration, virtual host, directory |

***Table 8–3 SSLCipher Suite Tags***

| Function | Tag | Meaning |
|---|---|---|
| Key exchange | kRSA | RSA key exchange |
| Key exchange | kDHr | Diffie-Hellman key exchange with RSA key |
| Authentication | aNULL | No authentication |
| Authentication | aRSA | RSA authentication |
| Authentication | aDH | Diffie-Hellman authentication |
| Encryption | eNULL | No encryption |
| Encryption | DES | DES encoding |
| Encryption | 3DES | Triple DES encoding |
| Encryption | RC4 | RC4 encoding |
| Data Integrity | MD5 | MD5 hash function |
| Data Integrity | SHA | SHA hash function |
| Aliases | SSLv3 | All SSL version 3.0 ciphers |
| Aliases | EXP | All export ciphers |
| Aliases | EXP40 | All 40-bit export ciphers only |
| Aliases | EXP56 | All 56-bit export ciphers only |
| Aliases | LOW | All low strength ciphers (export and single DES) |
| Aliases | MEDIUM | All ciphers with 128-bit encryption |
| Aliases | HIGH | All ciphers using triple DES |
| Aliases | RSA | All ciphers using RSA key exchange |
| Aliases | DH | All ciphers using Diffie-Hellman key exchange |

**Note:** There are restrictions if export versions of browsers are used. Oracle module, `mod_ossl`, supports RC4-40 encryption only when the server uses 512 bit key size wallets.

***Table 8–4 Cipher Suites Supported in Oracle Advanced Security 9i***

| Cipher Suite | Authentication | Encryption | Data Integrity |
|---|---|---|---|
| SSL_RSA_WITH_3DES_EDE_CBC_SHA | RSA | 3DES EDE CBC | SHA |
| SSL_RSA_WITH_RC4_128_SHA | RSA | RC4 128 | SHA |
| SSL_RSA_WITH_RC4_128_MD5 | RSA | RC4 128 | MD5 |
| SSL_RSA_WITH_DES_CBC_SHA | RSA | DES CBC | SHA |
| SSL_DH_anon_WITH_3DES_EDE_CBC_SHA | DH anon | 3DES EDE CBC | SHA |
| SSL_DH_anon_WITH_RC4_128_MD5 | DH anon | RC4 128 | MD5 |
| SSL_RSA_WITH_3DES_EDE_CBC_SHA | RSA | 3DES EDE CBC | SHA |
| SSL_DH_anon_WITH_DES_CBC_SHA | DH anon | DES CBC | SHA |
| SSL_RSA_EXPORT_WITH_RC4_40_MD5 | RSA | RC4 40 | MD5 |
| SSL_RSA_EXPORT_WITH_DES40_CBC_SHA | RSA | DES40 CBC | SHA |
| SSL_DH_anon_EXPORT_WITH_RC4_40_MD5 | DH anon | RC4 40 | MD5 |
| SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA | DH anon | DES40 CBC | SHA |

**SSLEngine** Toggles the usage of the SSL Protocol Engine. This is usually used inside a `<VirtualHost>` section to enable SSL for a particular virtual host. By default, the SSL Protocol Engine is disabled for both the main server and all configured virtual hosts.

***Example 8–8 Using SSL Engine Directive***

```
<VirtualHost_dafault_:4443>
  SSLEngine on
  ...
</VirtualHost>
```

| Category | Value |
|---|---|
| Syntax | `SSLEngine on\|off` |
| Default | `SSLEngine off` |
| Context | server configuration, virtual host |

**SSLLog** Specifies where the SSL engine log file will be written. (Error messages will also be duplicated to the standard Oracle HTTP Server log file specified by the ErrorLog directive.)

Place this file at a location where only root can write, so that it cannot be used for symlink attacks. If the filename does not begin with a slash (/), it is assumed to be relative to the `ServerRoot`. If the filename begins with a bar (|), then the string following the bar is expected to be a path to an executable program to which a reliable pipe can be established.

This directive should occur only once for each virtual server configuration.

| Category | Value |
|---|---|
| Syntax | `SSLVerifyClient` *`path/to/filename`* |
| Default | None |
| Context | server configuration, virtual host |

**SSLLogLevel** Specifies the verbosity degree of the SSL engine log file.

| Category | Value |
|---|---|
| Valid Values | The levels are (in ascending order, where each level is included in the levels preceding it):<br>■ `none`: No dedicated SSL logging is done. Messages of type 'error' are duplicated to the standard HTTP server log file specified by the `ErrorLog` directive.<br>■ `error`: Only messages of the type 'error' (conditions that stop processing) are logged.<br>■ `warn`: Messages that notify of non-fatal problems (conditions that do not stop processing) are logged.<br>■ `info`: Messages that summarize major processing actions are logged.<br>■ `trace`: Messages that summarize minor processing actions are logged.<br>■ `debug`: Messages that summarize development and low-level I/O operations are logged. |
| Syntax | `SSLLogLevel` *`level`* |
| Default | None |
| Context | server configuration, virtual host |

**SSLMutex** Type of semaphore (lock) for SSL engine's mutual exclusion of operations that have to be synchronized between Oracle HTTP Server processes.

| Category | Value |
|---|---|
| Valid Values | ■ `none`: Uses no mutex at all. Not recommended, because the mutex synchronizes the write access to the SSL session cache. If you do not configure a mutex, the session cache can become garbled.<br>■ `file:`*`path/to/mutex`*: Uses a file for locking. The process ID (PID) of the Oracle HTTP Server parent process is appended to the filename to ensure uniqueness. If the filename does not begin with a slash (/), it is assumed to be relative to `ServerRoot`. This setting is not available on Windows.<br>■ `sem`: Uses an operating system semaphore to synchronize writes. On UNIX, it would be a Sys V IPC semaphore; on Windows, it is a Windows Mutex. This is the best choice, if the operating system supports it. |
| Example | `SSLMutex file:`*`/usr/local/apache/logs/ssl_mutex`* |
| Syntax | `SSLMutex` *`type`* |
| Default | `SSLMutex` *`none`* |
| Context | server configuration |

**SSLOptions** Controls various runtime options on a per-directory basis. In general, if multiple options apply to a directory, the most comprehensive option is applied (options are not merged). However, if all of the options in an `SSLOptions` directive are preceded by a plus ('+') or minus ('-') symbol, then the options are merged. Options preceded by a plus are added to the options currently in force, and options preceded by a minus are removed from the options currently in force.

