

Oracle9i Application Server

Concepts

Release 2 (9.0.2)

May 2002

Part No. A95926-02

ORACLE®

Copyright © 2002 Oracle Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

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

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle*MetaLink*, Oracle Store, Oracle9i, Oracle9iAS Discoverer, SQL*Plus, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	xv
Preface.....	xvii
1 Introduction to Oracle9i Application Server	
The Challenges of Creating and Maintaining an E-Business	1-2
Development Challenges.....	1-2
J2EE Applications	1-3
Web Services	1-3
Enterprise Integration.....	1-4
Portals.....	1-4
Wireless.....	1-4
Personalization	1-4
Deployment Challenges	1-5
Performance	1-5
Caching	1-6
Availability	1-6
Scalability.....	1-6
User and Security Management	1-6
Systems Management	1-6
Building an E-Business Solution with Oracle9i.....	1-7

Overview of Oracle9i Application Server	1-7
Oracle9iAS Solutions	1-8
J2EE and Internet Applications	1-8
Portals	1-8
Wireless	1-9
Caching	1-9
Business Intelligence	1-9
E-Businesses Integration	1-9
Management and Security	1-10
Oracle9iAS Components	1-10

2 J2EE and Internet Application Development and Deployment

Application Development and Deployment in Oracle9iAS	2-2
What Types of Applications Can Be Developed for Deployment in Oracle9iAS?	2-2
Oracle9iAS Containers for J2EE	2-3
J2EE Application Development Concepts	2-5
What Is a J2EE Application?	2-5
J2EE Distributed Multitiered Application Model	2-5
Types of J2EE Clients	2-7
Types of J2EE Application Components	2-7
Types of J2EE Containers	2-7
J2EE Application Packaging Concepts	2-8
How OC4J Runs and Manages Servlets	2-8
About the Servlet Container	2-8
How the Servlet Container Works	2-9
How OC4J Runs and Manages JavaServer Pages	2-9
About the JSP Translator	2-10
How the JSP Translator Works	2-10
How OC4J Runs and Manages EJBs	2-10
About the OC4J EJB Container	2-10
How the EJB Container Works	2-11
J2EE Services	2-12
Oracle J2EE Services	2-13

Oracle9iAS Web Services	2-15
Oracle9iAS Web Services Architecture	2-16
Web Services Framework.....	2-16
Oracle XML Developer Kit	2-17
XML and XSLT Parsers	2-17
XML Schema Processors.....	2-18
XML Class Generators	2-18
XSQL Servlet.....	2-19
XML Transviewer Beans.....	2-19
Oracle9iAS PL/SQL Platform	2-19
Oracle9iAS PL/SQL	2-20
Oracle PL/SQL Server Pages.....	2-20
Oracle PL/SQL Web Toolkit.....	2-20
Oracle9iAS Forms Services	2-21
Oracle9iAS Forms Services Architecture	2-22
Oracle HTTP Server	2-23
Oracle HTTP Server Components.....	2-23
Oracle HTTP Server Features	2-25
Oracle HTTP Server Architecture	2-25
Modular Architecture	2-27
Common Gateway Interface (CGI) Support.....	2-27
Oracle9iAS Clustering	2-28
Clustering Terminology.....	2-28
Scalability.....	2-30
Availability	2-30
Manageability.....	2-31
Architecture	2-32
Front-End Load Balancer	2-33
Metadata Repository in the Infrastructure	2-34
Farm.....	2-34
Cluster	2-34
Application Server Instance.....	2-35
Management Features.....	2-37

High Availability	2-38
Benefits of High Availability.....	2-39
Architecture	2-40
The mod_oc4j Module	2-40

3 Portal Development and Deployment

Introduction to Oracle9iAS Portal	3-2
What is Oracle9iAS Portal?	3-2
E-Business Support with Oracle9iAS Portal	3-3
Oracle9iAS Portal Features	3-4
Portal Page Creation, Management, and Customization.....	3-5
Portal Content Publishing and Management	3-5
Content Searching.....	3-6
Content Syndication	3-7
Portals and Wireless Devices	3-8
Oracle9iAS Portal Architecture	3-8
Application Access and Integration	3-9
Integrating with Portlet Providers	3-10
Oracle9iAS Portlets.....	3-10
Custom Portlets.....	3-11
Partner Portlets	3-11
Portal Security	3-12
User Authentication in Oracle9iAS Portal	3-12
Access Control in Oracle9iAS Portal.....	3-12

4 Wireless Development and Deployment

Introduction to Oracle9iAS Wireless	4-2
What is Oracle9iAS Wireless?	4-2
Oracle9iAS Wireless Architecture	4-2
Oracle9iAS Wireless HTTP Adapter	4-3
Device/Network Adapter	4-4
Oracle9iAS Wireless Features	4-5
Core.....	4-5
Services.....	4-6

Deploy Applications for Any Device, Protocol, or Network	4-6
Location-Based Services	4-8
Mobile Personal Information Management (PIM) and E-mail Support.....	4-8
m-Commerce and Billing Support	4-9
m-Commerce APIs	4-9
Mobile Wallet.....	4-9
FormFiller	4-10
Payment Processing.....	4-10
Extending m-Commerce Drivers	4-10
Studio for Building, Testing, and Deploying Mobile Applications	4-10
Advanced Customization.....	4-11
Server Push and SMS.....	4-11
Transcoding.....	4-11
Off-Line Management.....	4-12

5 Caching for Performance

About Caching	5-2
Oracle9iAS Web Cache Overview	5-2
Performance.....	5-3
Scalability.....	5-3
High Availability	5-3
Cost Savings	5-3
Reduce Network Traffic	5-3
Oracle9iAS Web Cache Architecture	5-4
Oracle9iAS Web Cache Features	5-5
Full-Page Static and Dynamic Content Caching.....	5-5
Partial-Page Caching and Personalized Content Assembly.....	5-6
Page Assembly Components	5-6
Edge Side Includes (ESI).....	5-7
Edge Side Includes for Java (JESI).....	5-7
Invalidation for Cache Consistency Management.....	5-8
Performance Assurance Heuristics.....	5-8
Automatic Content Compression.....	5-9
Web Server Load Balancing and Failover	5-10
Cache Clustering.....	5-11

Flexible Deployment Options	5-12
Oracle9iAS Web Cache Security	5-14
Restricted Administration	5-14
Secure Sockets Layer (SSL) Support.....	5-15
Certificate Authority	5-15
Certificate.....	5-16
Wallet.....	5-16

6 Business Intelligence

Introduction to Oracle9iAS Business Intelligence	6-2
Introduction to Oracle9iAS Clickstream Intelligence	6-3
What is Oracle9iAS Clickstream Intelligence?	6-3
Oracle9iAS Clickstream Intelligence Features.....	6-5
Enterprise Warehousing Environment.....	6-5
Runtime Administrator	6-6
Customize Information Delivery	6-6
Different Data Sources Supported	6-7
Oracle9iAS Clickstream Intelligence Architecture	6-7
Introduction to Oracle9iAS Personalization.....	6-9
What is Oracle9iAS Personalization?.....	6-9
Oracle9iAS Personalization Features.....	6-10
Real-Time Recommendation Engine Architecture	6-10
Data Mining Technology	6-10
Single Administrative Interface.....	6-10
Oracle9iAS Personalization Architecture.....	6-11
Introduction to Oracle9iAS Reports Services.....	6-12
What is Oracle9iAS Reports Services?	6-12
Oracle9iAS Reports Services Features	6-14
Event-Based Reporting	6-14
Extensible Architecture.....	6-14
Enhanced Report Bursting and Distribution.....	6-15
Oracle9iAS Reports Services Architecture	6-16
Introduction to Oracle9iAS Discoverer	6-17
What is Oracle9iAS Discoverer?	6-17

Oracle9iAS Discoverer Tools	6-19
Oracle9iAS Discoverer Viewer	6-19
Oracle9iAS Discoverer Plus and Oracle9iAS Discoverer Desktop	6-20
Oracle9iAS Discoverer and Oracle9iAS Portal.....	6-20
Oracle9i Discoverer Administrator.....	6-20
Oracle9iAS Discoverer Architecture.....	6-21
Integrate Oracle9iAS Business Intelligence	6-22
Create an E-Business Dashboard Using Oracle9iAS Portal	6-22
Leverage Single Signon Functionality.....	6-22

7 Oracle9iAS Integration

About Oracle9iAS Integration	7-2
Oracle9iAS InterConnect	7-3
Oracle9iAS InterConnect Adapters.....	7-7
Oracle Workflow	7-8
Oracle Advanced Queuing.....	7-9
Additional Integration Features.....	7-10
Oracle Internet File System	7-10
Development Platform	7-10
Oracle9iAS Unified Messaging.....	7-11
Message Store.....	7-12
Open Standards-Based Messaging	7-12
Thin Client.....	7-12
Web-Based Calendar.....	7-13
Telephone Applications.....	7-13
Extended Server Side Rules	7-13
Application Integration	7-13
Administration.....	7-13

8 System Management

System Management Overview	8-2
System Management Architecture	8-2
About Management.....	8-4
Using the Oracle Enterprise Manager Web Site.....	8-4
Using the Oracle Enterprise Manager Console.....	8-4

Management Features	8-5
Single Point for Distributed Configuration and Management	8-5
Performance Monitoring	8-5
Dynamic Monitoring Service	8-6
J2EE Deployment and Administration	8-6
About Security	8-7
Security Features	8-9
Oracle9iAS Single Sign-On	8-9
Partner and External Applications	8-11
Java Authentication and Authorization Service (JAAS)	8-11
Oracle Internet Directory	8-12
Self-Service Console	8-12
Password Management	8-12
Synchronization with Third Party LDAP Servers	8-12
Key Directory Features	8-13
Using Oracle Internet Directory with Middle Tier Components	8-13

Glossary

Index

List of Figures

1-1	Application Development Challenges	1-3
1-2	Application Deployment Challenges	1-5
1-3	Oracle9i Application Server Solutions	1-8
2-1	Oracle9iAS Containers for J2EE Architecture	2-4
2-2	J2EE Distributed Multitiered Application Architecture	2-6
2-3	Oracle9iAS Forms Services Request Flow	2-22
2-4	Oracle HTTP Server Request Flow	2-24
2-5	Oracle HTTP Server Process Architecture	2-26
2-6	Oracle HTTP Server HTTP Request-Response Cycle	2-27
2-7	Oracle9iAS Cluster	2-30
2-8	Application Server Instance Failure in a Cluster	2-31
2-9	Oracle Enterprise Manager Manages a Cluster	2-32
2-10	Oracle9iAS Cluster Architecture	2-33
2-11	High Availability Architecture	2-40
3-1	Sample Portal Page	3-3
3-2	Oracle9iAS Portal Request Flow	3-9
4-1	Oracle9iAS Wireless	4-2
4-2	Oracle9iAS Wireless Architecture	4-3
4-3	Oracle9iAS Wireless - Device/Network Adapter	4-4
4-4	Oracle9iAS Wireless Core and Services	4-5
4-5	Deploy applications on any device, protocol, or network	4-7
5-1	Caching Architecture	5-4
5-2	Template Page	5-7
5-3	ESI and JESI	5-8
5-4	Performance Assurance Heuristics Graph	5-9
5-5	Oracle9iAS Web Cache Load Balancing Architecture	5-10
5-6	Oracle9iAS Web Cache Failover Architecture	5-11
5-7	Oracle9iAS Web Cache Cluster	5-12
5-8	Simple Deployment Scenario	5-13
5-9	Oracle9iAS Web Cache on Same Host as Application Server	5-13
5-10	Oracle9iAS Web Cache on Different Host from Application Server	5-14
5-11	SSL for Server Connection	5-15
6-1	Oracle9iAS Clickstream Intelligence Data Acquisition and Processing	6-5
6-2	Oracle9iAS Clickstream Intelligence Architecture	6-8
6-3	Oracle9iAS Personalization Architecture	6-11
6-4	Oracle Report Example	6-13
6-5	Oracle9iAS Reports Services Architecture	6-16
6-6	Oracle9iAS Discoverer Result Set Example	6-18
6-7	Oracle9iAS Discoverer Architecture	6-21
7-1	Oracle9iAS Application Integration Overview	7-3

7-2	Oracle9iAS InterConnect	7-5
7-3	Agent and Bridge Components of an Adapter	7-8
8-1	Oracle9i Application Server System Management Architecture.....	8-3
8-2	Security Architecture of Oracle9i Application Server	8-8
8-3	Oracle9iAS Single Sign-On Architecture.....	8-9

List of Tables

1-1	Oracle9i Application Server Components	1-10
2-1	Supported Technologies and Programming Languages	2-2
2-2	Oracle9iAS Containers for J2EE Supported APIs	2-3
2-3	Prominent Oracle HTTP Server Modules	2-23
2-4	Oracle9i Application Server Clustering Terminology	2-29
7-1	Oracle9iAS InterConnect Adapters	7-7

Send Us Your Comments

Oracle9i Application Server Concepts, Release 2 (9.0.2)

Part No. A95926-02

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: iasdocs_us@oracle.com
- FAX: 650-506-7407 Attn: Oracle9i Application Server Documentation Manager
- Postal service:

Oracle Corporation
Oracle9i Application Server Documentation
500 Oracle Parkway, M/S 2op3
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 preface contains these topics:

- [Intended Audience](#)
- [Documentation Accessibility](#)
- [Organization](#)
- [Related Documentation](#)
- [Conventions](#)

Intended Audience

Oracle9i Application Server Concepts is intended for anyone with an interest in Oracle9i Application Server or the latest Internet technologies.

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 Corporation 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 Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Organization

This document contains:

Chapter 1, "Introduction to Oracle9i Application Server"

This chapter provides an overview of application servers and Oracle9i Application Server.

Chapter 2, "J2EE and Internet Application Development and Deployment"

This chapter describes the Java 2 Platform, Enterprise Edition (J2EE) and Internet application development and deployment environment in Oracle9i Application Server.

Chapter 3, "Portal Development and Deployment"

This chapter provides an overview of Oracle9iAS Portal features and benefits.

Chapter 4, "Wireless Development and Deployment"

This chapter provides an overview of Oracle9iAS Wireless features and benefits.

Chapter 5, "Caching for Performance"

This chapter provides an overview of Oracle9i Application Server caching features and benefits.

Chapter 6, "Business Intelligence"

This chapter provides an overview of the Oracle9iAS business intelligence features and benefits.

Chapter 7, "Oracle9iAS Integration"

This chapter describes Oracle9iAS integration features and benefits.

Chapter 8, "System Management"

This chapter provides an overview of Oracle9i Application Server system management features and benefits.

Glossary

The glossary defines terminology used throughout this guide and the Oracle9iAS documentation set.

Related Documentation

For more information, see these Oracle resources:

- Oracle9i Application Server Documentation Library
- Oracle9i Application Server Platform-Specific Documentation on Oracle9i Application Server Disk 1

In North America, printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

<http://www.oraclebookshop.com/>

Other customers can contact their Oracle representative to purchase printed documentation.

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/admin/account/membership.html>

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/docs/index.htm>

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 Microsoft 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 .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle9i Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> 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. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. 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. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter <code>sqlplus</code> to open SQL*Plus. The password is specified in the <code>orapwd</code> file. Back up the datafiles and control files in the <code>/disk1/oracle/dbs</code> directory. The <code>department_id</code> , <code>department_name</code> , and <code>location_id</code> columns are in the <code>hr.departments</code> table. Set the <code>QUERY_REWRITE_ENABLED</code> initialization parameter to <code>true</code> . Connect as <code>oe</code> user. The <code>JRepUtil</code> class implements these methods.
lowercase italic monospace (fixed-width) font	Lowercase italic monospace font represents placeholders or variables.	You can specify the <code>parallel_clause</code> . Run <code>Uold_release.SQL</code> where <code>old_release</code> 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.	<code>DECIMAL (digits [, precision])</code>
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	<code>{ENABLE DISABLE}</code>
	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.	<code>{ENABLE DISABLE}</code> <code>[COMPRESS NOCOMPRESS]</code>

Convention	Meaning	Example
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> ■ That we have omitted parts of the code that are not directly related to the example ■ That you can repeat a portion of the code 	<pre>CREATE TABLE ... AS subquery; SELECT col1, col2, ... , coln FROM employees;</pre>
.	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	<pre>acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;</pre>
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	<pre>CONNECT SYSTEM/system_password DB_NAME = database_name</pre>
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.	<pre>SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;</pre>
lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	<pre>SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;</pre>

Conventions for Microsoft Windows Operating Systems

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

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Oracle Database Configuration Assistant, choose Start > Programs > Oracle - <i>HOME_NAME</i> > 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 <i>command prompt</i> in this manual. 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:\oracle\oradata> C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\" C:\>imp SYSTEM/ <i>password</i> FROMUSER=scott TABLES=(emp, dept)
<i>HOME_NAME</i>	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 <i>HOME_NAME</i> TNSListener

Convention	Meaning	Example
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default used one of the following names:</p> <ul style="list-style-type: none"> ■ C:\orant for Windows NT ■ C:\orawin95 for Windows 95 ■ C:\orawin98 for Windows 98 <p>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is C:\oracle. If you install Oracle9i release 1 (9.0.1) on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is C:\oracle\ora90. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p> <p>Refer to <i>Oracle9i Database Getting Starting for Windows</i> for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.</p>	Go to the <i>ORACLE_BASE\ORACLE_HOME\rdms\admin</i> directory.

Introduction to Oracle9i Application Server

This chapter provides an overview of Oracle9i Application Server. The topics include:

- [The Challenges of Creating and Maintaining an E-Business](#)
- [Building an E-Business Solution with Oracle9i](#)
- [Overview of Oracle9i Application Server](#)

The Challenges of Creating and Maintaining an E-Business

The Internet introduces significant opportunities for companies to reach new markets and to streamline their internal business processes. At the same time, these opportunities generate new challenges because applications must be delivered quickly and must serve vast numbers of users.

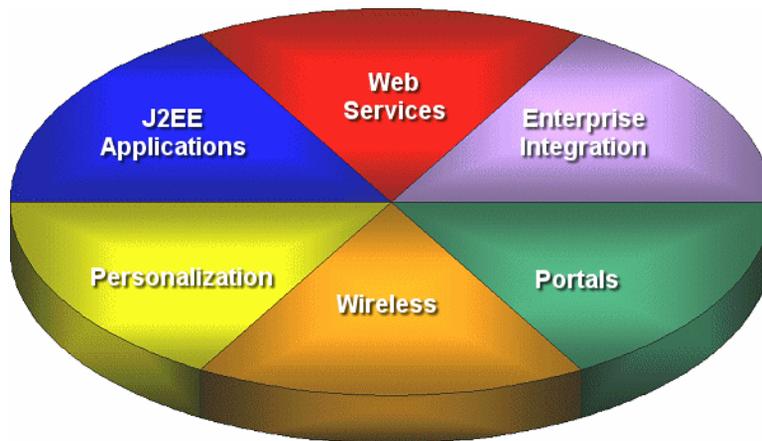
Quickly delivering highly scalable applications often requires integrating diverse products that were not designed to work together. When this is the only way to get a consolidated set of business applications, companies must spend large amounts of capital, time, and effort, both initially and continuously, to support the complicated technologies or systems that they have built.

Companies that create e-business Web sites commonly experience problems and challenges associated with developing and deploying applications.

Development Challenges

Developing applications in a complex Internet environment involves dozens of interfaces, programming languages, and platforms. Assembling the infrastructure to support such applications requires comprehensive knowledge on Internet application developers. In addition, developing, testing, and deploying applications is very time consuming.

There are numerous challenges associated with application development. It is especially challenging to develop applications that incorporate Java2 Platform, Enterprise Edition (J2EE) requirements, Web services, enterprise integration, portals, wireless, and personalization (see [Figure 1-1](#)).

Figure 1–1 Application Development Challenges

J2EE Applications

Most e-business application developers face the fact that development cycles have become shorter, while demand for an increasing number of programming languages and multiple platforms keeps growing. In recent years, developers began using Java, a platform-independent programming language, in application development. However, their applications are still vendor-dependent because of proprietary Application Programming Interfaces (APIs). Complying with industry standards to create a level of cross-vendor standardization has become an increasing demand and challenge. The J2EE platform and component specifications define a standard platform for developing multi-tier, Web-based enterprise applications. J2EE provides a solution to the industry problem.

Web Services

As e-business evolves, companies rely increasingly on their Web sites to conduct business. Accessing information efficiently, and presenting vital information and application interfaces in a unified manner through their Web sites, becomes crucial for most companies. Providing reliable and efficient Web services is important since Web services directly affect business efficiency.

Enterprise Integration

Most companies need to integrate their existing applications and data sources with new business processes. As companies grow, it is essential to integrate existing applications and data sources with those of their partners, customers, and suppliers. These companies must perform a complete enterprise application integration without rewriting the applications or building extensive customized program logic.

Portals

Portals allow users to access information through Web browsers and to combine information from different sources into a single entry point. Portals also support personalized views, so that users or user groups can customize the content and the appearance of the portal to suit individual preferences and requirements. Secure, easy-to-use, and dynamic portals are essential infrastructure elements for e-business.

Wireless

An increasing number of people conduct business out of the office. Wireless allows users to rapidly access any existing application through mobile devices, such as Web-enabled phones, personal digital assistants, or pagers. However, wireless applications are limited by small screen size, limited data entry capacity, and heterogeneity in standards supporting wireless devices.

Personalization

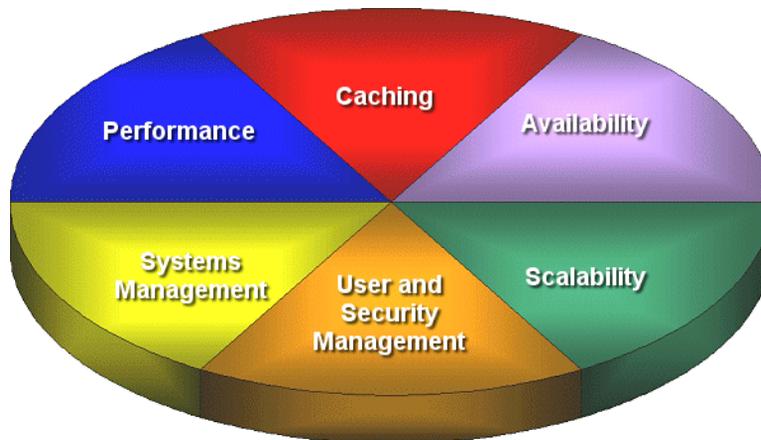
As companies become e-businesses, the amount of data that tracks the fundamentals and operations of the business increases. Web users provide valuable information about themselves and their interests. When this information is extracted effectively, you can deliver tailored customer experiences and drive business decisions.

Deployment Challenges

The key to the success of any Web site is how quickly and reliably the server can deliver the appropriate content to users. If a server takes too long to respond or if it fails, then users will take their business elsewhere. If content and applications are not secure, then sensitive information or information technology assets are vulnerable. If deployed applications cannot use hardware efficiently, then information technology budgets are depleted quickly on equipment purchases.

Concerns associated with application deployment, summarized in [Figure 1-2](#), include performance, caching, availability, scalability, user and security management, and systems management.

Figure 1-2 Application Deployment Challenges



Performance

Increasing the performance of your Web site and increasing the speed of your applications without redesigning or rebuilding the Web site are common goals. Typical performance factors include the ability to handle multiple requests simultaneously, competition for resources, latency, response time, service time, throughput, wait time, and scalability.

Caching

Many high-volume Web sites serve thousands of users concurrently and need to provide them with accurate data in a dynamic environment where content changes frequently. Caching is one of the key technologies that promises to alleviate the computational and economic burdens resulting from complicated Web site infrastructure and technologies. However, it is challenging to cache the dynamic, personalized content that the e-business model requires.

Availability

Availability of an overall system is measured by the percentage of time that it works normally. A successful e-business requires a 100% operation without downtime. To improve availability, companies may use redundant components, but only when the back-up component can take over for a failed primary component immediately.

Scalability

The scalability of a system refers to how well it responds as user demands increase. The number of Web users has increased tremendously in the past few years. It is crucial to maintain high quality Web sites that scales when you increase your hardware resources.

User and Security Management

With the Internet continually growing, the threat to information traveling over the network increases exponentially. Maintaining a secure Web site is crucial. Once your Web site is running, you must protect the databases and the servers on which they reside. You must administer and protect the rights of internal database users, and you must guarantee the confidentiality of e-business customers as they access your database.

Systems Management

Administrators of application servers require sophisticated tools to monitor and manage the systems infrastructure, and the privileges and access rights of individual users. Maintaining a simplified process in managing your Web site allows you to manage the site infrastructure and monitor the performance from one integrated console.