| Category | Value |
|---|---|
| Valid Values | ▪ `StdEnvVars`: Creates the standard set of CGI/SSI environment variables that are related to SSL. This is disabled by default because the extraction operation uses a lot of CPU time and usually has no application when serving static content. Typically, you only enable this for CGI/SSI requests.<br>▪ `ExportCertData`: Enables the following additional CGI/SSI variables:<br>`SSL_SERVER_CERT`<br>`SSL_CLIENT_CERT`<br>`SSL_CLIENT_CERT_CHAIN_n` (where n= 0, 1, 2...)<br>These variables contain the Privacy Enhanced Mail (**PEM**)-encoded X.509 certificates for the server and the client for the current HTTPS connection, and can be used by CGI scripts for deeper certificate checking. All other certificates of the client certificate chain are provided. This option is "Off" by default because there is a performance cost associated with using it.<br>`SSL_CLIENT_CERT_CHAIN_n` variables are in the following order: `SSL_CLIENT_CERT_CHAIN_0` is the intermediate CA who signs `SSL_CLIENT_CERT`. `SSL_CLIENT_CERT_CHAIN_1` is the intermediate CA who signs `SSL_CLIENT_CERT_CHAIN_0`, and so forth, with `SSL_CLIENT_ROOT_CERT` as the root CA.<br>▪ `FakeBasicAuth`: Translates the subject **distinguished name** of the client **X.509** certificate into an HTTP basic authorization user name. This means that the standard HTTP server authentication methods can be used for access control. Note that no password is obtained from the user; the string 'password' is substituted. |

| Category | Value |
|---|---|
| Valid Values (for `SSLOptions` continued) | ■ `StrictRequire`: Denies access when, according to `SSLRequireSSL` or `SSLRequire` directives, access should be forbidden. Without `StrictRequire`, it is possible for a `'Satisfy any'` directive setting to override the `SSLRequire` or `SSLRequireSSL` directive, allowing access if the client passes the host restriction or supplies a valid user name and password.<br>Thus, the combination of `SSLRequireSSL` or `SSLRequire` with `SSLOptions +StrictRequire` gives `mod_ossl` the ability to override a `'Satisfy any'` directive in all cases.<br>■ `CompatEnvVars`: Exports obsolete environment variables for backward compatibility to Apache SSL 1.x, `mod_ssl` 2.0.x, Sioux 1.0, and Stronghold 2.x. Use this to provide compatibility to existing CGI scripts.<br>■ `OptRenegotiate`: This enables optimized SSL connection renegotiation handling when SSL directives are used in a per-directory context. |
| Syntax | `SSLOptions` *`[+-] option`* |
| Default | None |
| Context | server configuration, virtual host, directory |

**SSLPassPhraseDialog** Type of pass phrase dialog for wallet access. `mod_ossl` asks the administrator for a pass phrase in order to access the wallet.

| Category | Value |
|---|---|
| Valid Values | ■ `builtin`: when the server is started, `mod_ossl` prompts for a password for each wallet.<br>This cannot be used when Oracle HTTP Server is managed by OPMN. No user interaction is allowed when Oracle HTTP Server is started by OPMN.<br>■ `exec:`*`path/to/program`* - when the server is started, `mod_ossl` calls an external program configured for each wallet. This program is invoked with two arguments: *`servername:portnumber`* and `RSA` or `DSA`. |
| Syntax | `SSLPassPhraseDialog` *type* |
| Example | `SSLPassPhraseDialog exec:`*`/usr/local/apache/sbin/pfilter`* |
| Default | `SSLPassPhraseDialog` *`builtin`* |
| Context | server configuration |

**SSLProtocol** Specifies SSL protocol(s) for `mod_ossl` to use when establishing the server environment. Clients can only connect with one of the specified protocols.

| Category | Value |
|---|---|
| Valid Values | `SSLv3`<br>SSL version 3.0 |
| Example | To specify only SSL version 3.0, set this directive to the following:<br>`SSLProtocol` *`+SSLv3`* |
| Syntax | `SSLProtocol` *`[+-] protocol`* |
| Default | `SSLProtocol +SSLv3` |
| Context | server configuration, virtual host |

**SSLRequire** Denies access unless an arbitrarily complex boolean expression is true. The expression must match the following syntax (given as a BNF grammar notation):

| Category | Value |
|---|---|
| | `expr ::= "true" \| "false"`<br>`"!" expr`<br>`expr "&&" expr`<br>`expr "\|\|" expr`<br>`"(" expr ")"` |
| | `comp ::=word "==" word \| word "eq" word`<br>`word "!=" word \|word "ne" word`<br>`word "<" word \|word "lt" word`<br>`word "<=" word \|word "le" word`<br>`word ">" word \|word "gt" word`<br>`word ">=" word \|word "ge" word`<br>`word "=~" regex`<br>`word "!~" regex`<br>`wordlist ::= word`<br>`wordlist "," word` |
| | `word ::= digit`<br>`cstring`<br>`variable`<br>`function` |
| | `digit ::= [0-9]+` |
| | `cstring ::= "..."` |
| | `variable ::= "%{varname}"`<br><br>Table 8–5 and Table 8–6 list standard and SSL variables. These are valid values for varname. |
| | `function ::= funcname "(" funcargs ")"`<br><br>For funcname, the following function is available:<br>`file(`*`filename`*`)`<br>The file function takes one string argument, the filename, and expands to the contents of the file. This is useful for evaluating the file's contents against a regular expression. |
| Syntax | `SSLRequire` *`expression`* |
| Default | None |
| Context | directory |

Table 8–5 lists the standard variables for `SSLRequire` varname.

***Table 8–5 Standard Variables for SSLRequire Varname***

| Standard Variables | Standard Variables | Standard Variables |
|---|---|---|
| HTTP_USER_AGENT | PATH_INFO | AUTH_TYPE |
| HTTP_REFERER | QUERY_STRING | SERVER_SOFTWARE |
| HTTP_COOKIE | REMOTE_HOST | API_VERSION |
| HTTP_FORWARDED | REMOTE_IDENT | TIME_YEAR |
| HTTP_HOST | IS_SUBREQ | TIME_MON |
| HTTP_PROXY_CONNECTION | DOCUMENT_ROOT | TIME_DAY |
| HTTP_ACCEPT | SERVER_ADMIN | TIME_HOUR |
| HTTP:headername | SERVER_NAME | TIME_MIN |
| THE_REQUEST | SERVER_PORT | TIME_SEC |
| REQUEST_METHOD | SERVER_PROTOCOL | TIME_WDAY |
| REQUEST_SCHEME | REMOTE_ADDR | TIME |
| REQUEST_URI | REMOTE_USER | ENV:variablename |
| REQUEST_FILENAME | | |

Table 8–6 lists the SSL variables for `SSLRequire` varname.

***Table 8–6 SSL Variables for SSLRequire Varname***

| SSL Variables | SSL Variables | SSL Variables |
|---|---|---|
| HTTPS | SSL_PROTOCOL | SSL_CIPHER_ALGKEYSIZE |
| SSL_CIPHER | SSL_CIPHER_EXPORT | SSL_VERSION_INTERFACE |
| SSL_CIPHER_USEKEYSIZE | SSL_VERSION_LIBRARY | SSL_SESSION_ID |
| SSL_CLIENT_V_END | SSL_CLIENT_M_SERIAL | SSL_CLIENT_V_START |
| SSL_CLIENT_S_DN_ST | SSL_CLIENT_S_DN | SSL_CLIENT_S_DN_C |
| SSL_CLIENT_S_DN_CN | SSL_CLIENT_S_DN_O | SSL_CLIENT_S_DN_OU |
| SSL_CLIENT_S_DN_G | SSL_CLIENT_S_DN_T | SSL_CLIENT_S_DN_I |
| SSL_CLIENT_S_DN_UID | SSL_CLIENT_S_DN_S | SSL_CLIENT_S_DN_D |
| SSL_CLIENT_I_DN_C | SSL_CLIENT_S_DN_Email | SSL_CLIENT_I_DN |

*Table 8–6 SSL Variables for SSLRequire Varname (Cont.)*

| SSL Variables | SSL Variables | SSL Variables |
|---|---|---|
| SSL_CLIENT_I_DN_O | SSL_CLIENT_I_DN_ST | SSL_CLIENT_I_DN_L |
| SSL_CLIENT_I_DN_T | SSL_CLIENT_I_DN_OU | SSL_CLIENT_I_DN_CN |
| SSL_CLIENT_I_DN_S | SSL_CLIENT_I_DN_I | SSL_CLIENT_I_DN_G |
| SSL_CLIENT_I_DN_Email | SSL_CLIENT_I_DN_D | SSL_CLIENT_I_DN_UID |
| SSL_CLIENT_CERT | SSL_CLIENT_CERT_ CHAIN_n | SSL_CLIENT_ROOT_CERT |
| SSL_CLIENT_VERIFY | SSL_CLIENT_M_VERSION | SSL_SERVER_M_VERSION |
| SSL_SERVER_V_START | SSL_SERVER_V_END | SSL_SERVER_M_SERIAL |
| SSL_SERVER_S_DN_C | SSL_SERVERT_S_DN_ST | SSL_SERVER_S_DN |
| SSL_SERVER_S_DN_OU | SSL_SERVER_S_DN_CN | SSL_SERVER_S_DN_O |
| SSL_SERVER_S_DN_I | SSL_SERVER_S_DN_G | SSL_SERVER_S_DN_T |
| SSL_SERVER_S_DN_D | SSL_SERVER_S_DN_UID | SSL_SERVER_S_DN_S |
| SSL_SERVER_I_DN | SSL_SERVER_I_DN_C | SSL_SERVER_S_DN_Email |
| SSL_SERVER_I_DN_L | SSL_SERVER_I_DN_O | SSL_SERVER_I_DN_ST |
| SSL_SERVER_I_DN_CN | SSSL_SERVER_I_DN_T | SSL_SERVER_I_DN_OU |
| SSL_SERVER_I_DN_G | SSL_SERVER_I_DN_I | |

**SSLRequireSSL** Denies access to clients not using SSL. This is a useful directive for absolute protection of a SSL-enabled virtual host or directories in which configuration errors could create security vulnerabilities.

| Category | Value |
|---|---|
| Syntax | `SSLRequireSSL` |
| Default | None |
| Context | directory |

**SSLSessionCache** Specifies the global/interprocess session cache storage type. The cache provides an optional way to speed up parallel request processing.

| Category | Value |
|---|---|
| Valid Values | ■ `none`: disables the global/interprocess session cache. Produces no impact on functionality, but makes a major difference in performance.<br>■ `shmht:`*`/path/to/datafile[bytes]`*: Uses a high-performance hash table (`bytes` specifies approximate size) inside a shared memory segment in RAM, which is established by the `/path/to/datafile`. This hash table synchronizes the local SSL memory caches of the server processes.<br>■ `shmcb:`*`/path/to/datafile[bytes]`*: Uses a high-performance Shared Memory Cyclic Buffer (SHMCB) session cache to synchronize the local SSL memory caches of the server processes. The performance of `shmcb` is more uniform in all environments when compared to `shmht`. |
| Syntax | `SSLSessionCache` *`type`* |
| Examples | `SSLSessionCache shmht: /`*`ORACLE_HOME`*`/Apache/Apache/logs/ssl_scache(512000)`<br>`SSLSessionCache shmcb: /`*`ORACLE_HOME`*`/Apache/Apache/logs/ssl_scache(512000)` |
| Default | `SSLSessionCache` *`none`* |

**SSLSessionCacheTimeout** Specifies the number of seconds before a SSL session in the session cache expires.

| Category | Value |
|---|---|
| Syntax | `SSLSessionCacheTimeout` *`seconds`* |
| Default | 300 |
| Context | server configuration |

**SSLVerifyClient** Specifies whether or not a client must present a certificate when connecting.

| Category | Value |
|---|---|
| Valid Values | ▪ `none`: No client certificate is required<br>▪ `optional`: Client may present a valid certificate<br>▪ `require`: Client must present a valid certificate |
| Syntax | `SSLVerifyClient` *`level`* |
| Default | None |
| Context | server configuration, virtual host |

**Note:** The level `optional_no_ca` included with `mod_ssl` (in which the client can present a valid certificate, but it need not be verifiable) is not supported in `mod_ossl`.

**SSLWallet** Specifies the location of the wallet with its **WRL**.

| Category | Value |
|---|---|
| Syntax | `SSLWallet` *`wrl`*<br>The format of `wrl` is: *`file:path to wallet`* |
| Example | `SSLWallet` *`file:/etc/`*`ORACLE/WALLETS/`*`server`*<br>Other values of wrl may be used as permitted by the Oracle SSL product. |
| Default | None |
| Context | server configuration, virtual host |

**SSLWalletPassword** Specifies the Wallet password needed to access the wallet specified within the same context. You can choose either a cleartext wallet password or an obfuscated password. The obfuscated password is created with the command line tool `iasobf`. If you must use a regular wallet, Oracle recommends that you use the obfuscated password instead of a cleartext password.

**See Also:** "Using the iasobf Utility" on page 8-33

| Category | Value |
|---|---|
| Syntax | `SSLWalletPassword` *`password`*<br>If no password is required do not set this directive.<br>Note: If a wallet created with the Auto Login feature of Oracle Wallet Manager is used, then do not set this directive because these wallets do not require passwords. |
| Default | None |
| Context | server configuration, virtual host |

**Note:** `SSLWalletPassword` has been deprecated. A warning message is generated in the Oracle HTTP Server log if this directive is used.

For secure wallets, Oracle recommends that you get a SSO wallet instead. Refer to the *Oracle Application Server 10g Security Guide* for information on SSO wallet.

### Using mod_proxy Directives

The following directives are for `mod_proxy` support only:

- SSLProxyCache
- SSLProxyCipherSuite
- SSLProxyProtocol
- SSLProxyWallet
- SSLProxyWalletPassword

**SSLProxyCache** Specifies whether the proxy cache will be used. The proxy will use the same session as the SSL server uses.

| Category | Value |
|---|---|
| Syntax | `SSLProxyCache` *`on/off`* |
| Default | `SSLProxyCache off` |
| Context | server configuration, virtual host |

**SSLProxyCipherSuite** Specifies the proxy server's cipher suite.

| Category | Value |
|---|---|
| Syntax | `SSLCipherSuite` *`cipher-spec`* |
| Default | `None` |
| Context | server configuration, virtual host |

**SSLProxyProtocol** Controls the proxy server's SSL protocol flavors.

| Category | Value |
|---|---|
| Syntax | `SSLProxyProtocol [+-]` *`protocol`* |
| Default | `None` |
| Context | server configuration, virtual host |

**SSLProxyWallet** Specifies the location of the wallet containing the certificates to use when opening proxy connections.

| Category | Value |
|---|---|
| Syntax | `SSLProxyWallet` *`wrl`* |
| Default | `None` |
| Context | server configuration, virtual host |

**SSLProxyWalletPassword** Specifies the proxy wallet password.

| Category | Value |
|---|---|
| Syntax | `SSLProxyWalletPassword` *`password`* |
| Default | `None` |
| Context | server configuration, virtual host |

**Note:** `SSLProxyWalletPassword` has been deprecated. A warning message is generated in the Oracle HTTP Server log if this directive is used.

For secure wallets, Oracle recommends that you get a SSO wallet instead.

Refer to the *Oracle Application Server 10g Security Guide* for information on SSO wallet.

### Using mod_ossl Directives to Configure Client Authentication

This section provides instructions on how you can use the directives mentioned earlier to set up configurations that enable you to use client certificates for authenticating clients. Following are some scenarios:

- **Authenticating clients based on certificates when all clients are known.**

  The server wallet has imported the CA certificate which signed all the client certificates.