Building an E-Business Solution with Oracle9i

Oracle9i Application Server (Oracle9iAS) is a part of the Oracle9i platform, a complete and integrated e-business platform. Oracle9i platform consists of:

- **Oracle9i Developer Suite** for developing applications
- **Oracle9i Application Server** for deploying Internet applications
- **Oracle9i Database Server** for storing content

To deliver scalable and high performance e-business solutions successfully, you must be able to leverage an integrated, comprehensive, flexible, and open platform. Oracle9i provides integrated development, deployment, and management tools, as well as a runtime platform. The Oracle9i platform simplifies the process for creating and deploying the applications that run your business on the Internet.

See Also: The product pages on the Oracle Technology Network for more information about Oracle9i Database Server and Oracle9i Developer Suite at <http://otn.oracle.com>

Overview of Oracle9i Application Server

Oracle9iAS is a completely standards-based application server that provides a comprehensive and fully integrated platform for running Web sites, J2EE applications, and Web services. It addresses all the challenges that you face as you refine your business processes to become an e-business.

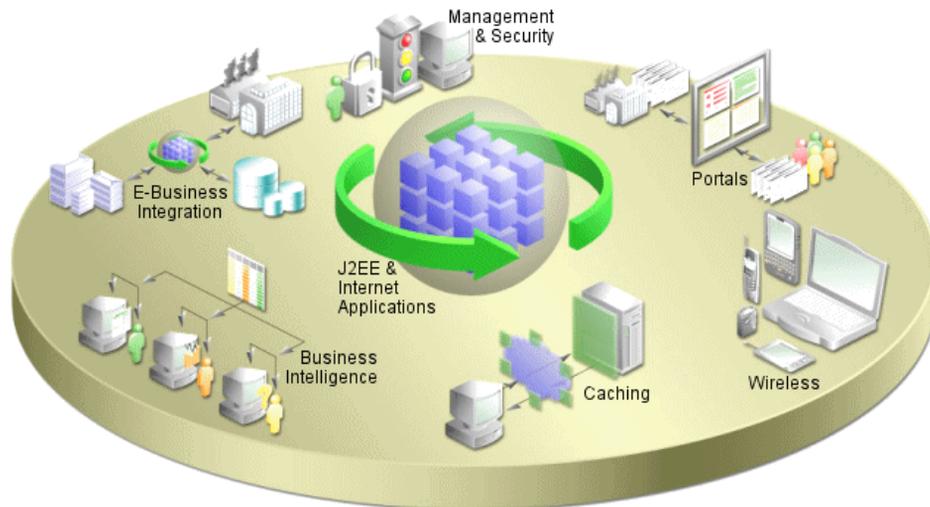
Oracle9iAS provides full support for the J2EE platform, XML, and emerging Web services standards. With Oracle9iAS you can simplify information access for your customers and trading partners by delivering enterprise portals, which can be customized and accessed from a network browser or wireless devices. It allows you to redefine your business processes, and integrate your applications and data sources with those from your customers or partners. You can deliver tailored customer experiences via real-time personalization, and assess and correlate Web site traffic patterns using Oracle9iAS integrated business intelligence services.

You can also implement a centralized management, security, and directory framework to manage and monitor all of your distributed systems and diverse user communities. Oracle9iAS allows you to save on Web site infrastructure by deploying your fast, scalable Internet applications through built-in Web caching, load balancing and clustering capabilities.

Oracle9iAS Solutions

Figure 1–3 illustrates solutions that Oracle9iAS provides, which include J2EE and Internet applications, portals, wireless, caching, business intelligence, e-business integration, and management and security.

Figure 1–3 Oracle9i Application Server Solutions



J2EE and Internet Applications

Oracle9iAS is built entirely on a J2EE framework that supports the latest industry standard technologies and programming languages, including J2EE API specifications, XML, and Web services. This comprehensive and flexible framework allows you to design, develop, and deploy dynamic Web sites, portals, and transactional applications using familiar programming languages and technologies. Oracle9iAS also provides comprehensive Web services to expose business functions to authorized parties over the Internet from any Web device.

Portals

Oracle9iAS provides an out-of-the-box portal that does not require endless programming and maintenance. You can use Oracle9iAS to build, deploy, and maintain self-service and integrated enterprise portals. Oracle9iAS allows for self-service content management and publishing, wizard-based development, and deploying, publishing, and consuming Web services on an extensible framework.

Wireless

Oracle9iAS Wireless simplifies wireless development and deployment by providing the ability to deliver content to any device, to use any protocol, and to work across any wireless network. In addition, Oracle9iAS Wireless includes wireless services, such as e-mail and location-based services, that simplify wireless-enabling applications and portals. Oracle9iAS provides application developers independence from the underlying wireless infrastructure. Oracle9iAS Wireless is built on the core Oracle9iAS infrastructure, leveraging open standards support in XML and J2EE to deliver a high-performance and scalable wireless infrastructure.

Caching

Oracle9iAS provides a Web caching solution with the unique capability of caching both static and dynamically generated Web content. The Oracle9iAS Web cache significantly improves the performance and scalability of heavily loaded Web sites. In addition, the Web cache provides a number of features to ensure consistent and predictable responses. These features include page fragment caching, dynamic content assembly, Web server load balancing, Web cache clustering, and failover.

Business Intelligence

Oracle9iAS provides comprehensive personalization and business intelligence services through Web analytic applications. Using Oracle9iAS business intelligence features, you can generate reports on Web site activity by analyzing clickstream data; dynamically serve personalized content recommendations to both registered and anonymous visitors as they browse your site; perform dynamic, ad-hoc query reporting and analysis using a standard Web browser; and publish high quality, dynamically generated reports on a scalable, secure platform.

E-Businesses Integration

Oracle9iAS has a powerful set of features that provide communications and integration capabilities for e-business applications. Using Oracle9iAS, you can integrate enterprise applications, trading partners, and Web services, emphasizing scalability and manageability; provide seamless query and transaction access to many non-Oracle data sources; provide integrated e-mail, voice, and fax messaging for access through multiple client devices; and support multiple media types in a centrally managed file system that resides on a database.

Management and Security

Oracle9iAS provides a set of management facilities that are based on industry standards to simplify all aspects of Web site administration. Using Oracle9iAS, you can:

- Configure and monitor Oracle9iAS instances to optimize them for performance and scalability from a centralized console
- Respond proactively to problem conditions from a centralized console
- Use encrypted Secure Sockets Layer (SSL) connections, user and client certificate-based authentication, and single signon across all applications
- Implement an LDAP directory that provides a single repository and administration environment for user accounts

Oracle9iAS Components

Table 1-1 presents the Oracle9iAS components that are associated with these solutions.

Table 1-1 Oracle9i Application Server Components

Solution	Oracle9iAS Components
J2EE and Internet Applications	Oracle HTTP Server Oracle9iAS Containers for J2EE Oracle9iAS Web Services Oracle PL/SQL Oracle9iAS Forms Services Oracle XML Developer Kit Oracle9i Client
Portals	Oracle9iAS Portal Oracle9iAS Portal Developer Kit
Wireless	Oracle9iAS Wireless
Caching	Oracle9iAS Web Cache
Business Intelligence	Oracle9iAS Reports Services Oracle9iAS Discoverer Oracle9iAS Personalization Oracle9iAS Clickstream Intelligence

Table 1–1 Oracle9i Application Server Components (Cont.)

Solution	Oracle9iAS Components
E-Business Integration	Oracle9iAS InterConnect Oracle9iAS Unified Messaging Oracle Internet File System
Management and Security	Oracle Enterprise Manager Oracle9iAS Single Sign-On Oracle Internet Directory Oracle9iAS Infrastructure

J2EE and Internet Application Development and Deployment

This chapter describes the Java 2 Platform, Enterprise Edition (J2EE) and Internet application development and deployment environment in Oracle9i Application Server (Oracle9iAS), explaining its features and concepts. The topics include:

- [Application Development and Deployment in Oracle9iAS](#)
- [Oracle9iAS Containers for J2EE](#)
- [Oracle9iAS Web Services](#)
- [Oracle XML Developer Kit](#)
- [Oracle9iAS PL/SQL Platform](#)
- [Oracle9iAS Forms Services](#)
- [Oracle HTTP Server](#)
- [Oracle9iAS Clustering](#)
- [High Availability](#)

Application Development and Deployment in Oracle9iAS

Oracle9iAS provides an integrated, standards-based infrastructure that makes it easy to develop applications for deployment, and that reliably delivers those deployed applications to users across an enterprise. The following sections describe how this infrastructure aids application development and deployment.

What Types of Applications Can Be Developed for Deployment in Oracle9iAS?

Oracle9i Application Server allows Web application developers to program their site in a variety of languages and technologies:

- Java and J2EE including Web services
- XML
- PL/SQL

[Table 2–1](#) lists the different technologies and programming languages that you can use to build applications for deployment with Oracle9iAS.

Table 2–1 Supported Technologies and Programming Languages

Java and J2EE	XML	PL/SQL
<ul style="list-style-type: none"> ■ JavaServer Pages (JSP) ■ Servlets ■ Enterprise JavaBeans (EJB) ■ Java Database Connectivity (JDBC) ■ Java Transaction API (JTA) ■ Java Naming and Directory Interface (JNDI) ■ Java Message Service (JMS) ■ Java Authentication and Authorization Service (JAAS) ■ Web Services 	<ul style="list-style-type: none"> ■ Document Object Model (DOM) ■ Extensible Stylesheet Language Transformations (XSLT) ■ XML Schemas ■ Simple API for XML (SAX) ■ XSQL ■ Internet Data Access Presentation (IDAP) ■ XML Path Language (XPath) ■ Simple Object Access Protocol (SOAP) ■ Web Services Description Language (WSDL) ■ Universal Description, Discovery, and Integration (UDDI) 	<ul style="list-style-type: none"> ■ PL/SQL Server Pages ■ PL/SQL Web Toolkit ■ Oracle Forms

The following sections explain how Oracle9iAS supports these technologies and programming languages.

Oracle9iAS Containers for J2EE

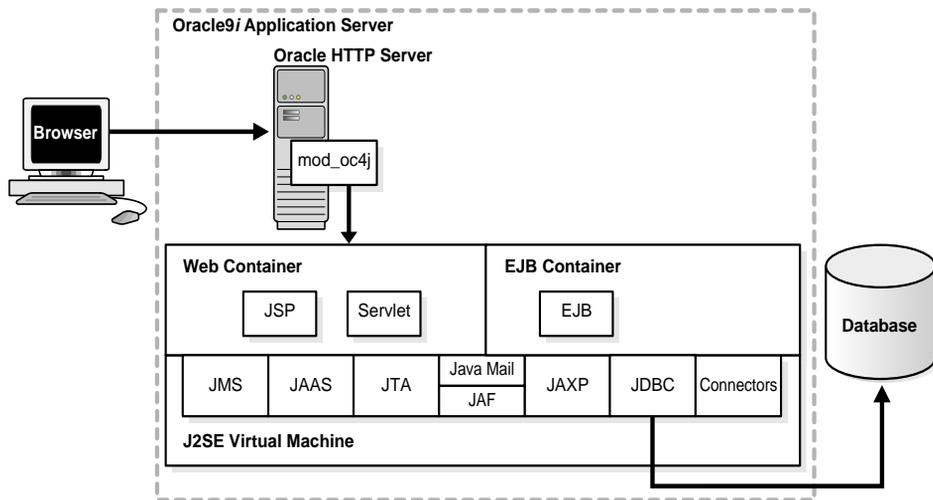
Oracle9iAS Containers for J2EE (OC4J) are fast, lightweight, and scalable J2EE containers that are written in Java and run on a standard Java Virtual Machine (JVM). They have been designed for ease of use and to support the standard APIs in [Table 2-2](#).

Table 2-2 Oracle9iAS Containers for J2EE Supported APIs

API	Version
JavaServer Pages (JSP)	1.2
Java Servlet	2.3
Enterprise JavaBeans (EJB)	2.0
Java Database Connectivity (JDBC)	2.0
Java Transaction API (JTA)	1.0
Java Message Service (JMS)	1.0
JavaMail	1.2
JavaBeans Activation Framework	1.0
Java API for XML (JAXP)	1.1
J2EE Connector API	1.0
Java Authentication and Authorization Service (JAAS)	1.0

[Figure 2-1](#) shows the architecture of OC4J within Oracle9i Application Server.

Figure 2–1 Oracle9iAS Containers for J2EE Architecture



OC4J is supported by the Java 2 Platform, Standard Edition (J2SE) infrastructure as shown in [Figure 2–1](#). This means that the OC4J Web container and OC4J EJB container use the J2SE virtual machine. J2EE applications are modularized for reuse of application components, such as

- User interfaces, which can be composed of JavaServer Pages (JSPs), dynamic HTML, and so on,
- Business logic, which is usually contained in an Enterprise JavaBean (EJB) or normal Java classes.

The J2EE containers also perform services for applications, such as providing access to the APIs and lifecycle management.

J2EE Application Development Concepts

The following sections provide introductory definitions and summaries of application programming technologies that are supported by Oracle9iAS. For detailed information on the technologies introduced in this section, refer to the following information sources.

See Also:

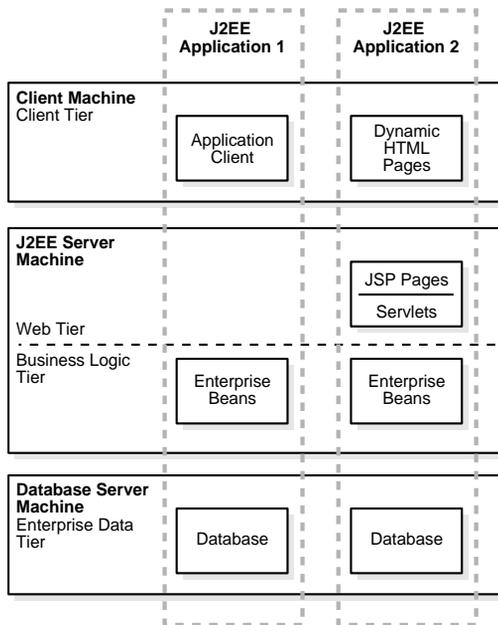
- *The J2EE Tutorial* at <http://java.sun.com>
- The Oracle9i Application Server Documentation Library

What Is a J2EE Application?

J2EE applications are composed of components. A J2EE component is a self-contained functional software unit that is assembled into a J2EE application with its related classes and XML descriptor files that communicate with other components. J2EE components are written in the Java programming language and are compiled in the same way as any program written in Java.

J2EE Distributed Multitiered Application Model

The J2EE platform provided in Oracle9i Application Server uses a multitiered distributed application model. A multitiered distributed application model divides application logic into components according to function, and the various application components that make up J2EE applications can be installed on different machines depending on which tier in the multitiered J2EE environment the application component belongs. [Figure 2-2](#) shows two multitiered J2EE applications divided into the client, Web, business logic, and enterprise data tiers.

Figure 2–2 J2EE Distributed Multitiered Application Architecture

You can distribute J2EE applications across the four tiers shown in this figure, but generally they are considered to be three-tier applications because they are usually distributed over the following machine locations:

- client machines
- J2EE server machines hosting presentation services, like JSPs and servlets, and business logic components, like EJBs
- database servers or legacy machines at the back end.

Three-tiered applications that run in this way extend the standard two-tiered client and server model by placing an application server between the client and the back-end storage.

Types of J2EE Clients

J2EE applications support the following clients:

- **Application Clients:** Applications running on a client machine that directly access enterprise beans that are running in the business logic tier. Application clients can also open an HTTP connection to establish communication with a servlet running on the Internet tier if a J2EE application requires it.
- **Dynamic HTML and XML Pages:** In the context of J2EE applications, dynamic HTML and XML pages are generated by servlets or pages created with JavaServer Pages technology running in the Web tier. These pages can be extensions to traditional static HTML pages allowing application developers to offer customized and personalized pages to the client.

Types of J2EE Application Components

You can use the following components in J2EE applications:

- **Servlets:** A servlet is a Java class used to extend the capabilities of servers that host applications accessed through a request-response programming model.
- **JavaServer Pages:** JavaServer pages (JSPs) are text files that contain two types of information: static template data, which can be expressed in any text-based format, such as HTML, WML (Wireless Markup Language), or XML, and JSP elements, which construct dynamic content.
- **Enterprise Beans:** Enterprise beans are server-side components that encapsulate the business logic of an application.

Types of J2EE Containers

Containers are the interface between a component and the low-level platform-specific functionality that supports the component. Before a Web or enterprise bean component can run, it must be assembled into a J2EE application and deployed into a J2EE container. The assembly process involves specifying container settings for each component, which customize the underlying support provided by the J2EE server. These settings can be standard J2EE settings or container-specific settings depending on your application requirements. Some of the container settings that you can specify include security services, transaction model, naming and directory lookup, or remote connectivity model.

The J2EE containers are:

- **Enterprise JavaBeans Container:** The Enterprise JavaBeans (EJB) container manages the execution of all enterprise beans for J2EE applications. Enterprise beans and their container run on the J2EE server.
- **Web Container:** The OC4J Web container manages the execution of servlets and JSPs for J2EE applications in the servlet container. The container provides services such as request dispatching, security, concurrency, and life cycle management. The Web container also gives Web components, typically JSPs and servlets, access to the J2EE APIs such as naming, transactions, and e-mail.

J2EE Application Packaging Concepts

J2EE components are packaged separately and bundled into a J2EE application. Each component, with its related files such as GIF and HTML files or server-side utility classes, are packaged together with a deployment descriptor (DD), and are assembled into a module that is added to the J2EE application. Typically, a J2EE application is composed of one or more enterprise beans and Web or application client component modules.

How OC4J Runs and Manages Servlets

A servlet is a Java program that runs on a J2EE server, such as OC4J. Think of a servlet as the server-side counterpart to a Java applet. A servlet is one of the application component types of a J2EE application. It must execute under the control of a servlet container, which is part of the OC4J Web container. The servlet container calls the servlet's methods and provides services that the servlet needs when running.

See Also: ["Types of J2EE Application Components"](#) on page 2-7

About the Servlet Container

The servlet container provides the servlet with access to properties of the HTTP request, such as headers and parameters. Also, the container provides the servlet with access to other Java APIs, such as JDBC to access a database, Remote Method Invocation (**RMI**) to call remote objects, or JMS to perform asynchronous messaging.

How the Servlet Container Works

When a request is mapped to a servlet, the servlet container performs the following steps:

1. If an instance of the servlet does not exist, the container:
 - a. Loads the servlet class.
 - b. Instantiates an instance of the servlet class.
 - c. Initializes the servlet instance.
2. The container then invokes the servlet, passing request and response objects. The request object contains information about the client, request parameters, and HTTP headers. The response object returns the servlet's output to the client.

The servlet extracts information from the client request, accesses external resources, and then populates the response based on that information.

How OC4J Runs and Manages JavaServer Pages

JavaServer Pages provide a convenient way to generate dynamic content in Web pages. JSP technology, which is closely coupled with Java servlet technology, allows you to include Java code fragments and call, in the form of tags and directives, external Java components from within your Web pages. Typically the markup code used to compose your Web pages is HTML or XML. JSPs work well as a front-end for business logic and dynamic functionality in JavaBeans or Enterprise JavaBeans (EJBs).

A JSP is translated into a Java servlet before being run, and it processes HTTP requests and generates responses like any servlet. However, JSP technology provides a more convenient way to code a servlet. Translation occurs the first time the application is run. A JSP translator is triggered by the `.jsp` file name extension in a URL.

JSPs are fully interoperable with servlets. You can include output from a servlet or forward the output to a servlet, and a servlet can include output from a JSP or forward output to a JSP.

About the JSP Translator

The JSP translator has a translator and a compiler. The JSP translator translates a JSP into a Java source file. The container compiles the source file into a Java bytecode (`.class`) file, which executes as a servlet in the servlet container using the JSP runtime library. The servlet container provides access to Java APIs and other services.

How the JSP Translator Works

When a user requests a URL that maps to a JSP file, such as `http://host/Hello.jsp`, the following steps occur:

1. The Web server invokes the JSP translator, which translates `Hello.jsp` and produces the file `Hello.java`.
2. The Java compiler is invoked, creating a `Hello.class` servlet.
3. `Hello.class` runs, using the JSP runtime library, which contains the supporting files to interpret the tags and directives from the JSP.
4. If the `Hello` class requires information from a database, then the servlet container provides JDBC access to the class so it can retrieve the information and return its output to the client browser.

How OC4J Runs and Manages EJBs

The OC4J EJB container manages the execution of enterprise beans for J2EE applications. Like the OC4J Web container, the EJB container uses the J2SE virtual machine. The following sections describe what services the EJB container provides to J2EE applications and how it works.

Enterprise beans are the J2EE components that implement Enterprise JavaBeans technology. Enterprise beans run in the EJB container. An enterprise bean is a portable server-side component that encapsulates the business logic of an application. There are three types of EJBs: **session beans**, **entity beans**, and **message-driven beans**.

About the OC4J EJB Container

The OC4J EJB container provides system-level services to EJBs similar to the services that the Web container provides to servlets and JSPs. The container has configurable settings that customize the underlying support provided by OC4J, the J2EE server. The configurable settings include security, transaction management, Java Naming and Directory Interface (JNDI) lookups, and remote connectivity. In

addition to the configurable settings, the container also manages EJB life cycles, database connection resource pooling, data persistence, and access to the J2EE APIs.

How the EJB Container Works

How the EJB container works depends on what type of enterprise bean you are using. The container manages the execution of the enterprise bean for one J2EE application.

For session beans, the EJB container provides all of the services that the Web container provides to Web components, such as access to APIs and the virtual machine, transaction services like Container Managed Transactions (CMTs), and secure and authorized EJB method invocation.

For entity beans, which represent business objects in a persistent storage mechanism, you can choose to have the bean manage its own persistence, or you can choose to have the container manage it. Persistence means that the entity bean's state exists beyond the lifetime of the application or the server process, like the data in a database, which is also persistent.

Depending on what entity manages a beans persistence, you can have either **bean-managed persistence (BMP) beans** or **container-managed persistence (CMP) beans**. With bean-managed persistence, the entity bean code contains the calls that access the database and the EJB container triggers callback methods on your code. Entity beans with bean-managed persistence execute in the EJB container with the typical container support and services. However, if you are using container-managed persistence, then the EJB container automatically generates the necessary database access calls. The EJB methods do not require any JDBC code to manage EJB data persistence.

J2EE Services

J2SE provides core services for writing J2EE components. The J2EE containers manage access to these services for the application components. The services are:

- **Java Database Connectivity (JDBC)** lets you invoke SQL commands from Java programming methods. You use the JDBC API in an enterprise bean when you override the default container-managed persistence or have a session bean access the database. You can also use the JDBC API from a servlet or a JSP to access a database directly without going through an enterprise bean.

Oracle9i Application Server includes the Oracle JDBC driver and DataDirect Connect Type 4 JDBC drivers. These drivers provide highly scalable and reliable connectivity to both Oracle and non-Oracle data sources.

- **Java Message Service (JMS)** is a messaging standard that allows J2EE application components to create, send, receive, and read messages. It enables distributed communication that is loosely coupled, reliable, and asynchronous.
- **Java Transaction API (JTA)** provides a standard demarcation interface for demarcating transactions. Typically, it is used in J2EE applications that use two or more, separate database access operations that depend on each other to demarcate where the entire transaction begins, rolls back, and commits.

OC4J provides additional support for two-phase commits for applications that require commit coordination across machines and containers. The two-phase commit engine is responsible for ensuring that when a distributed transaction ends, changes to all participating databases are all either committed or rolled back.

- **Java Naming and Directory Interface (JNDI)** provides a standard interface to naming and directory services. J2EE applications use JNDI to find other distributed objects. The JNDI Interface has two parts: an application-level interface used by application programs to access naming and directory services, and a service provider interface to attach a provider of naming and directory services.
- **JavaMail Technology** provides J2EE applications with a JavaMail API and a JavaMail service provider with which to send e-mail notifications.
- **Java API for XML (JAXP)** enables J2EE components to read and handle programs or tools that use XML.
- **J2EE Connector API** provides an interface for resource adaptors that allow J2EE applications to access and interact with databases and other enterprise information systems.

- **Java Authentication and Authorization Service (JAAS)** provides a way for a J2EE application to authenticate users against different security provider systems and authorize a specific user or group of users to run J2EE applications under role permission and control enforcement.

Oracle9i Application Server provides an implementation of Java Authentication and Authorization Service (JAAS) that integrates with the Oracle9iAS J2EE security infrastructure to enforce security constraints for Web (servlets and JSPs) and EJB components. The Oracle JAAS implementation:

- Integrates with Java-based applications with Oracle9iAS Single Sign-On, includes authentication, thereby giving you extensible security for Java-based applications.
- Manages access control policies centrally in Oracle Internet Directory, controls access by role, and partitions security policy by subscriber.
- Supports impersonation of a specific user, allows an enterprise bean, servlet, or JSP to run with the permissions associated with the current client or a specified user.

See Also: The Sun Microsystems, Inc. published specifications for these services, and *The J2EE Tutorial* for instructions on how to use them in J2EE applications, at:

<http://java.sun.com>

Oracle J2EE Services

In addition to the standard J2EE Services, Oracle also provides the following services to Java developers:

- **Java Object Cache** stores frequently accessed or resource-intensive objects in memory or on disk. This eliminates the need to repeatedly create and load information within a Java application. The Java Object Cache retrieves content faster and greatly reduces the load on Oracle9i Application Server.

- **Oracle9i Business Components for Java (BC4J)** is a Java and XML framework that enables productive development, portable deployment, and flexible customization of multi-tier, database applications from reusable business components. Oracle Business Components is an application component framework providing developers with a set of intelligent software building-blocks that cooperate to manage all of the common facilities required to:
 - Productively author and test business logic in components which automatically integrate with relational databases
 - Flexibly reuse business logic through multiple SQL-based views of data, supporting different application tasks
 - Efficiently access and update the views from Servlets, JavaServer Pages (JSPs), XML clients and thin-Java Swing clients
 - Easily customize application functionality in layers without requiring modification of the delivered application

Note: Oracle9i Business Components for Java ships with Oracle9i JDeveloper in Oracle9i Developer Suite.

- **Oracle9iAS MVC Framework for J2EE** is a Model-View Controller (MVC) framework that provides development teams with a flexible, common, and systematic way of developing applications for potentially any deployment scenario.

Note: Oracle9iAS MVC Framework for J2EE is available from the Oracle Technology Network at

<http://otn.oracle.com/products/ias>

Oracle9iAS Web Services

The move to transform businesses to e-businesses has driven organizations around the world to begin to use the Internet to manage corporate business processes. Despite this transformation, business on the Internet still functions as a set of local hosts, or Web sites, with point-to-point communications between them. As more business moves online, the Internet should no longer be used in such a static manner, but rather should be used as a universal business network through which services can flow freely, and over which applications can interact and negotiate amongst themselves.

To enable this transformation, the Internet needs to support a standards-based infrastructure that enables companies and their enterprise applications to communicate with other companies and their applications more efficiently. These standards should allow discrete business processes to expose and describe themselves on the Internet, allow other services to locate and invoke them, and provide a predictable response.

Web services drive this transformation by promising a fundamental change in the way businesses function and enterprise applications are developed and deployed. A Web service is a discrete business process that:

- **Exposes and describes itself:** A Web service defines its functionality and attributes so that other applications can understand it. A Web service makes this functionality available to other applications.
- **Allows other services to locate it on the Web:** A Web service can be registered in an electronic Yellow Pages, so that applications can easily locate it.
- **Can be invoked:** Once a Web service has been located and examined, the remote application can invoke the service using an Internet standard protocol.
- **Returns a response:** When a Web service is invoked, the results are passed back to the requesting application over the same Internet standard protocol used to invoke the service.

Web services provide a standards based infrastructure through which any business can do the following:

- Offer appropriate internal business processes as value-added services that can be used by other organizations.
- Integrate its internal business processes and dynamically link them with those of its business partners.

Oracle9iAS Web Services Architecture

Oracle9iAS Web Services run as servlets in the OC4J servlet container. This gives Web services the same scalability, availability, and load balancing facilities that all J2EE applications have in Oracle9iAS.

See Also: ["How OC4J Runs and Manages Servlets"](#) on page 2-8

Web Services Framework

The following Internet standards provide the Web services framework:

- **Simple Object Access Protocol (SOAP)** is a lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment. SOAP supports different styles of information exchange, including: Remote Procedure Call style (RPC) and message-oriented exchange. RPC style information exchange allows for request-response processing, where an endpoint receives a procedure oriented message and replies with a correlated response message. Message-oriented information exchange supports organizations and applications that need to exchange business or other types of documents where a message is sent but the sender may not expect or wait for an immediate response.
- **Web Services Description Language (WSDL)** is an XML format for describing network services containing RPC-oriented and message-oriented information. Programmers or automated development tools can create WSDL files to describe a service and can make the description available over the Internet. Client-side programmers and development tools can use published WSDL descriptions to obtain information about available Web services and to build and create proxies or program templates that access available services.
- **Universal Description, Discover, and Integration (UDDI)** is an online electronic registry that serves as electronic Yellow Pages, providing an information structure where various business entities register themselves and the services they offer through their WSDL definitions. There are two types of UDDI registries, public UDDI registries that serve as aggregation points for a variety of businesses to publish their services, and private UDDI registries that serve a similar role within organizations.

Oracle XML Developer Kit

XML makes data portable and interoperable across heterogeneous systems. XML is a metamarkup language that supports markup tags that have been defined by the user to encapsulate and describe data in a Web page. Because it is a self-defined language, it is extremely flexible.

To provide basic XML infrastructure components to manipulate and transform XML documents, Oracle9iAS includes the Oracle XML Developer Kit (XDK) to support development of any applications that use XML. The XDK contains component libraries and utilities that are based on W3C specifications. You can use these components to generate, manipulate, render, and store XML-formatted data in an Oracle database or to share data between applications written in diverse programming languages. The following sections describe the tools that make the Oracle XML Developer Kit.

XML and XSLT Parsers

The XML parser APIs are defined by the W3C specifications so that standard programming interfaces can be used by application developers. However, if some functionality is not specified, then Oracle Corporation may implement its own enhancements to the specifications.

The Oracle XML and XSLT parsers provide international character set and multithreaded support, allow optional validation, and cache Document Type Definitions (DTDs) and stylesheets for performance.

Oracle XML Developer Kit provides the following APIs:

- **Document Object Model (DOM) APIs:** Because an XML document is structured in the sense that "start" tags have corresponding "end" tags and these tags are nested in an ordered fashion, an XML document can be viewed as a tree whose hosts consist of these tags and information between and corresponding to the tags. When the DOM APIs are used to navigate an XML document, the XML parser parses the document and forms a tree representation of it in memory.
- **Simple APIs for XML (SAX) APIs:** SAX APIs are event-based, meaning that notification of certain events and data encountered during the parsing of an XML document can be reported with callback functions to the application program. One advantage over DOM is that an in-memory representation of the parse tree does not have to be built, thus saving memory and improving performance. When application programs are notified of these events, then they must handle them.

- **Namespace APIs:** These interfaces resolve namespace prefixes that are used to qualify element and attribute names in XML documents. XML document namespaces are identified by Uniform Resource Identifier (URI) references that qualify element or attribute names and locate resources that may be on different machines or XML documents. Namespaces make it possible to have identical names for elements and attributes by qualifying them with URIs that differentiate the names.
- **Parser APIs:** Application programmers invoke the parser API to read an XML document and to provide access to its content and structure through DOM or SAX APIs. Typically, initialization and termination functions must also be invoked in association with the parse function. Various flags, such as to discard whitespace and turn on validation, can be set with some initialization functions before the parse function is invoked by the application program.
- **XSLT APIs:** These interfaces read the input stylesheet file and transform the input XML document according to the stylesheet. A command-line interface to the integrated XSLT processor is provided. This allows application programmers to specify the number of threads to use to parse and transform XML documents and other useful options. To enhance performance, the XSLT engines allow caching of both stylesheets and DTDs so they can be reused for multiple XML documents.

XML Schema Processors

Oracle XML schema processors comply with the Structures and Datatypes sections of the W3C XML Schema Working Drafts, except for certain features such as unique, key, keyref constraints when the SAX parser is used. XML schemas are a superset of DTDs, except for the support of primitive and complex datatypes in XML schemas. This allows you to validate XML documents, with embedded datatype information, against XML schemas.

XML Class Generators

The XML class generators create a set of Java or C++ classes that create XML documents corresponding to a DTD or XML schema definitions. This is useful when an application wants to send an XML message to another application based on an agreed-upon DTD or as the back end of a Web form to construct XML documents. You can use the generated classes to programmatically construct XML documents that comply with the DTD or schema definition.

XSQL Servlet

XSQL Servlet is a tool that processes SQL queries and outputs the result set as an XML document. This processor uses an XML file with embedded SQL queries as its input. You can use XSQL Servlet to perform the following tasks:

- Build dynamic XML datapages from the results of one or more SQL queries and serve the results over the Web as XML datagrams or HTML pages using server-side XSLT transformations.
- Receive XML posted to your Web server and insert it into your database.

XML Transviewer Beans

XML transviewer beans are a set of XML components that constitute XML for Java Beans. These are used by Java applications or applets to view and transform XML documents.

These beans are visual and non-visual Java components that are integrated into Oracle9i JDeveloper to enable the fast creation and deployment of XML-based database applications.

Oracle9iAS PL/SQL Platform

Web applications written in PL/SQL are sets of stored procedures that interact with Web browsers through the HTTP protocol:

- Visiting a Web page, following a hypertext link, or submitting an HTML form causes the database server to run a stored procedure.
- Any choices that a user makes on an HTML form are passed as parameters to the stored procedure. Parameters can also be hardcoded in the URL used to invoke the stored procedure.
- The results of the stored procedure are printed as tagged HTML text and are displayed in the browser as a Web page. Web pages generated this way are dynamic: code runs inside the database server, producing HTML that varies depending on the database contents and the input parameters.

This kind of dynamic content is different from dynamic HTML (DHTML). With DHTML, the code is downloaded as JavaScript or some other scripting language, and processed by the browser along with the HTML. A PL/SQL Web application can print JavaScript or other script code in its output, to produce complex DHTML that would be tedious to produce manually.

- The dynamic pages can contain links and HTML forms that call more stored procedures, to drill down or perform some other operation on the displayed data. The set of interlinked HTML pages forms the user interface of the Web application.
- There are many techniques for coding dynamic pages, but PL/SQL is especially good for producing dynamic pages based on database processing. Its support for DML statements, dynamic SQL, cursors, and tight server integration provide both power and flexibility for Web applications.
- Producing dynamic content using PL/SQL stored procedures gives you the flexibility and interactive behavior of CGI programs, without the memory overhead of starting a new CGI process each time.

Oracle9iAS PL/SQL

Oracle9i Application Server PL/SQL developers can create applications using either Oracle PL/SQL Server Pages or Oracle PL/SQL Web Toolkit.

Oracle PL/SQL Server Pages

To include dynamic content, including the results of SQL queries, inside Web pages, you can use server-side scripting through PL/SQL Server Pages (PSP). You can author the Web pages in a script-friendly HTML authoring tool, and drop the pieces of PL/SQL code into place. You may find this technique more convenient than using the HTP and HTF packages to write out HTML content line by line.

Because the processing is done on the database server rather than the Web server, the browser receives a plain HTML page with no special script tags, and you can support all browsers and browser levels equally. It also makes network traffic efficient by minimizing the number of server roundtrips.

Embedding the PL/SQL code in the HTML page that you create lets you write content quickly and follow a rapid, iterative development process. You maintain central control of the software, with only a Web browser required on the client machine.

Oracle PL/SQL Web Toolkit

The PL/SQL toolkit contains PL/SQL packages that allow you to create applications that generate dynamic HTML. Using the toolkit, you can access data and insert it into Web pages. The packages remove the need for you to know the specifics of HTML syntax and allows you to focus on writing applications.

Oracle9iAS Forms Services

Oracle Forms applications combine interactive, graphical interfaces with strong support for data validation. Forms developers can quickly create applications with powerful data manipulation features.

An Oracle Forms application gives you the power of a desktop application deployed in a browser. It provides a very rich and productive user interface by allowing:

- immediate data validation as clients enter data into the form instead of after submitting the form.
- automatic completion and list of value searches for fields that enable users to quickly enter correct information.

An Oracle Forms application is analogous to a traditional HTML form application. In the traditional application, the user enters data into a form on a Web page and submits the data. The Web server then processes the data from the form in a CGI application. The application then performs any necessary transactions and then returns feedback to the user's browser. JavaScript and the database perform data validation tasks to ensure that the data from the form is correct and complete.

Oracle9iAS Forms Services deploys Forms applications to Java clients in a Web environment. Oracle9iAS Forms Services automatically optimizes class downloads, network traffic, and interactions with the Oracle database. Applications are automatically load-balanced across multiple servers and, therefore, can easily scale to service any number of requests.

Oracle9iAS Forms Services consists of three components:

- **Forms Client Applet:** A Java applet running in the client browser. It provides the user interface for the corresponding Forms Runtime Process.
- **Forms Listener Servlet:** A servlet running in the OC4J servlet container. It manages the lifecycle of the Forms Runtime Process and network connections to the runtime process.
- **Forms Runtime Process:** A process running in an Oracle9iAS instance. The process maintains the connection between the client applet and the database.

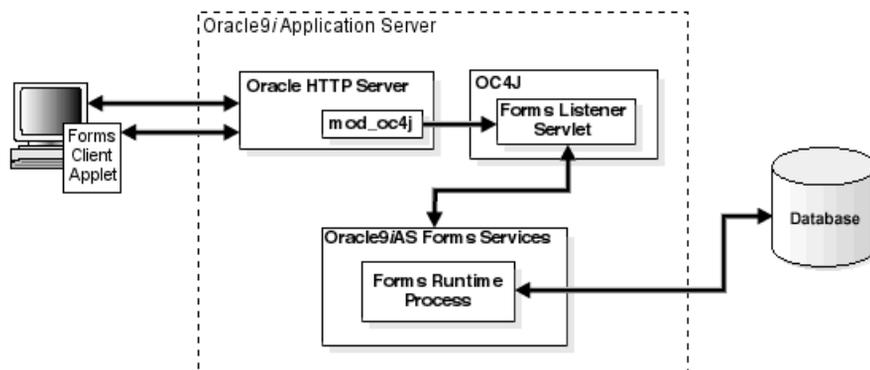
Oracle9iAS Forms Services Architecture

When a client requests an Oracle9iAS Forms application, the request has the following flow:

1. The client browser requests a forms application over HTTP or HTTPS.
2. Oracle HTTP Server receives this request.
3. Oracle HTTP Server forwards the request to the Forms Listener Servlet running in OC4J.
4. The servlet returns a page to the client that contains information the client needs to start the Forms Client Applet. The applet launches in the client browser.
5. The client applet starts a Forms session with the Forms Listener Servlet.
6. The servlet starts a new Forms Runtime Process for the client.
7. The runtime process opens a database connection and handles all interactions between the client and database.
8. The runtime process sends metadata to the client describing the interface. The Forms applet renders the interface in the client browser.

Figure 2-3 illustrates this flow in terms of the Oracle9iAS Forms Services architecture.

Figure 2-3 Oracle9iAS Forms Services Request Flow



Oracle HTTP Server

Oracle HTTP Server is the underlying deployment platform for all programming languages and technologies Oracle9iAS supports. It provides a Web listener for OC4J and the framework for hosting static and dynamic pages and applications over the Web. Based on the proven technology of the Apache HTTP Server, Oracle HTTP Server includes significant enhancements that facilitate load balancing, administration, and configuration. It also includes a number of enhanced modules, or **mods**, which are extensions to the HTTP server that extend its functionality for other enterprise applications and services.

Oracle HTTP Server allows developers to program their site in a variety of languages and technologies such as Java, Perl, C, C++, and PL/SQL. Additionally, it can serve as either a forward or reverse proxy server. The following sections describe how Oracle HTTP Server provides a robust deployment platform for dynamic Web sites and applications.

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. Major components include:

- **HTTP listener:** Oracle HTTP Server uses an Apache HTTP listener to serve client requests.
- **Modules (mods):** Many of the standard Apache mods are included with Oracle HTTP Server. Oracle also includes several internal modules that are specific to Oracle9iAS components. [Table 2-3](#) lists some of the Oracle HTTP Server modules.

Table 2-3 Prominent Oracle HTTP Server Modules

Module	Description
mod_fastcgi	Supports FastCGI, which allows C, C++, and Java CGI programs to run in a performant environment.
mod_perl	Routes requests to the Perl Interpreter.
mod_plsql	Routes requests for stored procedures to the database server.
mod_oc4j	Supports communication with Oracle9iAS Containers for J2EE and also performs some load balancing tasks.
mod_oradav	Supports file as well as database distributed authoring and versioning.

Table 2–3 Prominent Oracle HTTP Server Modules

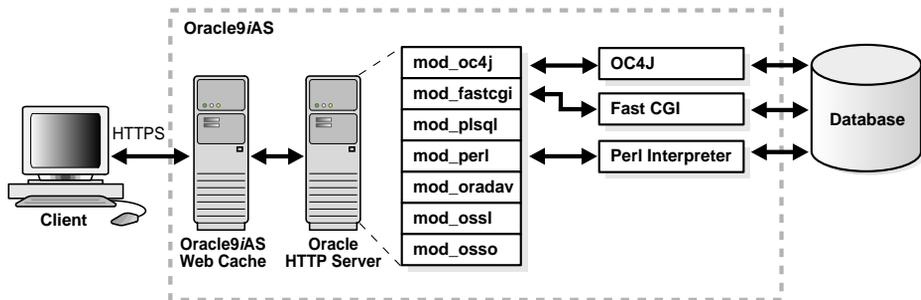
Module	Description
mod_oss1	Supports Secure Sockets Layer (SSL) and certificate sharing.
mod_osso	Routes requests to Oracle9iAS Single Sign-On server.

See Also: *Oracle HTTP Server Administration Guide* for a complete list of modules

- Perl interpreter:** A persistent Perl runtime environment embedded in Oracle HTTP Server. See also "[Common Gateway Interface \(CGI\) Support](#)" on page 2-27.

Figure 2–4 shows the path of various requests through Oracle HTTP Server components.

Figure 2–4 Oracle HTTP Server Request Flow



Oracle HTTP Server Features

In addition to the standard Web server functionality of serving client requests to other Oracle9iAS components, Oracle HTTP Server also provides enterprise support with the following features:

- **Dynamic monitoring services (DMS):** These services automatically measure runtime performance statistics for both Oracle HTTP Server and Oracle9iAS Containers for J2EE processes. As applications run, DMS collects detailed performance statistics. This data allows you to monitor the duration of important request processing phases and status information. With this information, you can locate performance bottlenecks and tune the application server to maximize throughput and minimize response time.
- **Proxy Plug-In for Using Non-Oracle HTTP Listeners:** To accommodate requirements for non-Oracle HTTP listeners, Oracle HTTP Server provides a proxy plug-in that can be plugged directly into Netscape iPlanet or Microsoft Internet Information Server (IIS). This proxy plug-in is used to forward requests for Oracle9iAS component services to Oracle HTTP Server, which is placed behind the non-Oracle listener.

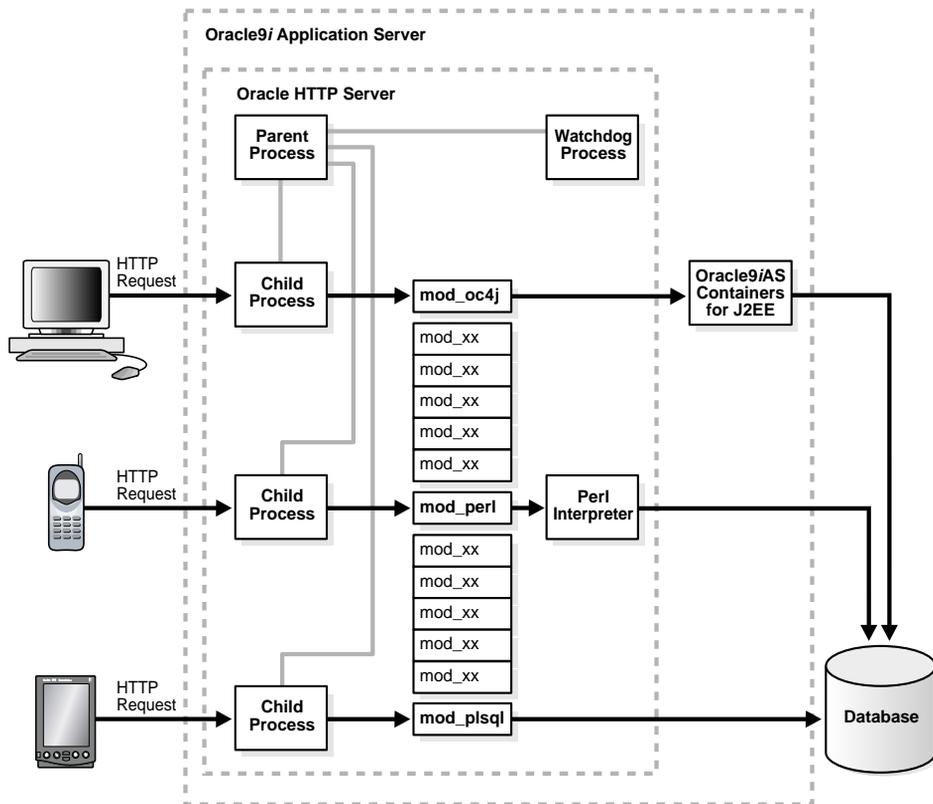
Oracle HTTP Server Architecture

At startup, the Web server parent process loads the entire configuration, the associated mods, and spawns a preconfigured number of child processes.

Note: On Windows systems, the Web server main process is a child process that spawns multiple threads.

Figure 2-5 shows the process architecture of Oracle HTTP Server in a UNIX environment.

Figure 2-5 Oracle HTTP Server Process Architecture



The parent process does not listen to HTTP requests. Its sole job is to ensure that the child processes are running or that new ones are started when the load requires it.

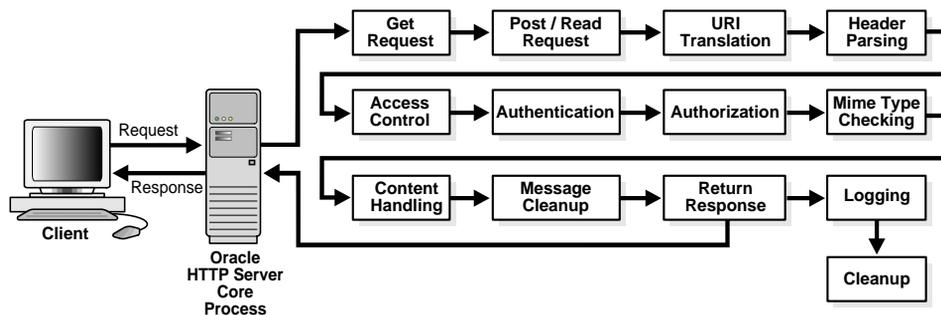
On UNIX platforms, each child process handles a single HTTP request. The child processes determine who should take the next request based on a mutex mechanism that you can configure.

Note: On Windows platforms, threads handle HTTP requests instead of child processes.

Modular Architecture

The architecture of Oracle HTTP Server is modular. The core HTTP listener is very small with all capabilities implemented as modules that plug in and are invoked at the appropriate place during the HTTP request lifecycle. [Figure 2-6](#) shows the lifecycle of an HTTP request in Oracle HTTP Server.

Figure 2-6 Oracle HTTP Server HTTP Request-Response Cycle



A child process guides the request through this entire lifecycle. The modules register their APIs, which are then either invoked automatically when the request reaches a certain stage in its lifecycle, or can be configured to be invoked only in certain situations.

Common Gateway Interface (CGI) Support

Requests that are sent to a Common Gateway Interface (CGI) program may invoke two new processes—the child process that handles the HTTP request and the CGI program itself. It is possible to avoid this overhead by configuring Oracle HTTP Server to pre-start child processes and keep them running, leveraging the Perl module to run the CGI programs in memory, or using the FastCGI mechanism.

The following Oracle HTTP Server features support CGI:

- **FastCGI:** FastCGI supports applications written with Perl, C, C++, and Java. Each CGI application runs in a single child process of the Web server. This improves server performance because it eliminates the need to start a new process for every application request.
- **Perl Interpreter:** The Perl Interpreter supports Perl applications and runs them inside of the Web server process. This allows for greater performance because running applications does not start new processes. Since the interpreter runs

inside of the Web server process, it has access to Web server services such as log files.

Oracle9iAS Clustering

A cluster is a set of application server instances configured to act in concert to deliver greater scalability and availability than a single instance can provide. While a single application server instance can only leverage the operating resources of a single host, a cluster can span multiple hosts, distributing application execution over a greater number of CPUs. While a single application server instance is vulnerable to the failure of its host and operating system, a cluster continues to function despite the loss of an operating system or host, hiding any such failure from clients.

Clusters leverage the combined power and reliability of multiple application server instances while maintaining the simplicity of a single application server instance. For example, browser clients of applications running in a cluster interact with the application as if it were running on a single server. The client has no knowledge of whether the application is running on a single application server or in an application server cluster. From the management perspective, an application server administrator can perform operations on a cluster as if the administrator was interacting with a single server. An administrator can deploy an application to an individual server; the application is propagated automatically to all application server instances in the cluster.

The following sections discuss how application server clustering increases scalability, availability, and manageability, and explain the server clustering architecture.