  For example, specify the following directives in the `httpd.conf` file:

  ```
  SSLVerifyClient require
  ```

- **Authenticating for a particular URL based on certificates, while allowing arbitrary clients to access the rest of the server**

  To enable this, use the per-directory reconfiguration feature of `mod_ossl`. Session re-negotiation enables an SSL session to be re-negotiated with a client after the initial request and URL have been read. This is only supported for requests that do not contain body data, such as `GET` requests.

  **See Also:**

  - "Classes of Directives" on page 2-3 for more information.
  - `mod_ssl` documentation.

  For example, specify the following directives in the `httpd.conf` file:

  ```
  <Location /secure/area>
    SSLVerifyClient require
  </Location>
  ```

### Using the iasobf Utility

The `iasobf` utility enables you to generate an obfuscated wallet password from a **cleartext** password.

If you are using an Oracle Wallet that has been created with Auto Login enabled (an SSO wallet), then you do not need to use this utility. However, if you must use a regular wallet with a password, then Oracle recommends that you use the password obfuscation tool `iasobf`, which is located in *`ORACLE_HOME`*`/Apache/Apache/bin`, to generate an obfuscated wallet password from a cleartext password.

To generate an obfuscated wallet password, the command syntax is:

```
iasobf -p password
```

The obfuscated password is printed to the terminal. The arguments are optional. If you do not type them, the tool will prompt you for the password.

---

**On Windows systems:** The corresponding tool for Windows environments is called `osslpassword`, which can be used in the same way as `iasobf`.

---

# 9

# Frequently Asked Questions

This chapter provides answers to frequently asked questions about Oracle HTTP Server.

> **See Also:** "Frequently Asked Questions" in the Apache Server documentation.

Documentation from the Apache Software Foundation is referenced when applicable.

> **Note:** Readers using this guide in PDF or hard copy formats will be unable to access third-party documentation, which Oracle provides in HTML format only. To access the third-party documentation referenced in this guide, use the HTML version of this guide and click the hyperlinks.

## Creating Application-specific Error Pages

Oracle HTTP Server has a default content handler for dealing with errors. You can use the `ErrorDocument` directive to override the defaults.

> **See Also:** "ErrorDocument directive" in the Apache Server documentation.

## Offering HTTPS to ISP (Virtual Host) Customers

For HTTP, Oracle HTTP Server supports two types of virtual hosts: name-based and IP-based. HTTPS supports only IP-based virtual hosts.

If you are using IP-based virtual hosts for HTTP, then the customer has a virtual server listening on port 80 of a per-customer IP address. To provide HTTPS for these customers, simply add an additional virtual host for each user listening on port 4443 of that same per-customer IP address and use SSL directives, such as SSLRequireSSL to specify the per-customer SSL characteristics. Note that each customer can have their own wallet and server certificate.

If you are using name-based virtual hosts for HTTP, each customer has a virtual server listening on port 80 of a shared IP address. To provide HTTPS for those customers, you can add a single shared IP virtual host listening on port 4443 of the shared IP address. All customers will share the SSL configuration, including the wallet and ISP's server certificate.

> **See Also:** "Running Oracle HTTP Server as Root" on page 4-2 for instructions on running Oracle HTTP Server with ports lesser than 1024.

## Using Oracle HTTP Server as Cache

You can use the Oracle HTTP Server as a cache by setting the `ProxyRequests` to "On" and `CacheRoot` directives.

> **See Also:** "`ProxyRequests` and `CacheRoot` directives" in the Apache Server documentation.

## Using Different Language and Character Set Versions of Document

You can use *multiviews*, a general name given to the Apache server's ability to provide language and character-specific document variants in response to a request.

**See Also:** "Multiviews" in the Apache Server documentation.

## Sending Proxy Sensitive Requests to Oracle HTTP Server Behind a Firewall

You should use the Proxy directives, and not the Cache directives, to send proxy sensitive requests across firewalls.

## Oracle HTTP Server Version Number

Oracle HTTP Server is based on Apache version 1.3.28.

## Apache v2.0 Support with Oracle Database, 10*g* Release 1 (10.1)

Oracle Database, 10*g* Release 1 (10.1) is still based on the 1.3.x stack from Apache organization.

## Applying Apache Security patches to Oracle HTTP Server

You cannot apply the Apache security patches to Oracle HTTP Server for the following reasons:

- Oracle tests and appropriately modifies security patches before releasing them to Oracle HTTP Server users.
- In many cases those alerts may not be applicable, for example, `openSSL` alerts, since Oracle has removed those components from the stack in use.
- Oracle releases these patches soon enough that the time-delay impact of getting the patch from Oracle versus open source organization should be minimal and the benefit with respect to supportability, tremendous.

## Supporting PHP

mod_php is not supported, however, you have the following two options:

- Install mod_php by yourself and use it. If there is a support question on any aspect of Oracle HTTP Server, you might be asked to reproduce the problem without mod_php.
- Use PHP in a CGI mode, in which case support of the rest of the Oracle HTTP Server stack would not be an issue.

## Creating Application Name Space that Works Across Firewalls and Clusters

The general idea is that all servers in a distributed Web site should agree on a single URL namespace. Every server serves some part of that namespace, and is able to redirect or proxy requests for URLs that it does not serve to a server that is "closer" to that URL. For example, your namespaces could be the following:

```
/app1/login.html
/app1/catalog.html
/app1/dologin.jsp
/app2/orderForm.html
/apps/placeOrder.jsp
```

We could initially map this namespace to two Web servers by putting app1 on server1 and app2 on server2. Server1's configuration might look like the following:

```
Redirect permanent /app2 http://server2/app2
Alias /app1 /myApps/application1
<Directory /myApps/application1>
  ...
</Directory>
```

Server2's configuration is complementary. If you decide to partition the namespace by content type (HTML on server, JSP on server2), change server configuration and move files around, but do not have to make changes to the application itself. The resulting configuration of server1 might look like the following:

```
RedirectMatch permanent (.*) \.jsp$ http://server2/$1.jsp
AliasMatch ^/app(.*) \.html$ /myPages/application$1.html
<DirectoryMatch "^/myPages/application\d">
  ...
</DirectoryMatch>
```

Note that the amount of actual redirection can be minimized by configuring a hardware load balancer to send requests to server1 or server2 based on the URL.

## Protecting Web Site From Hackers

There are many attacks, and new attacks are invented everyday. Following are some general guidelines for securing your site. You can never be completely secure, but you can avoid being an easy target.

- Use a commercial firewall between your ISP and your Web server. Recognize, however, that not all hackers are outside your organization.
- Use switched ethernet to limit the amount of traffic a compromised server can sniff. Use additional firewalls between Web server machines and highly sensitive internal servers running database and enterprise applications.
- Remove unnecessary network services such as RPC, Finger, telnet from your server machine.
- Carefully validate all input from Web forms. Be especially wary of long input strings and input that contains non-printable characters, HTML tags, or javascript tags.
- Encrypt or randomize the contents of cookies that contain sensitive information. For example, it should be difficult to guess a valid sessionID to prevent a hacker from hijacking a valid session.
- Check often for security patches for all your system and application software, and install them as soon as possible. Be sure these patches come from bona fide sources; download from trusted sites and verify the cryptographic checksum.
- Use an intrusion detection package to monitor for defaced Web pages, viruses, and presence of "rootkits" that indicate hackers have broken in. If possible, mount system executables and Web content on read-only file systems.
- Have a "forensic analysis" package on hand to capture evidence of a break in as soon as detected. This aids in prosecution of the hackers.