Clustering Terminology

[Table 2-4](#) lists the terms that Oracle9i Application Server documentation and administration tools use when referring to clustering.

Table 2–4 Oracle9i Application Server Clustering Terminology

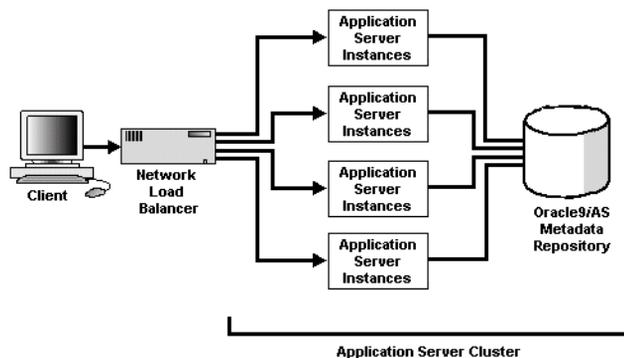
Term	Definition
Oracle9iAS Metadata Repository	The Oracle9iAS Metadata Repository maintains information about the available instances in a cluster. This simplifies the process of creating clusters and synchronizing applications and state information across a cluster because all Oracle9iAS instances share the same repository. Changing the configuration or deploying an application to one instance in a cluster triggers the infrastructure to update the other instances in the cluster with the new information.
Component Instance	Component instances include a single Oracle HTTP Server process or multiple Oracle9iAS Containers for J2EE (OC4J) instances.
Oracle9iAS Instance	An Oracle9iAS Instance (also called an application server instance) is the set of processes required to run the configured components within an application server installation. There can be only one application server instance per application server installation. The terms <i>installation</i> and <i>instance</i> are sometimes used interchangeably, however, it is important to remember that an installation is the set of files installed into an Oracle home and an instance is a set of processes associated with those files.
Oracle9iAS Cluster	A cluster is a collection of application server instances with identical configuration and application deployment. Clusters enforce homogeneity between member instances so that a cluster of application server instances can appear and function as a single instance. With appropriate front-end load balancing, any instance in an application server cluster can serve client requests. This simplifies configuration and deployment across multiple instances and enables fault tolerance among clustered instances.
Oracle9iAS Farm	A collection of clusters and instances that share the same Oracle9iAS Infrastructure.

Scalability

Oracle9iAS clustering enables you to scale your system beyond the limitations of a single application server instance on a single host. [Figure 2-7](#) shows how a cluster unifies multiple application server instances spread over multiple hosts to collectively serve a single group of applications. In this way, clustering makes it possible to serve increasing numbers of concurrent users after the capacity of a single piece of hardware is exhausted.

Clients interact with the cluster as if they are interacting with a single application server. An administrator can add an application server instance to the cluster while the cluster is operating, increasing system capacity without incurring downtime.

Figure 2-7 Oracle9iAS Cluster



Clients access the cluster through a load balancer, which hides the application server configuration. The load balancer can send requests to any application server instance in the cluster, as any instance can service any request. An administrator can raise the capacity of the system by introducing additional application server instances to the cluster, each of which derives its configuration from a shared Oracle9iAS Metadata Repository.

Availability

Oracle9iAS clustering enables you to achieve a higher level of system availability than is possible with only a single application server instance. An application running on a single instance of an application server is dependent on the health of the operating system and host on which the server is running. In this case, the host

poses as a single point of failure because if the host goes down, the application becomes unavailable.

An application server cluster eliminates the single point of failure by introducing redundancy and failover into the system. Any application server instance in the cluster can service any client request, and the failure of any single instance or host does not bring down the system. Client session state is replicated throughout the cluster, thereby protecting against the loss of session state in case of process failure. The administrator can configure the extent of session state replication.

Figure 2–8 Application Server Instance Failure in a Cluster

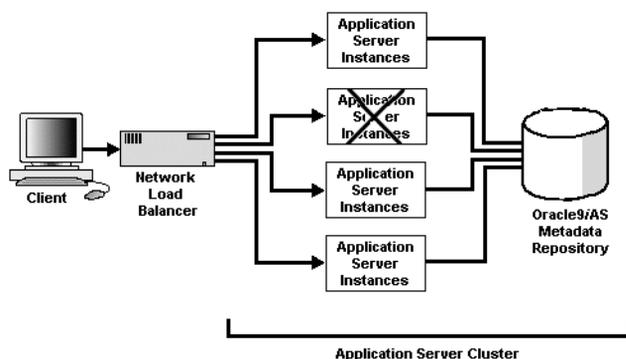


Figure 2–8 illustrates how application server clusters enable higher availability by providing redundancy and backup and eliminating a single point of failure. Clients access the cluster through a load balancer which can send requests to any application server instance in the cluster. In the case that an application server instance becomes unavailable, the load balancer can continue forwarding requests to the remaining application server instances, as any instance can service any request.

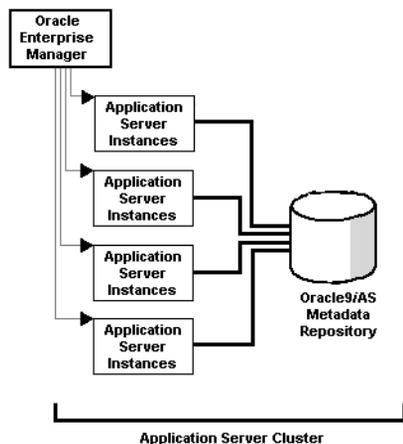
Manageability

Figure 2–9 demonstrates how managed clustering uses Oracle Enterprise Manager. While any clustered system requires all instances to be similarly configured in order to function properly, Oracle9iAS managed clustered instances synchronize their configurations automatically, relieving the administrator of the responsibility to manually update each individual instance. Using Oracle Enterprise Manager, the administrator can make configuration changes as if on a single application server

instance. Applicable changes are propagated automatically to all instances in the cluster.

Oracle9iAS cluster management simplifies the tasks of creating and administering clusters and reduces the chance of human error corrupting the system. An administrator creates a cluster in a single management operation. Then, the administrator adds the initial application server instance to the cluster to define the base configuration for the cluster. Any additional instances automatically inherit this base configuration.

Figure 2–9 Oracle Enterprise Manager Manages a Cluster



Architecture

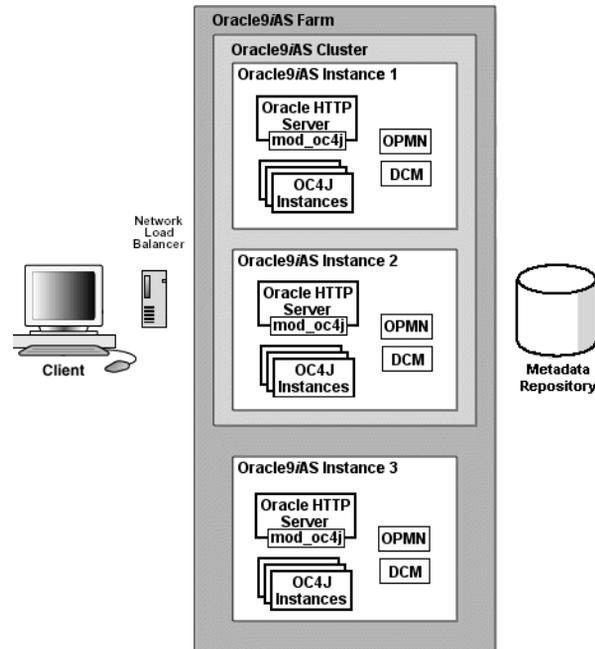
A cluster coordinates several application server instances and its components. The roles of the components included in the cluster are described in the following sections:

- [Front-End Load Balancer](#)
- [Metadata Repository in the Infrastructure](#)
- [Farm](#)
- [Cluster](#)
- [Application Server Instance](#)
- [Management Features](#)

Figure 2-10 shows the architecture of a farm and a cluster. There are three application server instances, where each instance shares the same Oracle9iAS Metadata Repository within an infrastructure. Thus, all three application server instances are part of the same farm.

Application server instances 1 and 2 are involved in a cluster together. In front of the cluster is a front-end load balancer. Included within each application server instance are its manageability features—Oracle Process Management and Notification (OPMN) and Dynamic Configuration Management (DCM)—and its installed components—Oracle HTTP Server and Oracle9iAS Containers for J2EE (OC4J).

Figure 2-10 Oracle9iAS Cluster Architecture



Front-End Load Balancer

After you have created a cluster, you can add a load balancer in front of all application server instances in the cluster, which provides availability and scalability for the application server instances.

We recommend that you purchase and install a hardware load balancer for the best performance. Alternatively, you could use Web Cache as a load balancer, which could be a single point of failure.

See Also: *Oracle9iAS Web Cache Administration and Deployment Guide* for instructions on how to set up Oracle9iAS Web Cache as your load balancer for your cluster

Metadata Repository in the Infrastructure

When you install Oracle9iAS, you have the option of installing the Oracle9iAS Infrastructure. An Oracle9iAS Infrastructure provides Oracle Internet Directory, Oracle9iAS Single Sign-On, and the Oracle9iAS Metadata Repository. The metadata repository is an Oracle9i database that is used to store the application server instance information and configuration. The application server instance tables are created in the metadata repository. Multiple application server instances can share the metadata repository of the infrastructure.

Application server instances associate with an infrastructure either during installation or through the Oracle Enterprise Manager after installation.

Farm

A farm is a group of multiple application server instances that associate with the same metadata repository. The application server instances that belong to a farm can be installed anywhere on the network.

- It is only within the constraint of a farm that you can create a cluster.
- A farm can host multiple clusters.

Cluster

A cluster is a logical group of application server instances that belong to the same farm. Each application server instance may be part of only one cluster. If an instance is part of a cluster, then all of its configured components are implicitly part of that cluster. Each application server instance can only be configured with Oracle HTTP Server and OC4J components to be contained in a cluster. A cluster can include zero or more application server instances.

All application server instances involved in the cluster have the same “cluster-wide” configuration. If you modify the configuration on one application server instance, then the modification is automatically propagated across all instances in the cluster.

Note: “Instance-specific” configuration parameter modifications are not propagated. For a description of these parameters, see the *Oracle9i Application Server Administrator’s Guide*.

Application Server Instance

An application server instance consists of a single Oracle HTTP Server and one or more OC4J instances. It is a single installation in one Oracle home. If you have multiple application servers on a single host, each is installed into its own Oracle home and uses separate port numbers.

To manage clusters from Oracle Enterprise Manager, the application server uses a metadata repository for storing its tables and configuration. Each application server instance in the cluster has the same base configuration. The base configuration contains the cluster-wide parameters and excludes instance-specific configuration. If you modify any part of the cluster-wide configuration, the modifications are propagated to all other application server instances in the cluster. If you modify an instance-specific parameter, it is not propagated, as it is only applicable to the specified application server instance. See the *Oracle9i Application Server Administrator’s Guide* for a listing of the instance-specific parameters. The cluster-wide parameters are all other parameters.

In order for each application server instance to be a part of a cluster, the following must be true:

- The application server instances you add to a cluster must be part of the same farm as the cluster, which means they all use the same metadata repository. You can associate application server instances with the same metadata repository during install time or after installation using Oracle Enterprise Manager.
- The first application server instance you add to the cluster must contain only OC4J and Oracle HTTP Server components. Web Cache can be configured, but it will be ignored for clustering operations. If other Oracle9iAS components are part of the application server instance, Oracle9iAS displays an error and does not add the application server instance to the cluster.

Note: Oracle9iAS Web Cache provides its own clustering functionality separate from application server clustering. See *Oracle9iAS Web Cache Administration and Deployment Guide* for more information.

- When you install additional application server instances, ensure that only Oracle HTTP Server, OC4J, and Oracle9iAS Web Cache are configured. The Oracle9iAS Web Cache will be ignored for clustering operations.
- Each application server instance can contain only one Oracle HTTP Server.
- Each application server instance can contain one or more OC4J instances.
- Each application server instance must reside on a host using the same operating system type (for example, UNIX) as the other instances in the cluster.

Once grouped in the same cluster, these application server instances have the following properties:

- Each application server instance has the same cluster-wide configuration. That is, if you modify any cluster-wide parameters, the modifications are propagated to all application server instances in the cluster. For instance-specific parameters, you must modify these on each individual application server instance.
- If you deploy an application to one application server instance, it is propagated to all application server instances in the cluster. The application is actually deployed to an OC4J Instance in the application server instance and propagated to the same OC4J Instance in the other application server instances in the cluster. You can change the configuration for the deployed application, and this change is propagated to the same OC4J Instance in the other application server instances in the cluster.
- Each application server instance is equal in the cluster. You can remove any of them at any time. The first instance does not have special properties. The base configuration is created from this instance, but the instance can be removed from the cluster in the same manner as the other instances.
- Most of the clustering management, configuration, and application deployment is handled through the Oracle Enterprise Manager. If you want to use a command-line tool, you can use the Distributed Configuration Management (DCM) command-line tool.
- You can remove application server instances from the cluster. The application server instance is stopped when removed from the cluster. When the last application server instance is removed, the cluster still remains. You must delete the cluster itself for it to be removed.

Management Features

Each application server instance contains management features that manage and monitor the application server instance, its components, and how it performs in a cluster. The management features do the following:

- Propagate the cluster-wide configuration for the application server instances and its components
- Manage the application server components by starting, stopping, and restarting these components
- Notice if a component dies and restarts it
- Notify the Oracle HTTP Server if any OC4J instances start or stop

All of these activities are provided by the following management features:

- [Distributed Configuration Management \(DCM\)](#)
- [Oracle Process Management and Notification \(OPMN\)](#)

Distributed Configuration Management (DCM)

Distributed Configuration Management (DCM) manages configuration by propagating the cluster-wide configuration for the application server instances and its components. When you add application server instances to the cluster, it is the DCM component that automatically replicates the base configuration to all instances in the cluster. When you modify the cluster-wide configuration, DCM propagates the changes to all application server instances in the cluster.

DCM is a management feature in each application server instance. However, it is not a process that exists at all times. DCM is invoked either by Oracle Enterprise Manager or manually by a user through `dcmctl` to do the following:

- Create or remove a cluster
- Add or remove application server instances to or from a cluster
- Synchronize configuration changes across application server instances
- Send application server instance start, restart, and stop requests to OPMN
- Enable automatic re-configuration on system failure

You can also manually execute the DCM command-line tool—`dcmctl`—to perform these duties. However, there are restrictions on how to use `dcmctl`, which are as follows:

- If Oracle Enterprise Manager is up and managing the cluster, you can invoke the DCM command-line tool from any host where a clustered application server instance exists. DCM informs Oracle Enterprise Manager of the requested function. Oracle Enterprise Manager then interfaces with the other DCM management features on the other application server instances in the cluster to complete the cluster-wide function.
- If Oracle Enterprise Manager is not up and managing the cluster, you must start the DCM command-line tool in the foreground on each application server instance in the cluster. Once started in the foreground, the DCM process in each application server instance communicate with each other about configuration changes and deployed applications.

See Also: *Oracle9i Application Server Administrator's Guide*, for directions on how to do the previous functions with the `dcmcctl` tool

Oracle Process Management and Notification (OPMN)

Oracle Process Management and Notification (OPMN) manages Oracle HTTP Server and OC4J processes within an application server instance. It channels all events from different components to all components interested in receiving them.

OPMN consists of the following two components:

- Oracle Process Manager
- Oracle Notification Manager

High Availability

The availability of a system or any component in that system is defined by the percentage of time that it works normally. A system works normally when it meets its correctness and performance specifications. For example, a system that works normally for twelve hours per day is 50% available. A system that has 99% availability is down 3.65 days per year on average. System administrators can expect critical systems to have 99.99% or even 99.999% availability. This means that the systems experience as little as four to five minutes of downtime per year.

Availability may not be constant over time. For example, availability may be higher during the daytime when most transactions occur, and lower during the night and on weekends. However, because the Internet provides a global set of users, it is a common requirement that systems are always available.

Redundant components can improve availability, but only if a spare component takes over immediately for a failed component. If it takes ten minutes to detect a component failure and twenty additional minutes to start the spare component, then the system experiences a 50% reduction in availability for that hour of service.

Oracle9iAS is designed to provide maximum system availability during many types of hardware and software failures.

Benefits of High Availability

Oracle9i Application Server keeps your system available by providing the following benefits:

- **Fault Tolerance:** Oracle9iAS has a flexible deployment model so you can deploy an architecture that exposes no single point of failure. This means that despite the failure of any component in the system, the application server is tolerant to failures and continues to function and service client requests.
- **Death Detection and Restart:** To have maximum availability, when a server process fails, the system should take appropriate action by cleaning up memory and restarting the failed processes. The management processes in each Oracle9iAS instance monitor the HTTP server and OC4J processes and restart failed processes as required.
- **Failover:** Failover hides system failure from users. Failures do occur, but clients should not perceive any loss of service as their requests are failed-over/re-routed to another redundant service that takes over for the failed one.

At the `mod_oc4j-OC4J` level, after the failure of a stateless service, the application server routes requests to alternate instances of the service in a similar fashion to connection rerouting. After a failure of a stateful service, the application server reroutes the request to an alternate instance to which the state has been replicated.

At application server instance level, you can use a hardware load balancer, Oracle9iAS Web Cache, or a DNS round robin strategy to load balance requests across redundant application server instances. This allows your Web site to continue functioning even if one of the application server instances goes down.

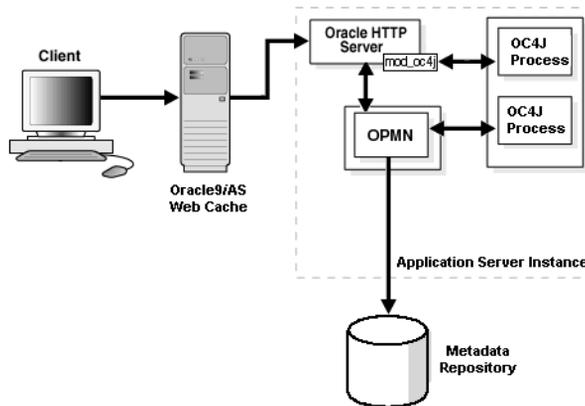
See Also: For instructions on configuring Oracle9iAS Web Cache as a load balancer to route requests, refer to *Oracle9iAS Web Cache Administration and Deployment Guide*.

- Connection Rerouting:** Rerouting occurs when a process or node servicing a request fails while a session is still executing. To address this failure, client requests to the session are transparently rerouted to alternate application server instances in the cluster.

Architecture

Figure 2–11 shows an example of high availability architecture where an Oracle9iAS instance can contain one Oracle HTTP Server instance and zero or more OC4J processes. Here, Oracle9iAS Web Cache is used as a load balancer to route requests. OPMN within the Oracle9iAS instance environment monitors the Oracle HTTP Server and OC4J processes. If either Oracle HTTP Server or OC4J fail, the OPMN restarts the failed process. This enables a maximum level of availability of the components within an Oracle9iAS instance.

Figure 2–11 High Availability Architecture



The mod_oc4j Module

The `mod_oc4j` module routes requests from the Oracle HTTP Server to Oracle Containers for J2EE (OC4J). It works in conjunction with OPMN to keep its routing table updated so that it load balances across only live OC4J processes in an OC4J instance. When a request comes in, `mod_oc4j` selects a process from its shared memory table. If a session is created with this request, related requests are routed to the same process. Otherwise, a new process is selected based on the load balancing algorithm. To allow fault tolerance, this module supports OC4J instances and islands. To handle failover with stateless requests, it tries to select an alternative

OC4J process within the same island and routes the request to that process. If all the OC4J processes in the instance have been tried and none of them are successful, `mod_oc4j` returns request failure. For session enabled requests, the module tries to failover to another OC4J process within the same island as the original process that serviced the request. Again, if all the OC4J processes in the island are unable to service the request, `mod_oc4j` returns request failure.

See Also: *Oracle HTTP Server Administration Guide, Oracle9i Application Server Administrator's Guide*

Portal Development and Deployment

This chapter provides an overview of Oracle9iAS Portal features and benefits. The topics include:

- [Introduction to Oracle9iAS Portal](#)
- [Oracle9iAS Portal Features](#)
- [Oracle9iAS Portal Architecture](#)
- [Application Access and Integration](#)
- [Portal Security](#)

Introduction to Oracle9iAS Portal

Portals allow clients to access information through any Web browser. This information usually comes from different data sources that the portal combines into a single entry point. That entry point is known as a page.

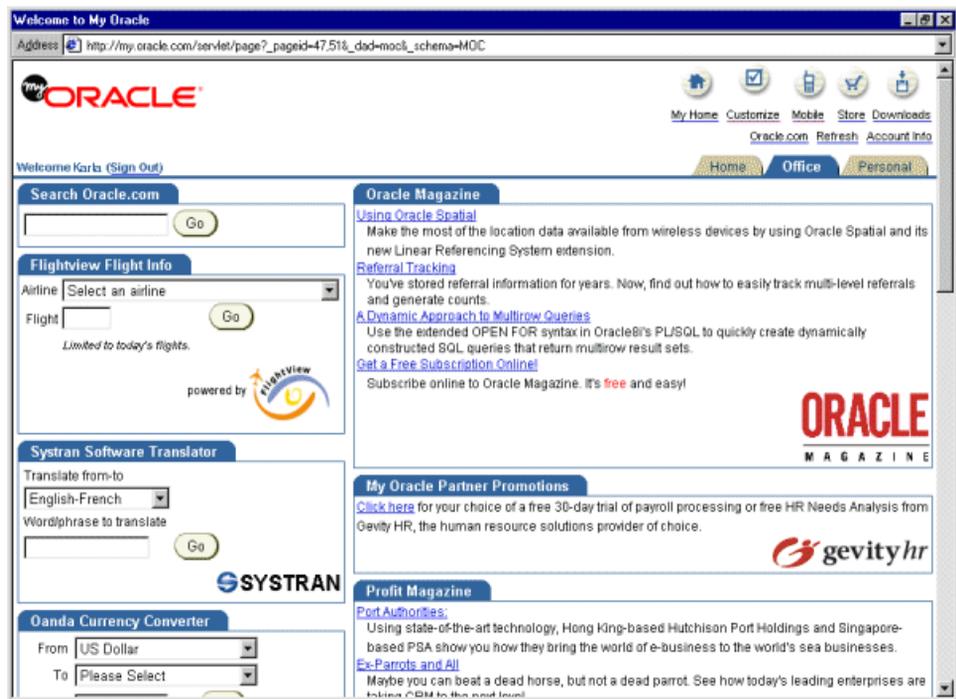
Portals also support personalized views, so that each user or user group can customize both the content and the appearance of the portal to suit individual preferences and requirements.

For example, a financial analyst's page would likely include information from real-time Internet-based stock quotes, financial reports from an online repository, and access to legacy financial accounting and banking systems. The data from these systems are independent of each other, but the portal allows them to exist within a single page.

What is Oracle9iAS Portal?

Oracle9iAS Portal is a Web-based tool for building and deploying e-business portals. It provides a secure, manageable environment for accessing and interacting with enterprise software services and information resources. It helps you to efficiently manage, access, and interact with information by enabling you to create portal pages. A portal page makes data from multiple sources accessible from a single location. [Figure 3-1](#) shows a sample portal page from <http://my.oracle.com>.

Figure 3–1 Sample Portal Page



E-Business Support with Oracle9iAS Portal

The growth of the Internet and Internet technologies creates new opportunities and challenges for businesses. The earlier vision of free-flowing information shared over corporate intranets is meeting roadblocks because of information overload, system management complexities, and conflicting data. Shortcomings in technology and its implementation are limiting the impact of a complete corporate intranet on user's core job functions. As a result, much of the potential revenue, productivity, and efficiency gains are not fully realized.

In response, these organizations are turning to portals as key components of their e-business strategy. Portals are emerging as essential problem-solving mechanisms that provide a single source of interaction with all corporate information and the

focal point for conducting day-to-day business. Companies are using portals and supporting applications:

- To provide access to instantly available, personalized, job-based information.
- In forums for groups to exchange, analyze, and discuss ideas.
- To automate business processes that integrate into daily activities.

Oracle9iAS Portal supports e-businesses by:

- **Providing secure access to existing information, no matter where it is.** Portal developers can organize and structure content in a consistent and logical way. This reduces or eliminates the need for users to hunt for information through a variety of sources.
- **Supporting personalized views.** Users and user communities can organize the information they access in ways that complement their work habits or interactions.
- **Providing self-service to users.** Employees, partners, and suppliers who have important information to share with the community can do so without specific technical skills or help from a technology expert.
- **Enabling single signon.** Users can log into the portal once per session and access all internal and external applications without logging into any of them.

Oracle9iAS Portal Features

Key features of Oracle9iAS Portal include:

- **An extensible framework** that integrates Web-based resources such as Web pages, applications, business intelligence reports, and syndicated content feeds, within standardized, reusable information components called portlets. Within a portlet, these resources are personalized and managed as a service of Oracle9iAS Portal. Companies can create their own portlets for their existing Web resources and can select additional portlets from the growing catalog of third-party portlet providers. The portal framework provides additional services including single signon, content classification, enterprise search, directory integration, and access control.
- **An easy-to-use, personalized interface** that provides an organized, consistent view of the business information, Web content, and applications needed by each user. Portal administrators and designers use a browser-based control panel to selectively grant access to applications and information by making portlets available only to specific users or user groups.

- **Self-service publishing** features allow authorized users to post and share any kind of document or Web content with other users anywhere in the world. Content contributors use controls for uploading documents, implementing version control, customizing page formatting and display, limiting access, and managing content without any requirement for technical expertise or HTML knowledge.
- **A scalable deployment architecture** that is easily configured for departmental, regional, and enterprise-wide deployment. The deployment model supports a variety of configurations, including single host and multi-tier, on a broad set of hardware platforms and operating systems.

Portal Page Creation, Management, and Customization

Oracle9iAS Portal incorporates a portal creation and deployment framework that defines Web information sources as information components, assembles these components within a portal page, and supports customization of the Web page to one or more user communities.

Each portal page is divided into either item regions or portlet regions. Item regions allow you to add text, images, and files to a portal page. Portlet regions provide an area where you can place one or more portlets.

A portlet is an HTML or XML area that summarizes, promotes, or provides basic access to an information resource. The information resources can take on many forms and can serve many purposes.

Items and portlets are the fundamental building blocks of an Oracle9iAS Portal page. Page owners create and maintain their own portal pages. Each portal page consists of content presented through one or more items and portlets and links that allow the client to navigate to another page or take some action.

Portal Content Publishing and Management

Oracle9iAS Portal provides an integrated set of features for self-service document publishing, file upload, page formatting, and access control. Collaborators and content publishers no longer need specific technical skills or a Webmaster to publish their content. Instead, they can use an item region on a portal page to publish their content and format content appearance using simple controls.

An item region includes built-in features for publishing, organizing, classifying, cross-referencing, and displaying content it manages. Key components that make up an item region include:

- **Items** that are the pieces of content themselves. Items are defined by the base content that makes up the item and one or more attributes that describes the item. Item publishers create items by completing a series of steps in a wizard. Additional features, such as item version control, check-in and check-out, expiration, and automatic indexing support collaborative document creation and ease content management tasks.
- **Styles** that define how items are displayed. Item region style properties govern the colors, font properties, size, background images, banners, and other graphical elements for the items, and navigation bars in an item region.

Item regions also include components that assist users in navigating or locating content of interest and allow for classifying content with categories and perspectives.

The category attribute classifies items. Each item must contain a single category attribute value. Categories answer the question "What is this item?"

The perspective attribute classifies items in an optional, one-to-many relationship. Each item may contain from none to many perspective attribute values. Perspectives answer the question "Who does this item interest?"

Content Searching

Oracle9iAS Portal searches can quickly and easily locate information managed within a portal page. Oracle9iAS Portal supports the following search methods:

- **Basic searches** that only match attributes of items in the current portal page against the search criteria.
- **Advanced searches** that allow the user to define or restrict the search by specific attributes. Users can specify that their searches
 - Return matches to any or all search terms.
 - Search within a specific portal page or across all portal pages.
 - Restrict searches to a particular category, perspective, item type, or attribute.
- **Oracle Ultra Search searches** that perform comprehensive searches against portal-managed information. These searches can also return matches based on near matches (terms that appear close together), soundex matches (terms that sound like the search term), and fuzzy matches (terms with similar spellings to the search term).

A search result portlet displays the matches of any search. You can add this portlet to any portal page and customize which attributes it displays.

Content Syndication

Content syndication is the aggregation, exchange, and distribution of information from content providers to syndicators to subscribers. The content providers provide the content, the syndicators send the content, and the subscribers use the content. A content subscriber acquires a content catalog of subscription offers from a content syndicator and selects the desired subscription offers. When subscribing for content, the subscriber can choose to search for new information or have it automatically provided to them either when it is updated, or after a specified time interval.

Oracle9iAS Syndication Server delivers any database, legacy file system, or Internet content to any Internet subscriber, and automatically provides any content updates using any standards across any network. This simplifies the process of syndication or automated content exchange. Oracle9iAS Syndication Server provides a comprehensive solution for content aggregation, syndication, and distribution by letting you make available any or all of your content.

Content syndicators can use Oracle9iAS Syndication Server with the following benefits:

- Aggregate content from databases, Web sites, enterprise applications, e-mail repositories, and legacy systems.
- Personalize the content to be delivered based on the profiles of their subscribers and deliver content updates based on the delivery policies associated with each subscription.
- Automate content delivery by sending content to subscribers when information relevant to them changes, or schedule a content delivery when establishing a subscription with subscribers, and deliver it to subscribers over multiple communication channels.
- Transform any content from any formatted source to a markup language suitable for the subscriber.

See Also: *Oracle Syndication Server User's and Administrator's Guide*

Portals and Wireless Devices

In addition to standard Web browsers, wireless clients can also access Oracle9iAS Portal pages. Working with Oracle9iAS Wireless, the portal automatically transforms the portal page structure to a format appropriate for the smaller screens of most wireless devices. Only portlets generating Oracle9iAS Wireless XML content display on the wireless device.

Oracle9iAS Portal developers also have access to a set of page design tools that help in creating portal pages that optimize the wireless experience. With the tools, developers can build a distinct portal structure for their wireless users. The wireless pages and portal pages can share portlet instances. This allows clients to reuse portlets on browser and wireless clients without reconfiguring each portlet.

Oracle9iAS Portal Architecture

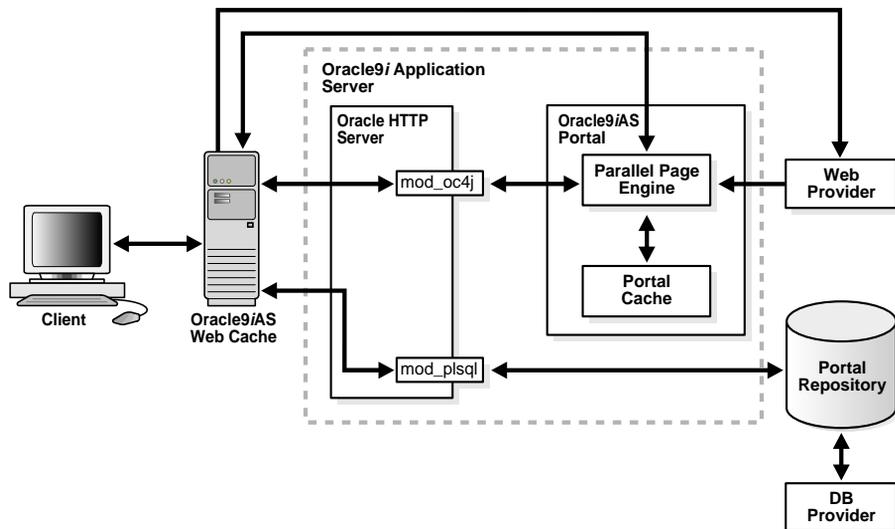
When a client requests an Oracle9iAS Portal page, many Oracle9iAS components service parts of the request. Requests have the following flow:

1. The client browser requests a portal page. Oracle9iAS Web Cache receives this request.
2. Oracle9iAS Web Cache forwards the request to the Oracle9iAS Portal Parallel Page Engine (PPE) through Oracle HTTP Server and mod_oc4j.
3. The PPE retrieves the portal page definition. The page definition contains information about the portlets on a page and their layout.
 - a. First, it checks if Oracle9iAS Web Cache has a valid, cached copy of the definition.
 - b. Next, it checks if the portal cache has a valid, cached copy.
 - c. Finally, if no cached copy of the definition exists, then the PPE generates a page definition from data in the portal repository. The portal repository is either in the [Oracle9iAS Metadata Repository](#) or in your customer database.
4. The PPE parses the page definition. If a fully cached copy of the page exists, then the page is returned to the client browser through Oracle9iAS Web Cache. If a fully cached copy of the page does not exist, the PPE builds the page from cached and non-cached data with the remaining steps.
5. For each portlet on the page, the PPE checks if a cached copy of the portlet content exists in the portal cache. Then the PPE forwards a request to the appropriate provider through Oracle9iAS Web Cache.

6. Each provider either validates the cached portlet or generates content for the portlet. Web providers return this directly to the PPE. Database (DB) providers return the results to the PPE through Oracle HTTP Server, `mod_plsql`, and Oracle9iAS Web Cache.
7. The PPE aggregates the content into a single page. This page is sent to Oracle9iAS Web Cache.
8. Oracle9iAS Web Cache returns the final page to the client browser.

Figure 3–2 illustrates this flow in terms of the Oracle9iAS Portal architecture.

Figure 3–2 Oracle9iAS Portal Request Flow



Application Access and Integration

Portal clients access Oracle9iAS Portal applications through portlets. Clients can select the portlets that appear on their page from a list of providers registered with Oracle9iAS Portal. In cases where the existing catalog of portlets is not sufficient, developers can use the Oracle9iAS Portal Developer Kit to create their own portlets.

Integrating with Portlet Providers

Applications and information sources, represented as portlets, communicate with the portal through a provider. Each portlet only has one provider, and a provider can have one or more portlets that expose an underlying application or information source.

All portlets from portlet providers make use of Oracle9iAS Single Sign-On. This ensures that only authorized users are able to subscribe to a particular portlet and that authorized users can access all registered portlets by logging into their main portal page.

Oracle9iAS Portlets

Some Oracle9iAS components act as portlet providers to Oracle9iAS Portal. This allows you to integrate information from various components into a single portal page.

Oracle9iAS Reports Services Oracle9iAS Portal includes a simple report building facility. However, as your reports become more complex, you can import the report into Oracle9iAS Reports Services reports. You can deploy any Oracle9iAS Reports Services report as a portlet.

See Also: ["Introduction to Oracle9iAS Personalization"](#) on page 6-9

Oracle9iAS Discoverer As a portlet provider, Oracle9iAS Discoverer offers worksheet portlets and list of workbooks portlets to Oracle9iAS Portal users. A worksheet portlet contains information from a single view in Discoverer. The portlet displays this information in either a table, a graph, or both. The list of workbooks portlet presents a list of worksheets.

See Also: ["Introduction to Oracle9iAS Discoverer"](#) on page 6-17

Oracle Internet File System The Oracle Internet File System portlet provides a summary of information about your Oracle 9iFS data. Portlet subscribers can see their quota, initiate a file search, list documents or folders, and upload documents. The portlet also links to the Oracle Internet File System Web interface.

See Also: ["Oracle Internet File System"](#) on page 7-10

Custom Portlets

The Oracle9iAS Portal Developer Kit (PDK) allows developers to either reuse existing applications as portlets or create new portlets. Developers can write portlets using familiar languages and technologies. These technologies can be:

- Java applications
- Oracle PL/SQL packages
- Web pages built with any standard technology like Active Server Pages (ASPs), or Perl
- Web services

See Also: <http://portalstudio.oracle.com> for more information on the Oracle9iAS Portal Developer Kit

Partner Portlets

In addition to the list of Oracle9i Application Server components, a growing community of independent software vendors (ISVs) and Internet content providers are creating standard, supported portlets that access their applications and services. For these partners, customers can access the partner's application or service through one or more pre-integrated portlets.

The current catalog of portlets includes the following services:

- Business intelligence and reporting
- Collaboration
- Document, content, and knowledge management
- e-business applications (including customer relationship management and enterprise resource management)
- News and information sources
- Portal tools
- Internet searches

See Also: <http://portalstudio.oracle.com> for a complete list of Partner portlets

Portal Security

This section discusses the following security topics:

- [User Authentication in Oracle9iAS Portal](#)
- [Access Control in Oracle9iAS Portal](#)

User Authentication in Oracle9iAS Portal

Oracle9iAS Single Sign-On provides a single point of validation for portal user credentials and governs user access to intranet resources based on employee profiles. When a user logs into a portal page, the single signon server validates the user name and password against user and group profiles stored in Oracle Internet Directory.

Oracle9iAS Single Sign-On also supports external, partner applications. For these applications, the single signon server logs in to the application automatically for the user.

See Also: [Chapter 8, "System Management"](#)

Access Control in Oracle9iAS Portal

Most portal elements have an access control list (ACL). This list controls which users and groups may access the element, and to what extent. For example, if you wanted all of the users in a group to be able to see the items on a portal page, in the page's ACL you would grant that group View privileges.

Besides ACLs, you can also use global privileges to grant access to all objects of a given type in Oracle9iAS Portal. For example, granting the Create privilege for All Pages to a group enables all members of that group to create pages.

Wireless Development and Deployment

This chapter provides an overview of Oracle9iAS Wireless features and benefits. The topics include:

- [Introduction to Oracle9iAS Wireless](#)
- [Oracle9iAS Wireless Architecture](#)
- [Oracle9iAS Wireless Features](#)

Introduction to Oracle9iAS Wireless

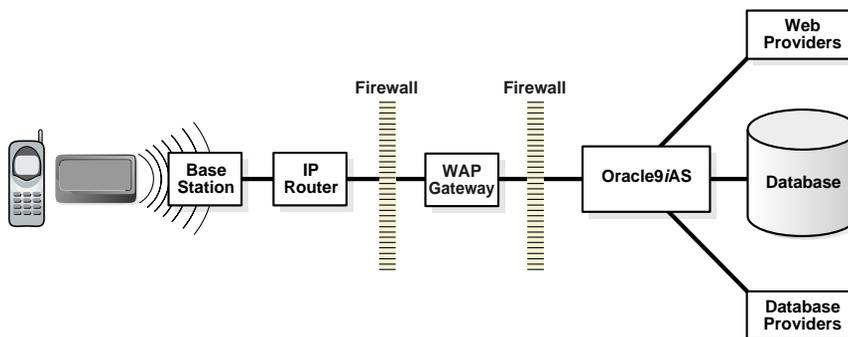
After you build a J2EE application and publish it to a Web service, you can link it to any system in the back end. Then, you can aggregate this application into a portal. Now, you want to get wireless access into it. It is basically the same kind of content. Oracle9iAS Wireless lets you rapidly extend access to any existing application to any mobile device including Web-enabled phones, personal digital assistants, pagers, and even ordinary telephones through voice recognition.

Oracle9iAS Wireless is a portal service for delivering information and applications to mobile devices.

What is Oracle9iAS Wireless?

Oracle9iAS Wireless makes Web and database applications, such as e-mail, news, and directory services, accessible to mobile device users without having to rewrite content for every target platform. It converts any Internet content to XML and transforms the XML to any markup language supported by any device such as HTML, WML, HDML, VoiceXML, VoxML, and SMS.

Figure 4–1 Oracle9iAS Wireless



Oracle9iAS Wireless Architecture

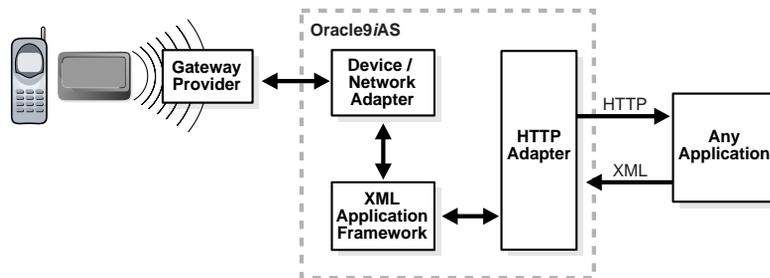
When users request wireless service, the following steps occur:

1. The wireless device connects to the Gateway Provider.
2. The provider provides subscriber, device identification, and user preferences to the Device/Network Adapter.

3. The Device/Network Adapter invokes the XML Application Framework and HTTP Adapter.
4. The HTTP Adapter forwards the request to the application.
5. The HTTP Adapter retrieves the XML result from the application.
6. The result is forwarded to the client.

Figure 4-2 illustrates this flow in terms of the Oracle9iAS Wireless architecture.

Figure 4-2 Oracle9iAS Wireless Architecture



Oracle9iAS Wireless HTTP Adapter

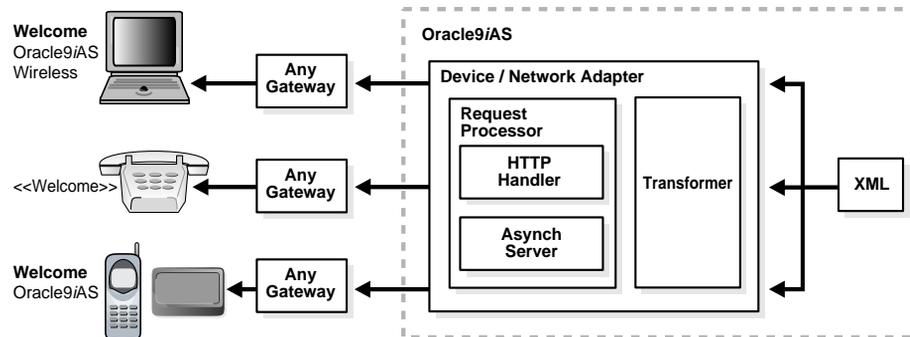
Oracle9iAS Wireless adapters pass application content to the core for processing. The HTTP Adapter retrieves content from a wide variety of sources, including existing Web sites, databases, existing applications and legacy systems.

The HTTP Adapter can securely access any content from any source that outputs XML. To render an application to any device, a developer points the HTTP Adapter to the application with a URL. The Oracle9iAS Wireless core then processes the XML object and delivers the result of the query to the wireless device, such as a WAP device.

Device/Network Adapter

Oracle9iAS Wireless device transformers convert the XML received from the HTTP adapter to the markup language appropriate for the requesting wireless device.

Figure 4-3 Oracle9iAS Wireless - Device/Network Adapter



The Device/Network Adapter automatically transforms and optimizes the application content for any wireless device and network. It supports the following mobile technologies:

- 2-way pagers for asynchronous services (SMTP/SMS)
- WAP devices
- Voice for access through regular phone lines
- PDA devices

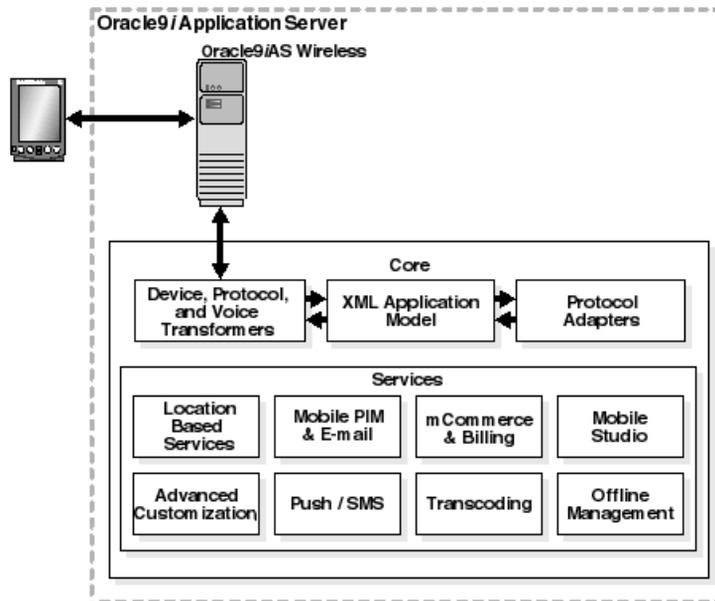
Oracle9iAS Wireless provides two types of transformers:

- Generic service transformers - For languages such as WML. Generic service transformers convert the Oracle9iAS Wireless-compliant XML to a generic WML format that works on any WAP-compliant wireless phone.
- Device-specific transformers - Optimized for a specific device. For example, instead of using the generic WML transformer, you can use a device-specific transformer that exploits the device characteristics of a specific phone.

Oracle9iAS Wireless Features

Oracle9iAS Wireless contains two parts: Core and Services.

Figure 4–4 Oracle9iAS Wireless Core and Services



Core

Oracle9iAS Wireless includes a core component that manages the Oracle9iAS Wireless repository and service requests:

- Device, Network Adaptation - Allows you to write your application once for any protocol, any device, or any network.
- XML Application Model - Provides an advanced application model to enable mobile access.
- Protocol Adapters - Provides open HTTP Protocol adapters to mobile enable any existing application.

Services

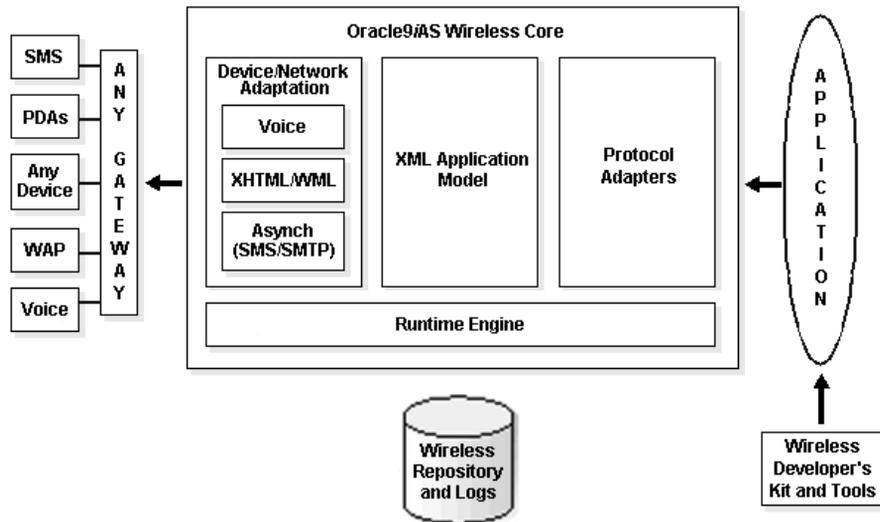
Oracle9iAS Wireless includes services that enhance applications and simplify mobile application development:

- Location Based Services - Allows you to create location-aware applications and personalize them with geographic coding, routing, mapping, and positioning.
- Mobile Personal Information Management (PIM) and E-mail - Integrates corporate e-mail, directory, address book, calendaring, and instant messaging applications into mobile enterprise portal.
- mCommerce and Billing - Provides a set of modules for on-line payment mechanisms to complete transactions.
- OracleMobile Studio - Provides an on-line wireless development tool for quickly building, testing, and deploying wireless applications.
- Advanced Customization - Allows you to customize your Personalization Portal pages in several different ways to alter the appearance of logos, banners, and icons.
- Push/Short Message Service - Allows content sources to send information to mobile users and alert them to important information or events.
- Transcoding - Allows you to mobile-enable existing HTML and WML Web content.
- Off-line Management - Allows you to create applications that run on mobile devices while off-line by using Oracle9i Lite.

Deploy Applications for Any Device, Protocol, or Network

Oracle9iAS Wireless allows you to write your applications once for any protocol, device, or network. The Oracle9iAS Wireless simplifies the design of your applications and decreases the software life-cycle time.

Figure 4–5 Deploy applications on any device, protocol, or network



Oracle9iAS Wireless has XML, HTTP, and J2EE adapters to make existing applications portable to mobile devices. You can receive any information via a standard protocol. You can also apply transforms using XML, and then depending upon what device you are going to or what network you are going over, Oracle9iAS Wireless adapts the content to go out. Regardless of data presentation or storage format, Oracle9iAS Wireless adapts and transforms complicated content into the markup language of any wireless device.

Oracle9iAS Wireless content adapters can adapt information to the wireless Internet from a wide variety of sources, including existing Web sites, databases, and legacy systems. The goal of adapters is to create MobileXML, an Oracle XML document type definition (DTD). MobileXML is the device-independent XML for delivering content through Oracle9iAS Wireless. MobileXML elements provide the user interface elements necessary to present content to mobile devices. You can create MobileXML through standard text editors or any popular development tool that is capable of outputting XML. The specific markup language supported by that device to which the MobileXML is transformed determines the specific appearance of each MobileXML element on a device.

See Also: *Oracle9iAS Wireless Developer's Guide*

Location-Based Services

With Oracle9iAS Wireless, you can create location-aware applications and extend them with geographic coding, routing, mapping, and positioning functions.

Oracle9iAS Wireless enables you to assign a location to a service, making the service location-based. It supports two types of location-based services:

- Location-based services that are visible to users only at certain locations.
- Services that provide content specific to the current location of the Oracle9iAS Wireless user. For example, an Oracle9iAS Wireless user located in San Francisco sees content related to San Francisco, such as listings of restaurants or businesses in San Francisco. Oracle9iAS Wireless supports content-specific location-based services through customized adapter implementation and location-aware runtime support.

In addition, Oracle9iAS Wireless renders mapping services from third-party providers without you having to write custom interfaces for each service.