# A

# Oracle HTTP Server Configuration Files

This appendix lists commonly used Oracle HTTP Server configuration files.

Files discussed are:

- httpd.conf
- opmn.xml

Documentation from the Apache Software Foundation is referenced when applicable.

> **Note:** Readers using this guide in PDF or hard copy formats will be unable to access third-party documentation, which Oracle provides in HTML format only. To access the third-party documentation referenced in this guide, use the HTML version of this guide and click the hyperlinks.

# httpd.conf

This is a server configuration file which typically contains directives that affect how the server runs, such as user and group IDs it should use, and location of other files. Because the server configuration file is the main file that the server starts with, Oracle HTTP Server does not include any directive that says where to locate it. The location is passed on command line when the server starts.

It is located at:

- UNIX: *ORACLE_HOME*/Apache/Apache/conf/httpd.conf
- Windows: *ORACLE_HOME*\Apache\Apache\conf\httpd.conf

You should use only this file, and not `srm.conf` or `access.conf` because it is must easier to manage a single configuration file.

## httpd.conf File Structure

`httpd.conf` is arranged in the following sections:

- Global Environment
- Main Server Configuration
- Virtual Hosts

### Global Environment

This is section one of the `httpd.conf` file. It contains configuration directives dealing with Oracle HTTP Server.

**See Also:**

- "Specifying File Locations" on page 3-4
- "Limiting the Number of Processes and Connections" on page 4-5
- Chapter , "Specifying Listener Ports and Addresses" on page 5-2

### Main Server Configuration

This is section two of the `httpd.conf` file. It contains the directives of the default server.

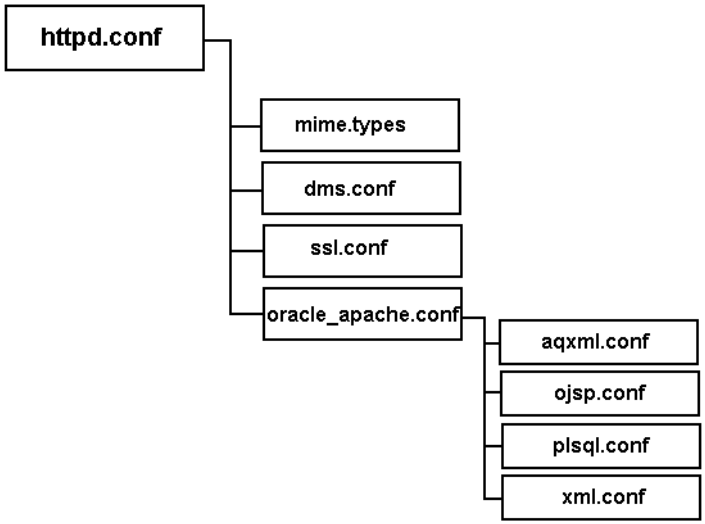
> **See Also:** "Setting Server and Administrator Functions" on page 3-2.

### Virtual Hosts

This is section three of the `httpd.conf` file.It contains parameters specific to virtual hosts, which override some of the main server configuration defaults.

Figure A–1 illustrates the file structure of the `httpd.conf` file.

***Figure A–1 httpd.conf File***



As shown in Figure A–1, `httpd.conf` contains directives to include configuration files such as:

- mime.types
- dms.conf
- oracle_apache.conf
- ssl.conf

## mime.types

`mime.types` controls the Multi Internet media types that are sent to the client for the given file extensions. Sending the correct media type to the client is important so that the client knows how to handle the content of the file. You can add extra types in the mime type file or add an `AddType` directive in the configuration file.

It is located at:

- UNIX: *`ORACLE_HOME`*`/Apache/Apache/conf`
- Windows: *`ORACLE_HOME`*`\Apache\Apache\conf`

**See Also:** "mod_mime" on page 7-12

## dms.conf

`dms.conf` enables you to monitor performance of site components with Oracle's Dynamic Monitoring Service (DMS).

It is located at:

- UNIX: *`ORACLE_HOME`*`/Apache/Apache/conf`
- Windows: *`ORACLE_HOME`*`\Apache\Apache\conf`

**See Also:** *Oracle Application Server 10g Performance Guide*

## oracle_apache.conf

`oracle_apache.conf` is included in the main configuration file to store configuration files of supported modules. It contains directives to include the following configuration files:

- aqxml.conf
- ojsp.conf
- plsql.conf
- xml.conf

### aqxml.conf

`aqxml.conf` enables and configures Advanced Queuing.

It is located at:

- UNIX: *ORACLE_HOME*`/Apache/Apache/conf`
- Windows: *ORACLE_HOME*`\Apache\Apache\conf`

### ojsp.conf

`ojsp.conf` configures Java Server Pages.

It is located at:

- UNIX: *ORACLE_HOME*`/Apache/jsp/conf`
- Windows: *ORACLE_HOME*`\Apache\jsp\conf`

### plsql.conf

`plsql.conf` configures and loads the PL/SQL module.

It is located at:

- UNIX: *ORACLE_HOME*`/Apache/modplsql/conf`
- Windows: *ORACLE_HOME*`\Apache\modplsql\conf`

**See Also:** "mod_plsql" on page 7-19

### xml.conf

`xml.conf` is associated the `.xsql` extension with the XSQL servlet.

It is located at:

- UNIX: *ORACLE_HOME*`/xdk/admin`
- Windows: *ORACLE_HOME*`\xdk\admin`

***Example A–1 oracle_apache.conf file***

```
# Advanced Queuing - AQ XML
include "/private1/oracle/Apache/Apache/conf/aqxml.conf"
#
#Directives needed for OraDAV module
include "/private1/oracle/Apache/oradav/conf/moddav.conf"
include "/private1/oracle/Apache/jsp/conf/ojsp.conf"
include "/private1/oracle/Apache/modplsql/conf/plsql.conf"
#
include "/private1/oracle/xdk/admin/xml.conf"
#
```

## ssl.conf

`ssl.conf` includes the SSL definitions and virtual host container. Out of the box, it is disabled by default.

It is located at:

- UNIX: *ORACLE_HOME*`/Apache/Apache/conf`
- Windows: *ORACLE_HOME*`\Apache\Apache\conf`

# opmn.xml

`opmn.xml` describes the processes that Oracle Process Manager and Notification Server (OPMN) manages within an Oracle Database installation.

The `opmn.xml` file is the main configuration file for OPMN. It contains information for the ONS, the PM, and Oracle Database component-specific configuration.The `opmn.xml` file shows you which Oracle Database components OPMN is managing on your system. It contains Oracle Database component entries arranged in the following hierarchical structure:

```
<ias-component>
  <process-type>
    <process-set>
```

- **<ias-component>**: This entry represents the Oracle Database component. It enables management of the component for processes such as starting and stopping.
- **<process-type>**: This subcomponent of the `<ias-component>` entry declares the type of process to run by association with a specific PM module.
- **<process-set>**: This sub-subcomponent of the `<ias-component>` entry enables you to declare different sets of optional runtime arguments and environments for the Oracle Database component.

`opmn.xml` is located at:

- UNIX: *`ORACLE_HOME`*`/opmn/conf`
- Windows: *`ORACLE_HOME`*`\opmn\conf`

> **See Also:** *Oracle Process Manager and Notification Server Administrator's Guide*

# B

# Third Party Licenses

This appendix includes the Third Party License for all the third party products included with Oracle Database.