Mobile Personal Information Management (PIM) and E-mail Support

Oracle9iAS Wireless includes application modules that add the functionality necessary to successfully deploy applications and portals. The modules are built on open standards and focus on reusability and flexibility for seamless integration. These modules are reusable applications that you can link to existing applications.

Oracle9iAS Wireless provides several modules that are ready for deployment, including Personal Information Management (PIM) tools such as calendar, address book, fax, and mail. Some of the modules are:

- mWallet supports mobile commerce transactions and tracking.
- Mobile E-mail supports accessing IMAP and POP e-mail.
- Mobile Directory supports access to LDAP directories.
- Mobile Calendar provides schedule and appointment management.
- Instant Messaging supports exchanging instant messages from mobile devices.

Oracle9iAS Wireless PIM Service allows you to integrate corporate e-mail, directory, address book, calendaring, and instant messaging applications into your mobile enterprise portals. Each of these applications is built as a module that can be called either directly by mobile users from their devices, or by other applications. In addition, they are fully integrated. You can access features such as directory or address book-based recipient selection while composing e-mail messages.

You can leverage from Oracle9iAS Wireless PIM Service modules to:

- Add communication features to these services.
- Retrieve corporate directory information.
- Add and manage appointments on behalf of users, for instance when they make travel or dining reservations.

The Mobile E-mail Client provides access from any mobile devices to any IMAP or POP3 server. This includes such servers as Microsoft Exchange and Lotus Domino.

The Mobile Directory Client connects to any LDAP directory server. Finally, the Mobile Calendar Client integrates natively with Exchange and Lotus Servers, and through published interfaces enables easy customization to support any Calendar Server.

See Also: *Oracle9iAS Wireless Getting Started and System Guide*

m-Commerce and Billing Support

Oracle m-Commerce Service is a set of Oracle9iAS Wireless modules that:

- Stores user profiles securely.
- Supplies information authorized by users of third-party applications.
- Interfaces with on-line payment mechanisms to complete transactions.
- Translates existing WML applications into MobileXML.
- Uses FormFiller to map forms and spare users from typing information into mobile devices.

m-Commerce APIs

You can build your m-Commerce application using Oracle9iAS Wireless. You can incorporate any m-Commerce component by adding URL links to the modules complying with their APIs. If you already have an m-Commerce application developed in WML, you can run it through the XLtor module by calling its API, and providing your application's URL. This action will add links from your application to all m-Commerce modules.

Mobile Wallet

Mobile Wallet module (m-Wallet) provides a convenient single-click commerce payment mechanism. It is a server side, encrypted entity that contains a payment

instrument, identification and address information for registered users. m-Wallet enables users to store all the information required to fill out commerce-related forms from any application. That information is used to complete transactions, and through APIs built and maintained by authorized third-party service providers, can be made available to authorized partners and e-merchants.

The power of the m-Wallet is that it securely stores this information for users, providing them an easy, secure shopping experience, and freeing them from repeatedly entering information.

FormFiller

FormFiller module dynamically saves and maintains mappings among e-commerce applications' form fields and wallet elements to enable one-click shopping. Along with the XLtor, FormFiller enables these legacy applications to be integrated into mobile portals, accessed from any mobile device, and participate in wallet-enhanced commerce transactions.

Payment Processing

Payment Processing drivers provide seamless integration with payment mechanisms such as Oracle's CRM iPayment, enabling the processing of credit card and bank account transactions via direct connections to financial networks.

Extending m-Commerce Drivers

The extensible architecture of the m-Commerce Service enables the development of drivers to integrate m-Commerce services with third-party applications.

Studio for Building, Testing, and Deploying Mobile Applications

OracleMobile Online Studio is an on-line developer portal for quickly building, testing, and deploying wireless applications. It lets you develop a mobile application that is immediately accessible from all devices. All you need is a standard application server that outputs XML to leverage shared online Web services to reach any wireless Web browser, any messaging device through SMS, E-Mail, WAP push, and any phone using voice.

OracleMobile Online Studio's build, test, and deploy model presents a hosted approach to developing dynamic content. You do not need to download any software or tools to start using it. It also provides you access to reusable modules and examples.

Advanced Customization

You can customize your Personalization Portal pages in several different ways. You can easily alter the appearance of logos, banners, and icons. Alternatively, you may want to create your own JSP to achieve the desired look and feel.

Users can customize or configure the following:

- Services
- Folders
- Alerts (alert users when preset conditions are met)
- Profile information
- Bookmarks
- Alert addresses
- Location marks

Server Push and SMS

You can use Oracle9iAS Wireless to build notification services that alert users to important information or events. Oracle9iAS Wireless supports these notifications, or push applications, as e-mail and Short Message Service (SMS). Users can also make their own notifications without having to access the system local notifications.

In addition, you can also build location-sensitive mobile notification services that send a mobile user personalized promotional offers from stores or restaurants within the vicinity of the mobile user.

Transcoding

The majority of applications available on the Web render content in format specific to certain types of clients or devices. Transcoding services reformat applications developed for a particular device or markup language to formats suitable for any Web-enabled device.

Transcoding services in Oracle9iAS Wireless supports a Web Content Adaptation service and a WML Translator service.

- The Web Content Adaptation service allows you to quickly extend your existing published and legacy Web application to wireless Web-enabled devices. Oracle9iAS Wireless services can connect any remote Web resource, like HTML or XML document, and acquire content for reformatting. The content acquired

is adapted and mapped to data elements in wireless XML format and rendered to the mobile devices.

- The WML Translator service allows interpretation of content authored in WML and translates the content for access from all Web enabled devices

See Also: *Oracle9iAS Wireless Getting Started and System Guide*

Off-Line Management

You can off-line enable your Oracle9iAS Wireless application with Oracle9i Lite, which provides a lightweight database for off-line mobile computing. Oracle9i Lite is a platform to develop, deploy, and manage e-business applications that run locally on mobile devices. It offers the following:

- Rapid development of Java or native off-line mobile applications for Palm PDAs, pocket PCs, smartphones, and standard laptops.
- Centralized application deployment, provisioning, and management for all mobile devices.
- Highly scalable, bi-directional data synchronization between mobile devices and Oracle9i Lite over any connection such as Internet and wireless.
- Same application data for on-line and off-line application.

Caching for Performance

This chapter provides an overview of Oracle9i Application Server caching features and benefits. The topics include:

- [About Caching](#)
- [Oracle9iAS Web Cache Overview](#)
- [Oracle9iAS Web Cache Architecture](#)
- [Oracle9iAS Web Cache Features](#)
- [Oracle9iAS Web Cache Security](#)

About Caching

Caching is one of the key technologies that promises to alleviate the computational and economic burdens faced by today's overstrained e-business infrastructures. Nearly all applications benefit from having Web content cached on hosts between the consumers searching for content and the content source itself. While there are numerous hardware and software caching solutions for caching static content, most are not well suited for the kind of dynamic, personalized content required by the e-business model.

Oracle9iAS Web Cache offers a combination of server acceleration and server load balancing. Unlike legacy cache servers which only handle static data, Oracle9iAS Web Cache accelerates the delivery for both static and dynamically generated Web content, thereby improving response time for feature-rich pages.

As the first and only caching solution on the market to support Edge Side Includes (ESI) for performing page assembly in edge servers, Oracle9iAS Web Cache delivers rich, personalized content from both the edge of the data center to the edge of the Internet. Deployed before a farm of application servers and/or globally at the network edge, Oracle9iAS Web Cache provides load balancing, failover, and patent-pending surge protection features for application servers.

These features ensure incomparable site performance and solid up-time while reducing the cost of doing business online. With Oracle9iAS Web Cache, e-businesses can now serve compelling content faster, to more customers, using fewer computing resources than ever before.

Oracle9iAS Web Cache Overview

Oracle9iAS Web Cache is a powerful state-of-the-art server acceleration solution, offering throughput rates of several thousand requests per second. Oracle9iAS Web Cache offers intelligent caching, page assembly and compression features which distinguish it from any other Web caching solution on the market. Unlike legacy proxy servers which cache only static objects, Oracle9iAS Web Cache accelerates the delivery of both static and dynamic Web content.

Oracle9iAS Web Cache provides the following benefits for Web sites:

Performance

Running on inexpensive hardware, Oracle9iAS Web Cache can increase the throughput of a Web site by several orders of magnitude. In addition, Oracle9iAS Web Cache significantly reduces response time to browser requests by storing documents in memory and by serving compressed versions of documents to browsers that support the GZIP encoding method.

Scalability

In addition to unparalleled throughput, Oracle9iAS Web Cache can sustain thousands of concurrent browser connections, meaning that visitors to a site see fewer application server errors, even during periods of peak load.

High Availability

Oracle9iAS Web Cache supports content-aware load balancing and failover detection. These features ensure that cache misses are directed to the most available, highest-performing application server in the cluster. Moreover, a patent-pending capacity heuristic guarantees performance and provides surge protection when application server load increases.

Cost Savings

Better performance, scalability, and availability translates into cost savings for Web site operators. Because fewer application servers are required to meet the challenges posed by traffic spikes and denial of service attacks, Oracle9iAS Web Cache offers a simple and inexpensive means of reducing a Web site's cost for each request.

Reduce Network Traffic

Most requests are resolved by Oracle9iAS Web Cache, reducing traffic to the application servers. The cache also reduces traffic to backend databases located on computers other than the application server.

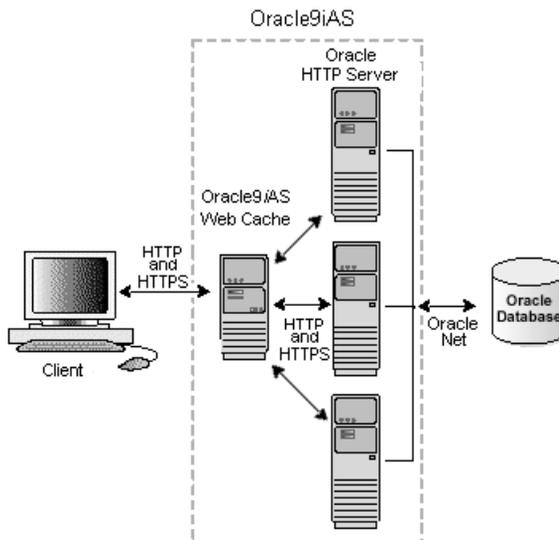
See Also: *Oracle9iAS Web Cache Administration and Deployment Guide*

Oracle9iAS Web Cache Architecture

Oracle9iAS Web Cache sits in front of application servers, caching their content in memory, and providing that content to Web browsers that request it. [Figure 5-1](#) shows the architecture of Oracle9iAS Web Cache.

1. When Web browsers access the Web site, they send HTTP(S) protocol requests to Oracle9iAS Web Cache.
2. Oracle9iAS Web Cache, in turn, acts as a virtual server to the application servers.
3. If the request content has changed, Oracle9iAS Web Cache retrieves the new content from the application servers.
4. The application servers may retrieve their content from an Oracle database.

Figure 5-1 Caching Architecture



Oracle9iAS Web Cache Features

Oracle9iAS Web Cache is a powerful solution for accelerating online catalogs, auctions, Internet and intranet portals, business-to-business exchanges, and Customer Relationship Management applications. Key features include:

- [Full-Page Static and Dynamic Content Caching](#)
- [Partial-Page Caching and Personalized Content Assembly](#)
- [Invalidation for Cache Consistency Management](#)
- [Performance Assurance Heuristics](#)
- [Automatic Content Compression](#)
- [Web Server Load Balancing and Failover](#)
- [Cache Clustering](#)
- [Flexible Deployment Options](#)

Full-Page Static and Dynamic Content Caching

Oracle9iAS Web Cache uses cacheability rules to store documents. These rules fall into two categories:

- Rules for static content, such as GIF, JPEG, or static HTML files
- Rules for dynamically generated content created using technologies like JSP, ASP, PSP, servlets, and CGI. Support for these technologies enables Oracle9iAS Web Cache to recognize rules for the following:
 - Multiple-version documents for the same URL, that is, the same URL with slightly different content
 - Session-aware rules for pages containing session information
 - Personalization rules for pages containing personalized greetings, such as “Welcome *Name*”, and session-encoded URLs.

Partial-Page Caching and Personalized Content Assembly

Oracle9iAS Web Cache provides dynamic assembly of pages with both cacheable and non-cacheable page fragments. It provides for assembly by enabling pages to be broken down into fragments of differing cacheability profiles. These fragments are each maintained as separate elements in the application server or content delivery network and assembled into HTML pages as appropriate when requested by end users.

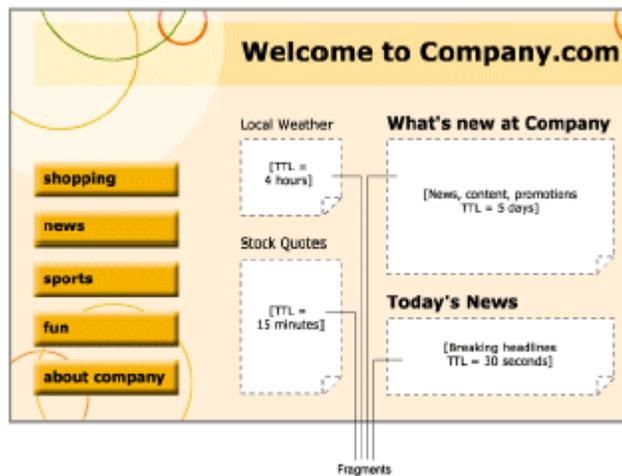
By enabling dynamic assembly of pages on Oracle9iAS Web Cache rather than on the application servers, you can choose to cache some of the fragments of assembled pages. This means that much more HTML content can be cached, then assembled and delivered by Oracle9iAS Web Cache when requested. Furthermore, page assembly can be conditional, based on information provided in HTTP request headers or end-user cookies.

Page Assembly Components

The basic structure a content provider uses to create dynamic content is a template page containing HTML fragments. The template consists of common elements, such as a logo, navigation bars, framework, and other “look and feel” elements of the page. The HTML fragments represent dynamic subsections of the page. The template page is associated with the URL that end users request. To include the HTML fragments, the template page is configured with Edge Side Includes (ESI) markup tags that tell Oracle9iAS Web Cache to fetch and include the HTML fragments. The fragments themselves are HTML files containing discrete text or other objects.

[Figure 5–2](#) shows an example of the template mentioned above.

Figure 5–2 Template Page



Edge Side Includes (ESI)

ESI is a markup language to enable partial page caching for HTML fragments. ESI can be used with HTML, XML, and any Web publishing technology. ESI is an open standard. For more information, see <http://www.esi.org>.

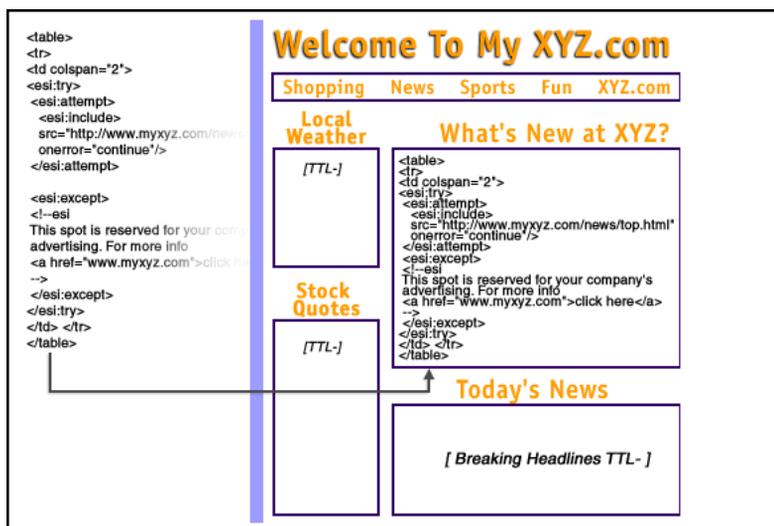
See Also: *Oracle9iAS Web Cache Administration and Deployment Guide*

Edge Side Includes for Java (JESI)

JESI is a specification and custom JSP tag library that developers can use to automatically generate ESI code using JSP syntax. Even though JSP developers can always use ESI, JESI provides an even easier way for JSP developers to express the modularity of pages and the cacheability of those modules, without requiring developers to learn a new syntax.

Figure 5–3 shows an example of the ESI and JESI.

Figure 5–3 ESI and JESI



Invalidation for Cache Consistency Management

Oracle9iAS Web Cache supports invalidation as a mechanism to keep its cache consistent with the content on the application servers, origin databases, or other dynamically generated means. Administrators can invalidate cache content in the following two ways:

- Send an invalidation message to the computer running Oracle9iAS Web Cache. When documents are invalidated and a browser requests them, Oracle9iAS Web Cache refreshes them with new content from the application server.
- Assign an expiration time limit to the documents. When a document expires, Oracle9iAS Web Cache treats it like an invalid document, that is, if requested by a browser, it refreshes it with an updated content from the application server.

Performance Assurance Heuristics

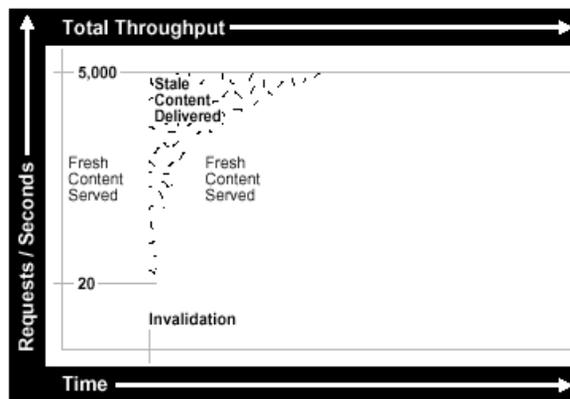
When a large number of documents have been invalidated, the retrieval of a new document can result in overburdened application servers. To handle performance issues while maintaining cache consistency, Oracle9iAS Web Cache uses built-in performance assurance heuristics that enable it to assign a queue order to documents. These heuristics determine which documents can be served stale and which documents must be refreshed immediately. Documents with a higher

priority are refreshed first whereas documents with a lower priority are refreshed at a later time.

The queue order of documents is based on the popularity of documents and the validity of documents assigned during invalidation. If the current load and capacity of the application server is not exceeded, the most popular and least valid documents are refreshed first.

Figure 5-6 illustrates how performance assurance heuristics are used during widespread invalidation.

Figure 5-4 Performance Assurance Heuristics Graph



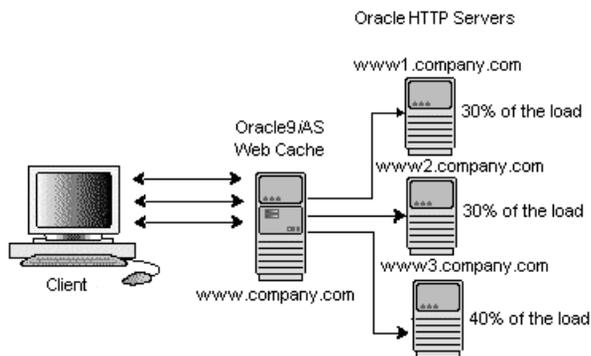
Automatic Content Compression

You can select to have Oracle9iAS Web Cache compress both cacheable and non-cacheable documents upon insertion into the cache for browsers. Because compressed documents are smaller in size, they are delivered faster to browsers with fewer round-trips, reducing overall latency. On average, Oracle9iAS Web Cache is able to compress text files by a factor for 4. For example, 300 KB files are compressed down to 75 KB.

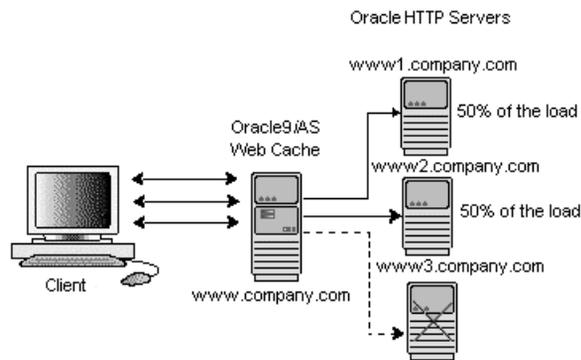
Web Server Load Balancing and Failover

Most Web sites are served by multiple application servers running in multiple computers that share the load of HTTP(S) requests. This feature enables Web sites to be built with a collection of servers for better scalability and reliability. Oracle9iAS Web Cache is designed to manage requests for up to 100 application servers. All requests that Oracle9iAS Web Cache cannot serve are passed to the application servers. Oracle9iAS Web Cache sends requests to the application server with the most available load using its *load balancing* feature. [Figure 5-5](#) shows the Oracle9iAS Web Cache load balancing architecture.

Figure 5-5 Oracle9iAS Web Cache Load Balancing Architecture

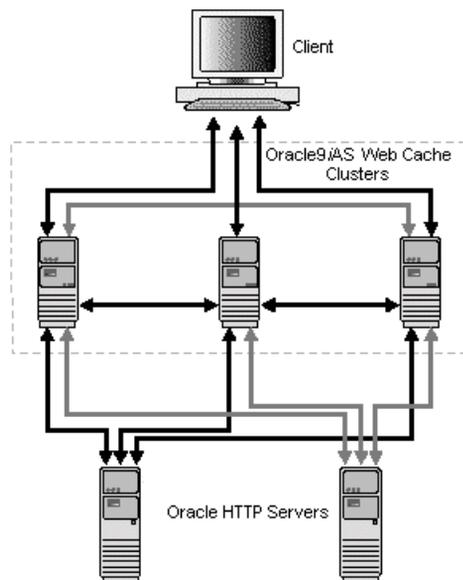


When an application server becomes unavailable because it is taken down for reconfiguration or there is a network or hardware failure, Oracle9iAS Web Cache automatically performs backend *failover*. Here, Oracle9iAS Web Cache distributes the load over the remaining application servers and polls the failed application server for its current up/down status until it is back online. Existing requests to the failed application server result in errors. However, new requests are directed to the other application servers. When the failed server returns to operation, Oracle9iAS Web Cache includes it in the load distribution. [Figure 5-6](#) shows the Oracle9iAS Web Cache failover architecture.

Figure 5–6 Oracle9iAS Web Cache Failover Architecture

Cache Clustering

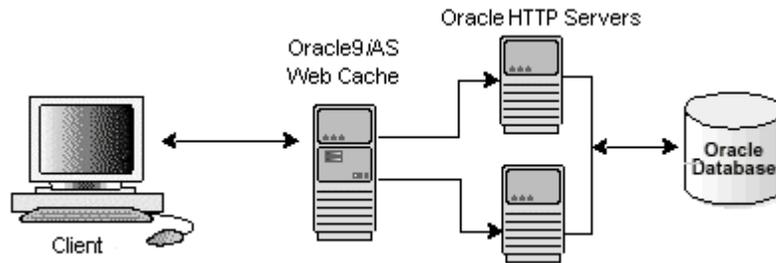
You can configure multiple instances of Oracle9iAS Web Cache to run as independent caches, with no interaction with one another. However, to increase the availability and scalability of your Web cache, you can configure multiple instances of Oracle9iAS Web Cache to run as members of a cache cluster. A cache cluster is a loosely coupled collection of cooperating Web cache instances working together to provide a single logical cache. [Figure 5–7](#) shows an example of Oracle9iAS Web Cache cluster.

Figure 5-7 Oracle9iAS Web Cache Cluster

See Also: *Oracle9i Application Server Performance Guide*

Flexible Deployment Options

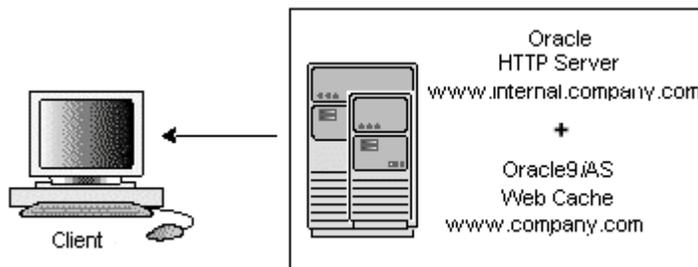
In the simplest of deployment scenarios, Oracle9iAS Web Cache is positioned in front of one or more Web servers to cache content generated by those servers. Oracle9iAS Web Cache then delivers that content to Web browsers. When Web browsers access the Web site, they send HTTP requests to Oracle9iAS Web Cache which acts as a virtual server from the Web site, masking the existence of the application server farm and the database. If the requested content has changed, Oracle9iAS Web Cache retrieves the new content from the application servers according to the relative load on each server. [Figure 5-8](#) shows an example of a simple deployment.

Figure 5–8 Simple Deployment Scenario

Oracle9iAS Web Cache can be deployed on the same host as the origin application server (co-located) or on a dedicated node of its own (dedicated).

In a cluster scenario, a network load balancer distributes requests across the cache instance running on each node on the server farm. Each Oracle9iAS Web Cache instance is typically configured with the host name of the application server(s) running on its own host, and interprocess communication (IPC) is used to pass requests between the cache and the application server(s). Because Oracle9iAS Web Cache consumes memory, co-location is only viable if the cache and the application server(s) do not contend for resources.

[Figure 5–9](#) shows an architecture of Oracle9iAS Web Cache on the same host as the Application server.

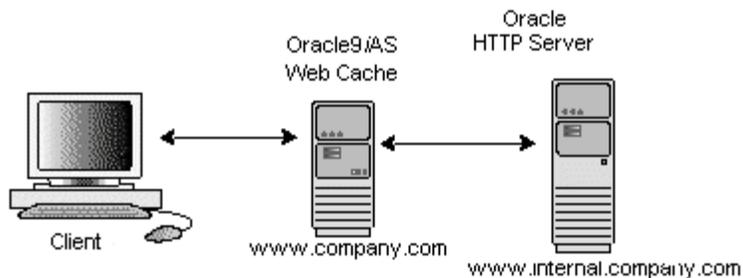
Figure 5–9 Oracle9iAS Web Cache on Same Host as Application Server

A dedicated deployment is often preferable to co-located deployment. In a dedicated scenario, there is no risk of resource contention with other server processes. Oracle9iAS Web Cache also performs well on commodity hardware, so a dedicated deployment need not be a costly one in terms of hardware expenditure.

For very high-volume Web sites, and to avoid a single point of failure, two or more hosts running Oracle9iAS Web Cache may be deployed behind a third-party network load balancing device.

[Figure 5–10](#) shows an architecture of Oracle9iAS Web Cache on a different host as the application server.

Figure 5–10 Oracle9iAS Web Cache on Different Host from Application Server



See Also: *Oracle9iAS Web Cache Administration and Deployment Guide*

Oracle9iAS Web Cache Security

Oracle9iAS Web Cache provides the following security-related features:

- [Restricted Administration](#)
- [Secure Sockets Layer \(SSL\) Support](#)

Restricted Administration

Oracle9iAS Web Cache restricts administration with the following features:

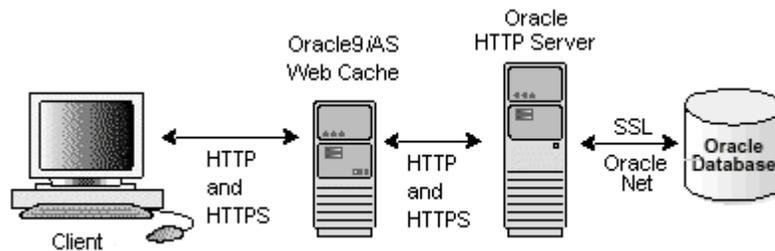
- Password authentication for administration and invalidation operations
- Control over ports from which administration and invalidation operations can be requested
- IP and subnet administration restrictions

Secure Sockets Layer (SSL) Support

The Secure Sockets Layer (SSL) protocol, developed by Netscape Corporation, is an industry standard for network transport layer security. SSL provides authentication, encryption, and data integrity, in a public-key infrastructure (PKI). By supporting SSL, Oracle9iAS Web Cache is able to cache pages from HTTPS requests.

Figure 5–11 shows how you can configure Oracle9iAS Web Cache to receive HTTPS requests and send HTTPS requests to the application server.

Figure 5–11 SSL for Server Connection



Note that HTTPS traffic can be process intensive. If Oracle9iAS Web Cache needs to have traffic travel over the open Internet, then configure Oracle9iAS Web Cache to send HTTPS requests to the application server. If traffic only travels through a LAN in a data center, then the traffic can be sent with HTTP so as to reduce the load on the application servers.

SSL interacts with the following entities:

- [Certificate Authority](#)
- [Certificate](#)
- [Wallet](#)

Certificate Authority

A certification authority (CA) is a trusted third party that certifies the identity of third parties and other entities, such as users, databases, administrators, clients, and servers. The CA verifies the party identity and grants a certificate, signing it with its private key. The Oracle9iAS Web Cache certificate must be signed by a CA.

Certificate

A certificate is created when a party's public key is signed by a trusted CA. A certificate ensures that a party's identification information is correct, and that the public key actually belongs to that party. A certificate contains the party's name, public key, and an expiration date, as well as a serial number and certificate chain information. It can also contain information about the privileges associated with the certificate. When a network entity receives a certificate, it verifies that it is a trusted certificate, one issued and signed by a trusted CA.

Wallet

A wallet is a transparent database used to manage authentication data such as keys, certificates, and trusted certificates needed by SSL. A wallet has an X.509 version 3 certificate, private key, and list of trusted certificates.

Security administrators use the Oracle Wallet Manager to manage security credentials on the Oracle9iAS Web Cache server. Wallet owners use it to manage security credentials on clients. Specifically, Oracle Wallet Manager is used to do the following:

- Generate a public-private key pair and create a certificate request for submission to a CA
- Install a certificate for the identity
- Configure trusted certificates for the identity

To support HTTPS support between browsers and Oracle9iAS Web Cache, configure a wallet on the Oracle9iAS Web Cache server for each supported site. To support HTTPS support between Oracle9iAS Web Cache and the application servers, configure a wallet on the application server.

See Also: *Oracle9i Application Server Security Guide*

Business Intelligence

This chapter provides an overview of the Oracle9iAS Business Intelligence features and benefits. The topics include:

- [Introduction to Oracle9iAS Business Intelligence](#)
- [Introduction to Oracle9iAS Clickstream Intelligence](#)
- [Introduction to Oracle9iAS Personalization](#)
- [Introduction to Oracle9iAS Reports Services](#)
- [Introduction to Oracle9iAS Discoverer](#)
- [Integrate Oracle9iAS Business Intelligence](#)

Introduction to Oracle9iAS Business Intelligence

In an increasingly competitive business world, business intelligence empowers decision makers to act on information to keep their products competitive. Complete business intelligence about the operation of an enterprise and its customers can be used to assess opportunities as well as potential risks.

Automated business processes and electronic interactions with customers and suppliers have yielded an abundance of valuable business intelligence data. This data tracks the fundamentals and operations of the business, and presents an opportunity to dramatically improve operational efficiency, reduce costs, and increase overall productivity.

With an e-business, traditional information sources such as Enterprise Resource Planning (ERP) systems and CRM (Customer Relationship Management) systems can be correlated with information that comes directly from the usage patterns of the companies' Web sites. This enables companies to more closely analyze consolidated business data, secure existing business relationships, forge new revenue generating opportunities, and assist with the formulation of strategies to improve operating efficiency.

In order for an e-business to successfully gather business intelligence information about Web site usage and provide meaningful data analysis, it must be able to:

- Collect data generated from Web site activities and transform the data into business intelligence information.
- Integrate Web traffic data with business transaction data and then condense the data into a meaningful report.
- Enable business users at all levels of the organization to perform ad-hoc queries of stored business intelligence data.
- Provide a deployment platform for creation of multi-source, end-user reports that are available online or on paper.
- Provide real-time analysis and recommendations for users of the Web site.

Oracle9iAS Business Intelligence provides a complete, integrated solution that addresses all business intelligence requirements. Oracle9iAS Business Intelligence consists of the following components:

- **Oracle9iAS Clickstream Intelligence:** Oracle9iAS Clickstream Intelligence collects data from Web site activities, transforms data into business information, integrates Web traffic data with business transaction data, and condenses data into meaningful reports.

- **Oracle9iAS Reports Services:** Oracle9iAS Reports Services provides a powerful deployment platform for publishing high-quality, dynamically generated end-user reports in a scalable, secure environment. You can build and publish sophisticated reports from any source, with unlimited data formatting, and deploy them seamlessly on Oracle9i Application Server.
- **Oracle9iAS Personalization:** Oracle9iAS Personalization provides real-time analysis and recommendations for e-business sales channels, such as Web stores, application hosting environments and business call centers. Oracle9iAS Personalization provides an integrated real-time recommendation engine built on predictive models that are automatically created within the Oracle9i Database Server using embedded data mining technology. Oracle9iAS Personalization delivers powerful, scalable real-time personalization for customer areas of interest. This enables businesses to deliver tailored, one-to-one customer experiences for Web site users.
- **Oracle9iAS Discoverer:** Oracle9iAS Discoverer enables business users at all levels of the organization to gain immediate access to information from data warehouses, clickstream data, and online transaction processing (OLTP) systems.

Introduction to Oracle9iAS Clickstream Intelligence

When you surf the Internet, a record of all your mouse clicks and connections is maintained. This record may include the Web sites you have accessed, which newsgroups you participate in, your email address, and additional information about your Web activity. This type of information is referred to as clickstream data. Web servers generate and record clickstream data in the form of Web logs. Web log files can be analyzed to extract business intelligence regarding Web site usage and user behavior.

What is Oracle9iAS Clickstream Intelligence?

Oracle9iAS Clickstream Intelligence provides business intelligence concerning Web traffic related data to increase Web site effectiveness and improve the customers' experience. Oracle9iAS Clickstream Intelligence enables you to:

- Read Web server logs, parse cookies and query string data to identify many Web site activities.
- Transform Web server log data into meaningful business intelligence that can be used to benefit your business (such as the number of times that a user accesses specific content).

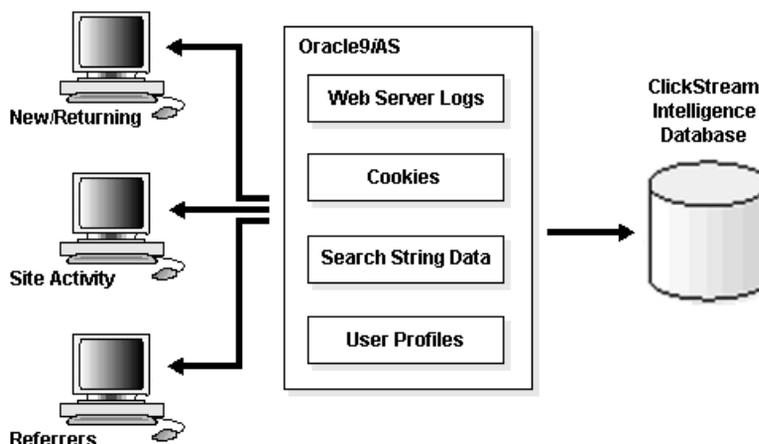
- Integrate Web traffic data with business transaction data from CRM and ERP applications.
- Condense enormous volumes of data into meaningful analytical reports.
- Provide Web site scalability through advanced data warehousing features to ensure performance is not compromised as Web site activity grows.
- View a full set of Web transaction reports with graphs that provide insight on key performance indicators.
- Use Web browser-based administration and configuration that can be updated and maintained from any computer running an Internet browser.
- Integrate with other Oracle9iAS Business Intelligence and data warehousing tools to ensure interoperability with all of your decision support tools.
- Integrate with Oracle9iAS Portal to create workbook portlets.

See Also: [Chapter 3, "Portal Development and Deployment"](#)

Oracle9iAS Clickstream Intelligence collects clickstream data from Web server and ISP database logs. Business intelligence data can be obtained through interface files or through published APIs. Oracle9iAS Clickstream Intelligence filters and loads all data into a predefined database, which provides an information resource for the creation of analytical reports. Standard reports include:

- General Site Activity
- Connecting Client Hosts
- Visitors and Users
- Search Analysis
- Browsers and Platforms
- Path Analysis
- Referrers

Figure 6–1 shows how Web data is collected using Web server logs, cookies, search string data, and user profiles. The collected data is stored in the Oracle9iAS Clickstream Intelligence Data Warehouse.

Figure 6–1 Oracle9iAS Clickstream Intelligence Data Acquisition and Processing

Oracle9iAS Clickstream Intelligence provides the performance, scalability, and extensibility (that is) essential for processing the potentially enormous data volumes generated by large-scale Web environments. Its extensible platform accommodates user-defined data sources, such as call centers, ad servers, and other data source types.

Oracle9iAS Clickstream Intelligence Features

Oracle9iAS Clickstream Intelligence provides the following features:

- [Enterprise Warehousing Environment](#)
- [Runtime Administrator](#)
- [Customize Information Delivery](#)
- [Different Data Sources Supported](#)

Enterprise Warehousing Environment

Oracle9iAS Clickstream Intelligence provides a rapidly deployable, prebuilt Clickstream Data Warehouse delivered in the form of an Oracle9i Warehouse Builder repository. This extensible Oracle9i Warehouse Builder data architecture enables organizations to integrate virtually any business type of transaction data into their Web analysis. In contrast, most competing solutions provide a fixed function data access architecture or require consulting fees for customization via complex APIs.

Oracle9iAS Clickstream Intelligence uses Oracle9i Warehouse Builder to provide an open Oracle database that can be accessed with a variety of tools. Integration with Oracle9i Warehouse Builder makes it possible for Oracle9iAS Clickstream Intelligence to provide meaningful clickstream analysis that can be targeted for customers, orders, or even marketing campaigns, producing results that are similar to those of traditional OLAP analysis.

Oracle9i Developer Suite provides the development tools for Oracle9iAS Clickstream Intelligence. Oracle9iAS Clickstream Intelligence, in conjunction with Oracle9i Warehouse Builder and Oracle9iAS Discoverer, provides the pre-built clickstream data warehouse model that can be extended to meet the growing and changing needs of an enterprise.

Oracle9iAS Clickstream Intelligence is specially designed to use advanced data warehousing features available in the Oracle9i Database Server. Oracle9iAS Clickstream Intelligence uses the Oracle9i Database Server to take advantage of materialized views, partitioning, and parallelism to optimize query performance, manage data volumes, and speed load times. Oracle9iAS Clickstream Intelligence is based on the Oracle9i Warehouse Builder repository.

Runtime Administrator

The Oracle9iAS Clickstream Intelligence Runtime Administrator simplifies the setup, management, and deployment of clickstream data sources. This Web-based tool enables you to define how Oracle9iAS Clickstream Intelligence interprets Web log file data, determine which types of data you want to track, and configure and populate the database for optimal storage of clickstream data.

Customize Information Delivery

Oracle9iAS Clickstream Intelligence provides a unique ability to customize information delivery to end-users by leveraging the infrastructure of Oracle9iAS Discoverer. Oracle9iAS Clickstream Intelligence provides a number of prebuilt Oracle9iAS Discoverer reports out of the box. In contrast, other solutions provide fixed function user interfaces with limited support for customization. The Oracle9iAS Discoverer Web-based reporting architecture makes it easy to access information over the intranet or Internet.

Oracle9iAS Clickstream Intelligence also uses the Oracle9i Discoverer Administrator End User Layer (EUL) that can be customized to address unique reporting needs. The EUL is a server-based, low maintenance, powerful metadata repository and query management engine for data warehouses and OLTP systems. It automatically performs query redirection to summary tables. The EUL is centrally stored in the RDBMS for scalability and ease of maintenance and is shared by both

the client and server. The EUL is created and maintained by the Oracle9i Discoverer Administrator.

See Also: *"Introduction to Oracle9iAS Discoverer" on page 6-17*

Different Data Sources Supported

Oracle9iAS Clickstream Intelligence supports the following data sources:

- Apache common and combined
- Microsoft IIS Extended
- W3C Extended Web log format

Oracle9iAS Clickstream Intelligence Architecture

Oracle9iAS Clickstream Intelligence features a browser-based tool called the Oracle9iAS Clickstream Intelligence Runtime Administrator. The Runtime Administrator defines the Web sites from which data will be acquired. It enables you to set up parameters that specify which data will be acquired and processed from Web server log files. Additionally, you can specify the connection to the Oracle9i Warehouse Builder repository and to the target data warehouse, as well as deploying the definitions and automatically creating ETL scripts (extraction, transformation, and loading).

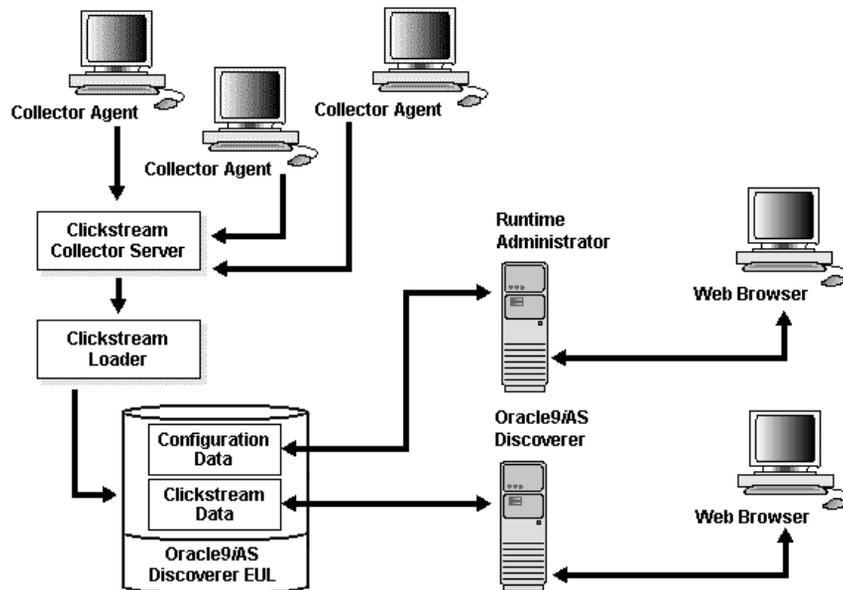
Figure 6-2, "Oracle9iAS Clickstream Intelligence Architecture" shows how data flows and is processed by Oracle9iAS Clickstream Intelligence:

1. The Collector Agents monitor the Web server log directory and create data packets from the log files. The Web logs may be non-Oracle9i Application Server server logs.
2. The Oracle HTTP Server performs compression, encryption, and transmission of the Web logs at intervals that are specified using the Oracle9iAS Clickstream Intelligence Runtime Administrator. Transmission may be completed in full or incrementally.
3. The Collector Server retrieves data packets from the Web server and decompresses the packets for subsequent transfer to the Clickstream Loader. The Collector Server resides on the database computer.
4. The Clickstream Loader parses, transforms and filters raw log files prior to database loading. The Loader uses a site's log format information and other site configuration parameters to parse and transform the Clickstream log files

before populating the database. The Clickstream Loader resides on the same computer as the Collector Server.

5. Oracle9iAS Discoverer performs query management.

Figure 6–2 Oracle9iAS Clickstream Intelligence Architecture



See Also: Oracle9iAS Clickstream Intelligence *Administrator's Guide*

Introduction to Oracle9iAS Personalization

Today's e-business must compete by maintaining personal, one-to-one relationships with their Web customers, including both registered customers and anonymous Web visitors. Companies must provide e-customers with customized product recommendations, ratings of the likelihood that the customer will “like” the recommendations, and improved site navigation based on their interests and profiles.

What is Oracle9iAS Personalization?

Oracle9iAS Personalization provides real-time analysis and recommendations for Web stores, application hosting environments, and business call centers. Oracle9iAS Personalization provides an integrated real-time recommendation engine that is deployed with Oracle9i Application Server.

By delivering real-time personalization, Oracle9iAS Personalization delivers powerful, scalable real-time personalization for customer areas of interest. This enables e-businesses to provide e-customers with customized product recommendations, rank the likelihood that a customer will like the recommendations, and improve site navigation based on customer interests and profiles.

Oracle9iAS Personalization is used by business-to-consumer and business-to-business customers and Application Service Providers (ASPs), who need to personalize their customer “touch points” to support business-critical operations to achieve a competitive advantage.

Oracle9iAS Personalization provides recommendations and answers to questions such as:

- Which item(s) is the customer most likely to buy or like?
- Customers that bought or like this item are likely to buy or like which other item(s)?
- How likely is the customer going to buy or like this item?
- Which item(s) is the customer most likely to buy or like given that the customer likes or is buying another items?

Oracle9iAS Personalization Features

Oracle9iAS Personalization provides the following features:

- [Real-Time Recommendation Engine Architecture](#)
- [Data Mining Technology](#)
- [Single Administrative Interface](#)

Real-Time Recommendation Engine Architecture

Oracle9iAS Personalization dynamically serves personalized recommendations (such as products, content, and navigational links) in real-time based on a registered customer's or anonymous visitor's explicit (transactions, purchases, ratings, and demographic data) and implicit (mouse clicks, pages visited, and banners viewed) information.

Data Mining Technology

Powerful data mining technology embedded in the Oracle9i Database Server automatically discovers individualized behavior patterns to generate highly accurate personalized recommendations in real-time. Oracle9iAS Personalization uses the data from data mining to build data mining models. The models predict what the Web site visitor will probably like or buy. The predictions are based on the data collected for that Web site visitor in previous sessions, in the current session, and on demographic information.

Oracle9iAS Personalization works in conjunction with existing Web applications. Web applications ask Oracle9iAS Personalization to record certain activities, and the data is saved by Oracle9iAS Personalization into an Oracle9i database schema. The Web application asks Oracle9iAS Personalization to produce a list of products likely to be purchased by a Web site visitor; a scored list of recommendations compiled from the visitor's current behavior and from data in another schema is passed to the Web application.

Single Administrative Interface

Oracle9iAS Personalization enables you to build, configure, manage, and deploy many recommendation engines throughout your enterprise from a single administrative interface.

Additionally, the administrative interface supports deployment of multiple recommendation strategies for different campaigns or time (such as holiday) periods. You can also capture and model behavior for specific events using the events scheduler.

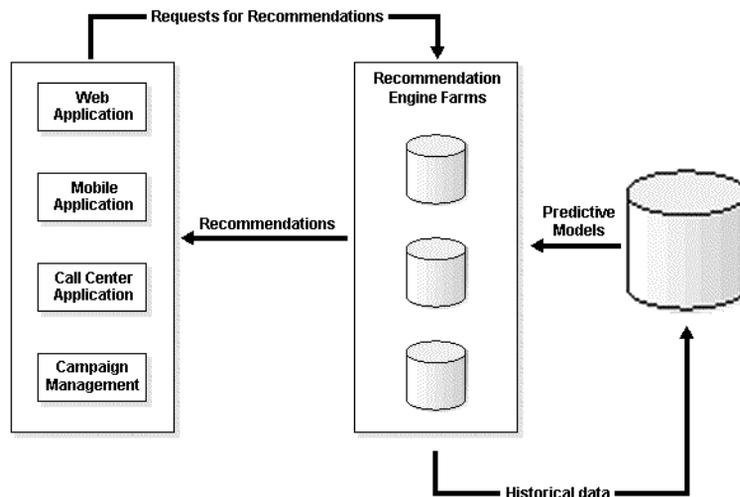
Oracle9iAS Personalization Architecture

Figure 6–3, "Oracle9iAS Personalization Architecture" shows how data flows and is processed by Oracle9iAS Personalization.

Oracle9iAS Personalization incorporates visitor activity, through Web, mobile, call center, and campaign applications into recommendations. This activity data is saved for that visitor in an Oracle9i database, where Oracle9iAS Personalization uses the data to build predictive models. Future visits to the Web site adds to the set of data for a customer. The new data allows for more accurate predictions.

Oracle9iAS Personalization uses SQL queries for obtaining scores, which can be executed in real-time or batch mode. Recommendation engines serve Oracle9iAS Personalization recommendations to Web sites across the enterprise.

Figure 6–3 Oracle9iAS Personalization Architecture



Oracle9iAS Personalization predictive models may be rebuilt on a periodic basis (e.g., daily, weekly, monthly) and deployed to the recommendation engines when completed. Oracle9iAS Personalization enables users to create "recommendation engine farms" that are comprised of many recommendation engines serving customized recommendations to the Web site.

The architecture is extremely scalable for high-traffic sites. Oracle9iAS Personalization stores the predictive models in memory to handle the high traffic and speed requirements associated with e-commerce sites. Data mining algorithms

find hidden patterns and customer profiles that drive personalized recommendations.

Oracle9iAS Personalization uses the Oracle9iAS Personalization Administrator to define a package that contains information needed to build a model or models, as well as information about the database connections. The Oracle9iAS Personalization Administrator creates and manages schedules for building the packages, and for deploying the packages to the recommendation engines that will produce the recommendations. Recommendation engines with the same package are grouped together in recommendation engine farms.

See Also: *Oracle9iAS Personalization User's Guide*

Introduction to Oracle9iAS Reports Services

In a business intelligence environment, data is generated by multiple sources in various formats. Publishing this business intelligence data should be fast and easy, without requiring multiple, convoluted steps to manipulate and consolidate the data into a single format.

What is Oracle9iAS Reports Services?

Oracle9iAS Reports Services provides a powerful deployment platform for publishing high-quality, dynamically generated end-user reports in a scalable, secure environment. You can build and publish sophisticated reports from any data source, in any format, and deploy them anywhere with Oracle9i Application Server.

Oracle9iAS Reports Services is part of Oracle9i Reports. Oracle9i Reports also includes Oracle9i Reports Developer, a component of the Oracle9i Developer Suite. Oracle9iAS Reports Services integration with servlets and JSPs is completed using Oracle9i Reports Developer. Using Oracle9i Reports Developer you can build a complex data model and share it between an existing high quality paper layout and an improved high quality Web layout using JSP technology.