Topics discussed are:

# Apache HTTP Server

Under the terms of the Apache license, Oracle is required to provide the following notices. However, the Oracle program license that accompanied this product determines your right to use the Oracle program, including the Apache software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the Apache software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or Apache.

## The Apache Software License

```
/* ====================================================================
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2000-2002 The Apache Software Foundation.  All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 *    if any, must include the following acknowledgment:
 *       "This product includes software developed by the
 *        Apache Software Foundation (http://www.apache.org/)."
 *    Alternately, this acknowledgment may appear in the software itself,
 *    if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "Apache" and "Apache Software Foundation" must
 *    not be used to endorse or promote products derived from this
 *    software without prior written permission. For written
 *    permission, please contact apache@apache.org.
 *
 * 5. Products derived from this software may not be called "Apache",
 *    nor may "Apache" appear in their name, without prior written
```

# Apache SOAP

Under the terms of the Apache license, Oracle is required to provide the following notices. However, the Oracle program license that accompanied this product determines your right to use the Oracle program, including the Apache software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the Apache software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or Apache.

## Apache SOAP License

Apache SOAP license 2.3.1

```
------------------------------------------------------------------------
/*
 * The Apache Software License, Version 1.1
 *
 *
 * Copyright (c) 1999 The Apache Software Foundation.  All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 *    if any, must include the following acknowledgment:
 *       "This product includes software developed by the
 *        Apache Software Foundation (http://www.apache.org/)."
 *    Alternately, this acknowledgment may appear in the software itself,
 *    if and wherever such third-party acknowledgments normally appear.
 *
 * 4. The names "SOAP" and "Apache Software Foundation" must
 *    not be used to endorse or promote products derived from this
 *    software without prior written permission. For written
```

```
*    permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache",
*    nor may "Apache" appear in their name, without prior written
*    permission of the Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* ====================================================================
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*/
```

# DBI Module

Oracle is required to provide the text of the third-party license, but the third-party program will be subject to the Oracle license, and Oracle will NOT provide warranties and technical support for the third-party technology.

This program contains third-party code from DBI. Under the terms of the DBI license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the DBI software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the DBI software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or DBI.

The DBI module is Copyright (c) 1994-2002 Tim Bunce. Ireland. All rights reserved.

You may distribute under the terms of either the GNU General Public License or the Artistic License, as specified in the Perl README file.

## Perl Artistic License

The "Artistic License"

### Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

### Definitions

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
   a. place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
   b. use the modified Package only within your corporation or organization.
   c. rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
   d. make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

   a. distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

   b. accompany the distribution with the machine-readable source of the Package with your modifications.

   c. give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

   d. make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package through the so-called “undump” or “unexec” methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.
9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.
10. THIS PACKAGE IS PROVIDED “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

# Perl

Oracle is required to provide the text of the third-party license, but the third-party program will be subject to the Oracle license, and Oracle will NOT provide warranties and technical support for the third-party technology.

This program contains third-party code from Perl. Under the terms of the Perl license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the Perl software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the Perl software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or Perl.

## Perl Kit Readme

Copyright 1989-2001, Larry Wall

All rights reserved.

This program is free software; you can redistribute it and/or modify it under the terms of either:

**a.** the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or

**b.** the "Artistic License" which comes with this Kit.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See either the GNU General Public License or the Artistic License for more details.

You should have received a copy of the Artistic License with this Kit, in the file named "Artistic". If not, I'll be glad to provide one.

You should also have received a copy of the GNU General Public License along with this program in the file named "Copying". If not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA or visit their Web page on the internet at http://www.gnu.org/copyleft/gpl.html.

For those of you that choose to use the GNU General Public License, my interpretation of the GNU General Public License is that no Perl script falls under the terms of the GPL unless you explicitly put said script under the terms of the GPL yourself. Furthermore, any object code linked with perl does not automatically fall under the terms of the GPL, provided such object code only adds definitions of subroutines and variables, and does not otherwise impair the resulting interpreter from executing any standard Perl script. I consider linking in C subroutines in this manner to be the moral equivalent of defining subroutines in the Perl language itself. You may sell such an object file as proprietary provided that you provide or offer to provide the Perl source, as specified by the GNU General Public License. (This is merely an alternate way of specifying input to the program.) You may also sell a binary produced by the dumping of a running Perl script that belongs to you, provided that you provide or offer to provide the Perl source as specified by the GPL. (The fact that a Perl interpreter and your code are in the same binary file is, in this case, a form of mere aggregation.) This is my interpretation of the GPL. If you still have concerns or difficulties understanding my intent, feel free to contact me. Of course, the Artistic License spells all this out for your protection, so you may prefer to use that.

## mod_perl 1.26 License

```
/* ====================================================================
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 1996-2000 The Apache Software Foundation.  All rights
 * reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. The end-user documentation included with the redistribution,
 *    if any, must include the following acknowledgment:
 *       "This product includes software developed by the
 *        Apache Software Foundation (http://www.apache.org/)."
```

```
*    Alternately, this acknowledgment may appear in the software itself,
*    if and wherever such third-party acknowledgments normally appear.
*
* 4. The names "Apache" and "Apache Software Foundation" must
*    not be used to endorse or promote products derived from this
*    software without prior written permission. For written
*    permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache",
*    nor may "Apache" appear in their name, without prior written
*    permission of the Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* ====================================================================
*/
```

## Perl Artistic License

The “Artistic License”

### Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

### Definitions

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
   a. place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.
   b. use the modified Package only within your corporation or organization.
   c. rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

**d.** make other distribution arrangements with the Copyright Holder.

**4.** You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

**a.** distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

**b.** accompany the distribution with the machine-readable source of the Package with your modifications.

**c.** give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

**d.** make other distribution arrangements with the Copyright Holder.

**5.** You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

**6.** The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package through the so-called “undump” or “unexec” methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

**7.** C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.
9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.
10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

# mod_dav

mod_dav has been licensed to Oracle free of charge by Greg Stein under a license similar to the Apache Software Foundation license. The following copyright notice applies to mod_dav and Oracle's use of mod_dav:

```
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

  ------------------------------------------------------------------------
Greg Stein
Last modified: Thu Feb 3 17:34:42 PST 2000
```

# FastCGI

Oracle is required to provide the text of the third-party license, but the third-party program will be subject to the Oracle license, and Oracle will NOT provide warranties and technical support for the third-party technology.

This program contains third-party code from FastCGI. Under the terms of the FastCGI license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the FastCGI software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the FastCGI software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or FastCGI.

## FastCGI Developer's Kit License

This FastCGI application library source and object code (the "Software") and its documentation (the "Documentation") are copyrighted by Open Market, Inc ("Open Market"). The following terms apply to all files associated with the Software and Documentation unless explicitly disclaimed in individual files.

Open Market permits you to use, copy, modify, distribute, and license this Software and the Documentation solely for the purpose of implementing the FastCGI specification defined by Open Market or derivative specifications publicly endorsed by Open Market and promulgated by an open standards organization and for no other purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions.

No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this Software and Documentation may be copyrighted by their authors and need not follow the licensing terms described here, but the modified Software and Documentation must be used for the sole purpose of implementing the FastCGI specification defined by Open Market or derivative specifications publicly endorsed by Open Market and promulgated by an open standards organization and for no other purpose. If modifications to this Software and Documentation have new licensing terms, the new terms must protect Open Market's proprietary rights in the Software and Documentation to the same extent as these licensing terms and must be clearly indicated on the first page of each file where they apply.

Open Market shall retain all right, title and interest in and to the Software and Documentation, including without limitation all patent, copyright, trade secret and other proprietary rights.

OPEN MARKET MAKES NO EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE SOFTWARE OR THE DOCUMENTATION, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL OPEN MARKET BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY DAMAGES ARISING FROM OR RELATING TO THIS SOFTWARE OR THE DOCUMENTATION, INCLUDING, WITHOUT LIMITATION, ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES OR SIMILAR DAMAGES, INCLUDING LOST PROFITS OR LOST DATA, EVEN IF OPEN MARKET HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS". OPEN MARKET HAS NO LIABILITY IN CONTRACT, TORT, NEGLIGENCE OR OTHERWISE ARISING OUT OF THIS SOFTWARE OR THE DOCUMENTATION.

## Module mod_fastcgi License

This FastCGI application library source and object code (the "Software") and its documentation (the "Documentation") are copyrighted by Open Market, Inc ("Open Market"). The following terms apply to all files associated with the Software and Documentation unless explicitly disclaimed in individual files.

Open Market permits you to use, copy, modify, distribute, and license this Software and the Documentation solely for the purpose of implementing the FastCGI specification defined by Open Market or derivative specifications publicly endorsed by Open Market and promulgated by an open standards organization and for no other purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions.

No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this Software and Documentation may be copyrighted by their authors and need not follow the licensing terms described here, but the modified Software and Documentation must be used for the sole purpose of implementing the FastCGI specification defined by Open Market or derivative specifications publicly endorsed by Open Market and promulgated by an open standards organization and for no other purpose. If modifications to this Software and Documentation have new licensing terms, the new terms must protect Open Market's proprietary rights in the Software and Documentation to the same extent as these licensing terms and must be clearly indicated on the first page of each file where they apply.

Open Market shall retain all right, title and interest in and to the Software and Documentation, including without limitation all patent, copyright, trade secret and other proprietary rights.

OPEN MARKET MAKES NO EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE SOFTWARE OR THE DOCUMENTATION, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL OPEN MARKET BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY DAMAGES ARISING FROM OR RELATING TO THIS SOFTWARE OR THE DOCUMENTATION, INCLUDING, WITHOUT LIMITATION, ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES OR SIMILAR DAMAGES, INCLUDING LOST PROFITS OR LOST DATA, EVEN IF OPEN MARKET HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED “AS IS”. OPEN MARKET HAS NO LIABILITY IN CONTRACT, TORT, NEGLIGENCE OR OTHERWISE ARISING OUT OF THIS SOFTWARE OR THE DOCUMENTATION.

# Jaxen

Oracle is required to provide the text of the third-party license, but the third-party program will be subject to the Oracle license, and Oracle will NOT provide warranties and technical support for the third-party technology.

This program contains third-party code from Jaxen. Under the terms of the Jaxen license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the Jaxen software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the Jaxen software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or Jaxen.

## The Jaxen Software License

Copyright (C) 2000-2002 bob mcwhirter & James Strachan. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.
3. The name "Jaxen" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact license@jaxen.org.
4. Products derived from this software may not be called "Jaxen", nor may "Jaxen" appear in their name, without prior written permission from the Jaxen Project Management (pm@jaxen.org).

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following: "This product includes software developed by the Jaxen Project (http://www.jaxen.org/)." Alternatively, the acknowledgment may be graphical using the logos available at http://www.jaxen.org/.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE Jaxen AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Jaxen Project and was originally created by bob mcwhirter and James Strachan. For more information on the Jaxen Project, please see http://www.jaxen.org/.

# Expat

Oracle is required to provide the text of the third-party license, but the third-party program will be subject to the Oracle license, and Oracle will NOT provide warranties and technical support for the third-party technology.

This program contains third-party code from Expat. Under the terms of the Expat license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the Expat software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the Expat software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or Expat.

## Expat License

# SAXPath

Oracle is required to provide the text of the third-party license, but the third-party program will be subject to the Oracle license, and Oracle will NOT provide warranties and technical support for the third-party technology.

This program contains third-party code from SAXPath. Under the terms of the SAXPath license, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the SAXPath software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the SAXPath software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or SAXPath.

## The SAXPath License

Copyright (C) 2000-2002 werken digital. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.
3. The name "SAXPath" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact license@saxpath.org.
4. Products derived from this software may not be called "SAXPath", nor may "SAXPath" appear in their name, without prior written permission from the SAXPath Project Management (pm@saxpath.org).

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following: "This product includes software developed by the SAXPath Project (http://www.saxpath.org/)." Alternatively, the acknowledgment may be graphical using the logos available at http://www.saxpath.org/.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE SAXPath AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This software consists of voluntary contributions made by many individuals on behalf of the SAXPath Project and was originally created by bob mcwhirter and James Strachan. For more information on the SAXPath Project, please see. */

# Glossary

### Apache

Apache is a public domain HTTP server derived from the National Center for Supercomputing Applications (NCSA).

### authentication

The process of verifying the identity of a user, device, or other entity in a host system, often as a prerequisite to granting access to resources in a system. A recipient of an authenticated message can be certain of the message's origin (its sender). Authentication is presumed to preclude the possibility that another party has impersonated the sender.

### availability

The percentage or amount of scheduled time that a computing system provides application service.

### CA

See **certificate authority**.

### certificate

Also called a **digital certificate**. An ITU x.509 v3 standard data structure that securely binds an identity to a public key.

A certificate is created when an entity's public **key** is signed by a trusted identity, a **certificate authority** The certificate ensures that the entity's information is correct and that the public key actually belongs to that entity.

A certificate contains the entity's name, identifying information, and public key. It is also likely to contain a serial number, expiration date, and information about the rights, uses, and privileges associated with the certificate. Finally, it contains information about the certificate authority that issued it.

### certificate authority

A trusted third party that certifies that other entities—users, databases, administrators, clients, servers—are who they say they are. When it certifies a user, the certificate authority first seeks verification that the user is not on the certificate revocation list (CRL), then verifies the user's identity and grants a certificate, signing it with the certificate authority's private **key**. The certificate authority has its own certificate and public key which it publishes. Servers and clients use these to verify signatures the certificate authority has made. A certificate authority might be an external company that offers certificate services, or an internal organization such as a corporate MIS department.

### CGI

Common Gateway Interface (CGI) is the industry-standard technique for transferring information between a Web server and any program designed to accept and return data that conforms to the CGI specifications.

### ciphertext

Data that has been encrypted. Cipher text is unreadable until it has been converted to plain text (decrypted) with a key. See **decryption**.

### cipher suite

A set of authentication, encryption, and data integrity algorithms used for exchanging messages between network nodes. During an SSL handshake, for example, the two nodes negotiate to see which cipher suite they will use when transmitting messages back and forth.

### cleartext

See **plaintext**.

### cryptography

The art of protecting information by transforming it (encrypting) into an unreadable format. See **encryption**.

### DAD

See **database access descriptor**.

### database access descriptor

A database access descriptor (DAD) is a set of values that specify how an application connects to an Oracle database to fulfill an HTTP request. The information in the DAD includes the username (which also specifies the schema and the privileges), password, connect-string, error log file, standard error message, and national language support (Globalization Support) parameters such as Globalization Support language.

### decryption

The process of converting the contents of an encrypted message (**ciphertext**) back into its original readable format (**plaintext**).

### DES

Data Encryption Standard. A commonly used symmetric **key encryption** method that uses a 56-bit key.

### Diffie-Hellman key negotiation algorithm

Diffie-Hellman key negotiation algorithm is a method that lets two parties communicating over an insecure channel to agree upon a random number known only to them. Though the parties exchange information over the insecure channel during execution of the Diffie-Hellman key negotiation algorithm, it is computationally infeasible for an attacker to deduce the random number they agree upon by analyzing their network communications. Oracle Advanced Security uses the Diffie-Hellman key negotiation algorithm to generate session keys.

### digital certificate

See **certificate**.

### digital wallet

See **wallet**.

### directory information tree

A hierarchical tree-like structure consisting of the DNs of the directory entries. See **distinguished name**.

### distinguished name

The unique name of a directory entry. It comprises all of the individual names of the parent entries back to the root in the **directory information tree**.

### DIT

See **directory information tree**.

### DN

See **distinguished name**.

### encryption

The process of disguising a message thereby rendering it unreadable to any but the intended recipient. Encryption is performed by translating data into secret code. There are two main types of encryption: **public-key encryption** (or asymmetric-key encryption) and symmetric-key encryption.

### entry

In the context of a directory service, entries are the building blocks of a directory. An entry is a collection of information about an object in the directory. Each entry is composed of a set of attributes that describe one particular trait of the object. For example, if a directory entry describes a person, that entry can have attributes such as first name, last name, telephone number, or e-mail address.

### failover

The ability to reconfigure a computing system to utilize an alternate active component when a similar component fails.

### HTTP

See **Hypertext Transfer Protocol**.

### Hypertext Transfer Protocol

Hypertext Transfer Protocol (HTTP) is the underlying format used by the Web to format and transmit messages and determine what actions Web servers and browsers should take in response to various commands. HTTP is the protocol used between Oracle Database and clients.

### key

A password or a table needed to decipher encoded data.

### LDAP

See **Lightweight Directory Access Protocol**.

### Lightweight Directory Access Protocol

A standard, extensible directory access protocol. It is a common language that **LDAP** clients and servers use to communicate. The framework of design conventions supporting industry-standard directory products, such as the Oracle Internet Directory.

### MD5

A hashing algorithm intended for use on 32-bit machines to create digital signatures. MD5 is a **one-way hash function**, meaning that it converts a message into a fixed string of digits that form a **message digest**.

### message digest

Representation of text as a string of single digits. It is created using a formula called a **one-way hash function**.

### modules

Modules extend the basic functionality of the Web server and support integration between Oracle HTTP Server and other Oracle Database components.

### one-way hash function

An algorithm that turns a message into a single string of digits. "One way" means that it is almost impossible to derive the original message from the string of digits. The calculated **message digest** can be compared with the message digest that is decrypted with a **public key** to verify that the message has not been tampered with.

### OPMN

See **Oracle Process Manager and Notification Server**.

### Oracle Process Manager and Notification Server

Oracle Process Manager and Notification Server (OPMN) manages Oracle HTTP Server processes within an application server instance. It channels all events from different components to all components interested in receiving them.

### PEM

Privacy-Enhanced Electronic Mail. An **encryption** technique that provides encryption, authentication, message integrity, and **key** management.

### PL/SQL

PL/SQL is Oracle's proprietary extension to the SQL language. PL/SQL adds procedural and other constructs to SQL that make it suitable for writing applications.

### plaintext

Also called cleartext. Unencrypted data in ASCII format.

### port

A port is a number that TCP uses to route transmitted data to and from a particular program.

### private key

In **public-key cryptography**, this **key** is the secret key. It is primarily used for decryption but is also used for encryption with digital signatures. See **public/private key pair**.

### proxy server

A proxy server typically sits on a network firewall and enables clients behind the firewall to access Web resources. All requests from clients go to the proxy server rather than directly to the destination server. The proxy server forwards the request to the destination server and passes the received information back to the client. The proxy server channels all Web traffic at a site through a single, secure port; this enables an organization to create a secure firewall by preventing Internet access to internal machines, while allowing Web access.

### public key

In **public-key cryptography**, this key is made public to all. It is primarily used for encryption but can be used for verifying signatures. See **public/private key pair**.

### public-key cryptography

Encryption method that uses two different random numbers (**key**s). See **public key** and **public-key encryption**.

### public-key encryption

The process where the sender of a message encrypts the message with the public **key** of the recipient. Upon delivery, the message is decrypted by the recipient using its private key.

### public/private key pair

A set of two numbers used for **encryption** and **decryption**, where one is called the **private key** and the other is called the **public key**. Public **key**s are typically made widely available, while private keys are held by their respective owners. Though mathematically related, it is generally viewed as computationally infeasible to derive the private key from the public key. Public and private keys are used only with asymmetric encryption algorithms, also called **public-key encryption** algorithms, or public-key cryptosystems. Data encrypted with either a public key or a private key from a key pair can be decrypted with its associated key from the key-pair. However, data encrypted with a public key cannot be decrypted with the same public key, and data encrypted with a private key cannot be decrypted with the same private key.

### RSA

A **public-key encryption** technology developed by RSA Data Security. The RSA algorithm is based on the fact that it is laborious to factor very large numbers. This makes it mathematically unfeasible, because of the computing power and time required to decode an RSA **key**.

### scalability

A measure of how well the software or hardware product is able to adapt to future business needs.

### SHA

See **Secure Hash Algorithm**.

### Secure Hash Algorithm

Secure Hash Algorithm assures data integrity by generating a 160-bit cryptographic message digest value from given data. If as little as a single bit in the data is modified, the Secure Hash Algorithm checksum for the data changes. Forgery of a given data set in a way that will cause the Secure Hash Algorithm to generate the same result as that for the original data is considered computationally infeasible.

An algorithm that takes a message of less than 264 bits in length and produces a 160-bit message digest. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.

### Secure Shell

Secure Shell (SSH) is a well known protocol and has widely available implementation that provide a secure connection tunneling solution. SSH provides a daemon on both the client and server sides of a connection. Clients connect to the local daemon rather than connecting directly to the server. The local SSH daemon then establishes a secure connection to the daemon on the server side. Communication is then routed from the client, through the client side daemon to the server side daemon and then on to the actual server. This enables a client/server program that uses an insecure protocol to be tunneled through a secure channel. For our purposes, the disadvantage of SSH is that it requires two hops to occur and that the implementations available do not perform and scale well enough. More information on SSH can be obtained from

`http:www.ssh.org`

### Secure Sockets Layer

Secure Sockets Layer (SSL) is a standard for the secure transmission of documents over the Internet using HTTPS (secure HTTP). SSL uses digital signatures to ensure that transmitted data is not tampered with.

### SSL

See **Secure Sockets Layer**.

### SSH

See **Secure Shell**.

### wallet

Also called a **digital wallet**. A wallet is a data structure used to store and manage security credentials for an individual entity. It implements the storage and retrieval of credentials for use with various cryptographic services. A **Wallet Resource Locator** (WRL) provides all the necessary information to locate the wallet.

### Wallet Resource Locator

A wallet resource locator (WRL) provides all necessary information to locate a wallet. It is a path to an operating system directory that contains a wallet.

### WRL

See **Wallet Resource Locator**.

### X.509

Public **key**s can be formed in various data formats. The X.509 v3 format is one such popular format.

# Index

## A



## B



## C

## D

## E

## F



## G



## H



## I



## J



## K

## L



## M

## N



## O

## P



## R

## S

## T



## U



## V



## W



## X