Developers can publish sophisticated, high quality reports from any data source, in any data format, and deploy them anywhere on Oracle9i Application Server. Oracle9iAS Reports Services can combine data from multiple data sources into a single report, including Oracle9i, XML feeds, and JDBC-enabled data sources. Developers can even add custom data sources using the open API for Pluggable Data Sources.

Reports may be generated on demand or scheduled for a specific time or interval. Once formatted, a report can be distributed to a standard Web browser, email,

printer, or Oracle9iAS Portal, be stored on the middle tier for rapid retrieval, or be stored in a custom destination via the Pluggable Destinations API.

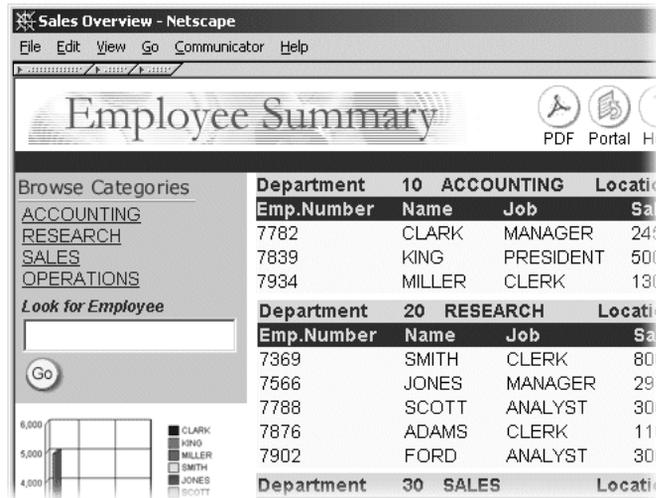
Oracle9iAS Reports Services enables you to:

- Access reports dynamically from any browser on demand.
- Leverage middle-tier load balancing to provide high volumes of reports, without excessive demands on limited resources.
- Generate reports in HTML, PDF for quality viewing, XML to communicate data to Web sites, RTF and delimited output for import into Microsoft Office, and PCL for printing.
- Optionally secure distribution of reports by allowing only specific users to access the report and grant secure access.
- Link to published reports and provide portlet support using Oracle9iAS Portal.

See Also: [Chapter 3, "Portal Development and Deployment"](#)

Figure 6–4 shows an example of an Oracle report.

Figure 6–4 Oracle Report Example



Note: Oracle9i Reports Developer is available with Oracle9i Developer Suite.

Oracle9iAS Reports Services Features

Oracle9iAS Reports Services provides the following features:

- [Event-Based Reporting](#)
- [Extensible Architecture](#)
- [Enhanced Report Bursting and Distribution](#)

Event-Based Reporting

Oracle9iAS Reports Services enables you to run a report that is triggered by database events. You can execute a report using a set of PL/SQL stored procedures that can be called from a database trigger. The trigger invokes Oracle9iAS Reports Services and the event data is submitted to the server. For example, when an employee submits an expense report, new data is being inserted into the database. The insert event causes a database trigger to fire, which sends a report to the employee manager on a portal page or in an e-mail message asking for expense report approval.

Extensible Architecture

Oracle9iAS Reports Services enables you to customize your access to data sources, runtime engines, security, notification, distribution, and cache:

- Write your own data access methods for data that is specific to your business. A Pluggable Data Source (PDS) is a set of Java APIs that provides openness to Oracle9iAS Reports Services by enabling data input from any source. PDSs allow Oracle9iAS Reports Services to combine data from various data sources including multiple Oracle database connections – each with potentially a different connection. When written to the PDS API, you can specify the icons to be displayed in the data model and Report Wizard of Oracle9i Reports Developer.
- Use Oracle9iAS Reports Services functionality with your own runtime engine or job type. The Reports engine is the default implementation provided by Oracle. However, you can take advantage of the security, notification, distribution and caching features of Oracle9iAS Reports Services using your own engine code. An example of runtime engine code is a Java class that uses

an operating system call to package a series of reports into a zip file and then distributes that file using e-mail to a defined set of users.

- Create your own security mechanisms for user-authentication against the Reports Server.
- Configure Oracle9iAS Reports Services to submit notifications upon success or failure of a job. Besides using the default implementation of sending an e-mail message, you can now implement your own notification mechanisms.
- Distribute to various destinations such as the Reports Services cache, e-mail messaging, and printers. You can add your own destinations such as fax machine and FTP servers using the pluggable destination API.
- Replace the default caching with your own caching mechanism for the output of the Oracle9iAS Reports Services using the pluggable caching API.

Enhanced Report Bursting and Distribution

Oracle9iAS Reports Services enhanced bursting and distribution enables you to distribute a report to different media, such as paper, e-mail message, Web page, or even delimited output like a spreadsheet.

You can burst on repeating data within a report. Bursting enables you to create multiple personalized reports out of a single report model. For example, department-based reports can be distributed to employees within that department. This ensures that each user only receives information that is relevant and appropriate for them. The manager of a sales department will get an e-mail with the salary information for employees in that department; the manager of the purchasing department will only be e-mailed the salary information for employees in that department; the CEO of the company will receive a print out containing a summary of each department's information.

The report runs only once, and the output is sent to different sets of users according to requirements. You only have a single set of data to fetch and format in order to distribute the report.

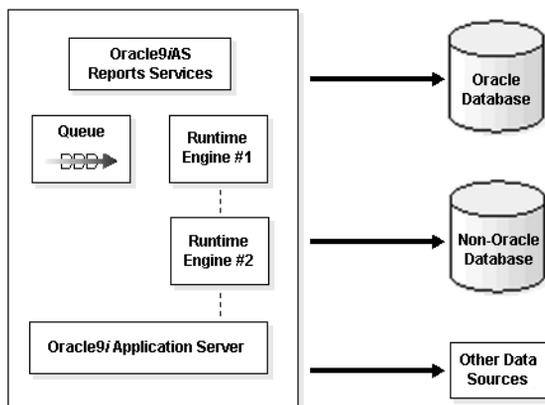
Oracle9iAS Reports Services Architecture

Oracle9iAS Reports Services handles client requests to run reports by entering all requests into a job queue. When one of the server's runtime engines becomes available, the next job in the queue is dispatched to run.

As the number of jobs in the queue increases, the server can start more runtime engines until it reaches the maximum limit specified in your server configuration. Oracle9iAS Reports Services runtime engines are shut down automatically after having been idle for a period of time that you specify.

Oracle9iAS Reports Services keeps track of all jobs submitted in the server, including jobs that are running, scheduled to run, or finished. The Reports Queue Manager (Windows), the Reports Queue Viewer (UNIX), or the `show jobs` command (Web) enable you to view information on when jobs are scheduled, queued, started, and finished, as well as the job output and the final status of the report.

Figure 6–5 Oracle9iAS Reports Services Architecture



Oracle9iAS Reports Services job objects are persistent. This means that if the server is shutdown then restarted, all jobs are recovered, not just scheduled jobs.

Oracle9i Reports Developer enables you to embed a report within a larger existing Web page. This technology enables you to open and save HTML, JSP, and XML files that contain report definitions. When a report is saved as a JSP file, the data model is embedded using XML tags. The entire report can also be defined using XML tags and saved as an XML file.

Oracle9i Reports Developer enables you to take retrieved data, using the data model, and embed it into an existing Web page. This provides tremendous flexibility in creating reports that meet business demands by completely integrating multiple sources of information within a single Web page. If you choose to create a JSP from scratch, Oracle9i Reports Developer supplies templates that can be used to build your report.

Developers can easily create a JSP layout using the Reports Block Wizard to generate the necessary JSP tags in Oracle9i Reports Developer. Alternatively, they can add the tags themselves manually for more precise control.

See Also: *Oracle9iAS Reports Services Publishing Reports to the Web*

Note: Oracle9i Reports Developer is available with Oracle9i Developer Suite.

Introduction to Oracle9iAS Discoverer

In order to formulate and deploy business strategies, businesses must transform gathered data into useful and informative information. The transformed data must be rapidly accessible through an easy to use interface. The user must be able to quickly query, graph, and create reports on the gathered data.

What is Oracle9iAS Discoverer?

Oracle9iAS Discoverer is an intuitive ad-hoc query, reporting, analysis, and Web publishing tool. Using Oracle9iAS Discoverer, business users at all levels of the organization gain immediate access to information from data warehouses, clickstream data, and OLTP systems.

Using Oracle9iAS Discoverer you can rapidly view information in user customized summary formats and perform complex calculations on data (these complex calculations can include analytic functions that are available in the Oracle database).

Oracle9iAS Discoverer comprises Oracle9iAS Discoverer Plus and Oracle9iAS Discoverer Viewer. Oracle9iAS Discoverer Plus and Oracle9iAS Discoverer Viewer enable business users to view and create reports.

[Figure 6–6](#) shows an example of an Oracle9iAS Discoverer result set.

Figure 6–6 Oracle9iAS Discoverer Result Set Example

The screenshot shows the Oracle Discoverer interface with a pivot table titled "Profit SUM". The table is filtered by "Department: Video Sale". The pivot table structure is as follows:

	Profit SUM							
	1998				1999			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Region								
Central	\$15,531	\$14,734	\$19,053	\$17,766	\$21,670	\$24,768	\$24,181	\$27,302
East	\$25,544	\$28,758	\$26,313	\$27,944	\$31,134	\$36,177	\$35,805	\$42,346
West	\$13,752	\$16,288	\$14,296	\$12,761	\$20,307	\$21,271	\$22,007	\$23,587

For example, a manager might view a summary report of sales that shows the top performing products for the quarter, and drill down to view results by region or by sales person. The manager could also use Oracle9iAS Discoverer to perform a quick comparison with previous quarters, or to calculate the three-month moving average of the organization's sales.

Oracle9iAS Discoverer enables you to present the data in a workbook format containing the data you want to see. Using Oracle9iAS Discoverer, you can:

- Obtain information from a database without knowing how the database is structured, or how to use a database programming language.
- Obtain the latest information without waiting for reports to run overnight, or for programmers to write reports for you.
- Analyze data interactively using Oracle9iAS Discoverer's powerful analysis tools to drill, pivot, and graph information.
- Create worksheets in tabular and cross-tab formats using Oracle9iAS Discoverer wizards.

With Oracle9iAS Discoverer data analysis tools, you can find the answers to business questions by:

- Pivoting data to create comparisons.
- Drilling into data to see more detailed information.
- Drilling out of data to see consolidated information.
- Drilling out to analyze data in other applications.
- Creating totals, calculating percentages, and creating custom calculations on your data.

The end result is a collection of worksheets and graphs that persuasively:

- Support your business decisions.
- Help you identify business opportunities.

Oracle9iAS Discoverer Tools

Oracle9iAS Discoverer provides the following tools:

- [Oracle9iAS Discoverer Viewer](#)
- [Oracle9iAS Discoverer Plus and Oracle9iAS Discoverer Desktop](#)
- [Oracle9iAS Discoverer and Oracle9iAS Portal](#)
- [Oracle9i Discoverer Administrator](#)

Oracle9iAS Discoverer Viewer

Oracle9iAS Discoverer Viewer enables business users to run, view, and graph reports created by users of Oracle9iAS Discoverer Plus. Oracle9iAS Discoverer Viewer is a true lightweight business intelligence tool, requiring no other software than a simple Web browser. Its Web interface is familiar to anyone who regularly uses a browser to view pages on the Internet, and it uses the same high performance, scalable architecture as other Oracle9iAS Business Intelligence solutions. Oracle9iAS Discoverer Viewer reports are live, enabling users to interact with their data by changing layouts and specifying parameters. In addition, Oracle9iAS Discoverer Viewer lets you embed a report within a standard Web page to integrate business intelligence with your Web site.

Oracle9iAS Discoverer Viewer converts large documents for viewing as pages. Oracle9iAS Discoverer Viewer breaks the document into pages in the background and displays the first page. Additional pages are displayed on demand reducing

network traffic between the client and the Oracle9iAS Discoverer server. The Oracle9iAS Discoverer server monitors client activity and ends inactive sessions after a specified time out limit. This frees up valuable resources for other users.

Oracle9iAS Discoverer Plus and Oracle9iAS Discoverer Desktop

Oracle9iAS Discoverer Plus and Oracle9iAS Discoverer Desktop enables business users to query, graph, and create reports. Oracle9iAS Discoverer Plus runs as a Java client and Oracle9iAS Discoverer Desktop runs in a client/server architecture. You can perform drill and pivot analysis, and publish results to the World Wide Web.

Similar to Oracle9iAS Discoverer Viewer, Oracle9iAS Discoverer Plus converts large documents for viewing as pages. Oracle9iAS Discoverer Plus breaks the document into pages in the background and displays the first page. Additional pages are displayed on demand reducing network traffic between the client and the Oracle9iAS Discoverer server.

Oracle9iAS Discoverer and Oracle9iAS Portal

Integration between Oracle9iAS Discoverer and Oracle9iAS Portal enables you to publish reports or a list of reports to Oracle9iAS Portal. This enables other users in the organization to easily access the information by subscribing to these portal pages. Oracle9iAS Discoverer is a portlet provider to Oracle9iAS Portal. There are two types of Oracle9iAS Discoverer portlet: a List of Workbooks Portlet, and a Worksheet Portlet. A List of Workbooks Portlet includes the names of workbooks available from Oracle9iAS Discoverer and a Worksheet Portlet may contain data only (table or cross-tabular), a graph only, or both.

See Also: Chapter 3, "Portal Development and Deployment"

Oracle9i Discoverer Administrator

The Oracle9iAS Discoverer manager uses Oracle9i Discoverer Administrator to perform a variety of tasks including maintenance of business areas, folders, summary table creation, and end user access. Oracle9i Discoverer Administrator provides extensive defaulting capability to enable managers to rapidly implement their end user environment. The Oracle9i Discoverer Administrator is a required component of a complete Oracle9iAS Discoverer installation and is a component of Oracle9i Developer Suite. Oracle9i Discoverer Administrator is required for support of Oracle9iAS Discoverer Plus and Oracle9iAS Discoverer Desktop.

Oracle9i Discoverer Administrator is used for the initial setup and ongoing maintenance of the EUL. The EUL is a server-based, low maintenance, powerful metadata repository and query management engine for data warehouses and OLTP

systems. It automatically performs query redirection to summary tables. The EUL is centrally stored in the RDBMS for scalability and ease of maintenance and is shared by both the client and server. The EUL is created and maintained by the Oracle9i Discoverer Administrator.

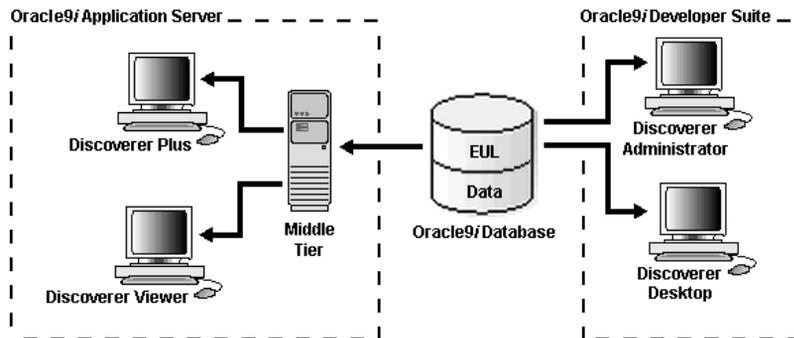
Note: Oracle9i Discoverer Administrator is available with Oracle9i Developer Suite.

Oracle9iAS Discoverer Architecture

Figure 6-7, "Oracle9iAS Discoverer Architecture" shows how data flows and is processed by Oracle9iAS Discoverer.

Oracle9iAS Discoverer uses an efficient cache that is completely transparent to both the end user and the administrator. The cache enables rapid analysis without re-querying the database. In a Web environment, this cache is located on a middle tier server and is associated with the user's session. This enables end users to ask subsequent questions about the data without having to re-execute the query on the database server. Pivoting of data and many calculations are handled on the middle tier to provide exceptional performance. As users drill for more detail, Oracle9iAS Discoverer fetches only the additional information requested and incorporates this into the existing cache, reducing both database and network traffic.

Figure 6-7 Oracle9iAS Discoverer Architecture



The Oracle9iAS Discoverer EUL enables you to access the same metadata and share reports regardless of client type.

See Also: *Oracle9iAS Discoverer User's Guide*

Integrate Oracle9iAS Business Intelligence

Oracle9iAS Business Intelligence provides a fully integrated e-business intelligence environment. Using Oracle9iAS Business Intelligence you can:

- [Create an E-Business Dashboard Using Oracle9iAS Portal](#)
- [Leverage Single Signon Functionality](#)

Create an E-Business Dashboard Using Oracle9iAS Portal

Integrate Oracle9iAS Portal and Oracle9iAS Business Intelligence to create an E-Business Intelligence dashboard. A portlet is a reusable component and basic building block of a portal page. Portlets produce part of a Web page in either HTML or XML.

All Oracle9iAS Business Intelligence components can be integrated into the portal as portlets. Using Oracle9iAS Business Intelligence components with Oracle9iAS Portal you can:

- Distribute Oracle9iAS Clickstream Intelligence workbooks as portlets.
- Integrate with Oracle9iAS Personalization applications to create portlets.
- Create and distribute Oracle9iAS Reports Services reports as portlets.
- Create Oracle9iAS Discoverer List of Workbooks Portlets or Worksheet Portlets.

See Also: [Chapter 3, "Portal Development and Deployment"](#)

Leverage Single Signon Functionality

With increasing use of Web-based e-business applications for deployment to employees, customers, and partners, businesses are considering a single signon functionality. With Oracle9iAS Single Sign-On, each user maintains only one identity and password for all applications they access. Oracle9iAS Single Sign-On creates an enterprise-wide user authentication to access multiple accounts and Oracle9i Application Server applications.

All Oracle9iAS Business Intelligence components leverage Oracle9iAS Single Sign-On functionality to provide a seamless point of access to all of your business intelligence users.

See Also: ["Oracle9iAS Single Sign-On"](#) on page 8-9

Oracle9iAS Integration

This chapter describes the integration features of Oracle9i Application Server.

This chapter contains these topics:

- [About Oracle9iAS Integration](#)
- [Oracle9iAS InterConnect](#)
- [Additional Integration Features](#)

About Oracle9iAS Integration

The conduct of business over the Internet, such as buying and selling products, servicing customers, and collaborating with business partners is known as e-business. A critical requirement for e-business success is application integration. Application integration is the ability to link different types of enterprise applications and business processes together so that they can smoothly and effectively communicate to conduct e-business. These applications can reside within a company's enterprise boundaries or across multiple company boundaries connected over the Internet.

Successfully integrating applications can be very challenging. For example:

- Different applications use different technologies and architectures.
- Applications constantly evolve, thus complicating implementation and maintenance of application integration.
- Managing complexity is difficult and expensive.
- Costs and schedules are often difficult to predict.

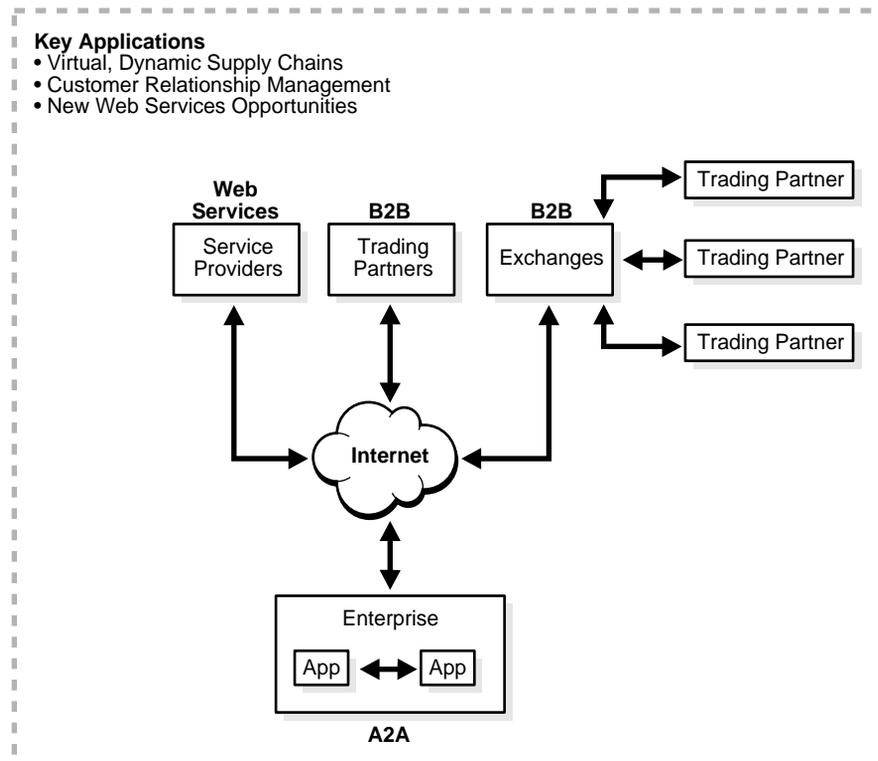
Oracle9iAS provides integration services in a single product. These services address the challenges of designing, deploying, and managing e-business application integration. Oracle9iAS enables organizations to capitalize on Internet-enabled applications such as:

- Dynamic supply chains
- Customer relationship management (CRM)
- Emerging Web services opportunities

Oracle9iAS provides an integration infrastructure that enables you to integrate virtually any packaged or legacy application. Oracle9iAS is also the best solution for Oracle E-Business Suite applications, as it enables organizations to quickly link these applications with third-party packaged applications such as SAP and PeopleSoft or virtually any legacy application.

Figure 7-1 provides an overview of Oracle9iAS application integration.

Figure 7-1 Oracle9iAS Application Integration Overview



Oracle9iAS InterConnect

Oracle9iAS InterConnect is the key Oracle9iAS integration feature. Oracle9iAS InterConnect acts as the integration hub that coordinates the communication and transformation of messages between two or more heterogeneous applications. Oracle9iAS InterConnect defines business events, their associated data, and any transformations required to map one application's view of a business object to another's view.

Oracle9iAS InterConnect enables applications to communicate within a company's enterprise or across a company's enterprise boundaries and over the Internet. Within a company's enterprise (inside the firewall), Oracle9iAS InterConnect provides application-to-application (A2A) integration support that enables automatic communication between:

- CRM applications
- Enterprise resource planning (ERP) applications
- Custom applications
- Legacy systems

Across a company's enterprise boundaries (outside the firewall), Oracle9iAS InterConnect enables business-to-business (B2B) integration over Internet protocols such as:

- HTTP(S)
- FTP
- SMTP

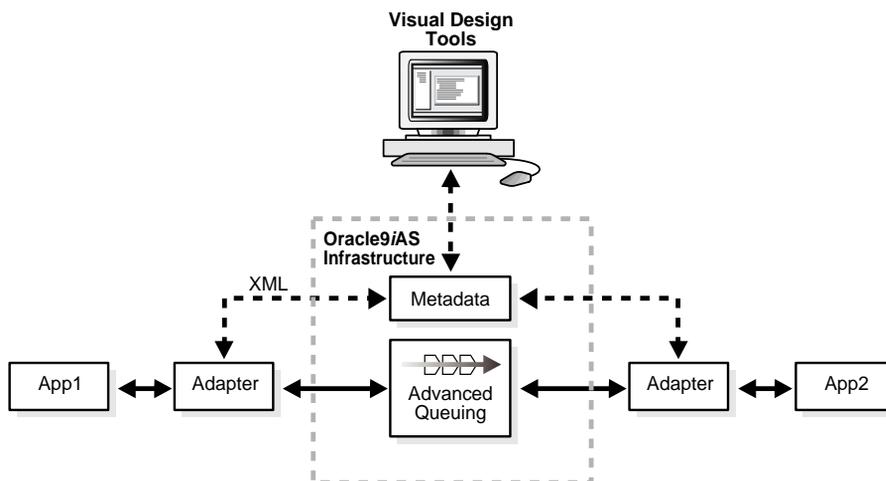
B2B integration links businesses with trading partners indirectly through exchanges or directly through bilateral communication. B2B integration also enables application integration between hosted applications in an application service provider (ASP) environment and in-house applications.

Oracle9iAS InterConnect provides the following integration benefits:

- Enables implementers to focus on business requirements rather than low-level technical implementation details
- Reduces costs and complexity by providing a single solution for A2A, B2B, and ASP integration requirements. This enables organizations to use the same skills to implement both internal and external integration solutions.

[Figure 7-2](#) provides a high-level overview of Oracle9iAS InterConnect's role as the integration hub between two applications.

Figure 7–2 Oracle9iAS InterConnect



Oracle9iAS InterConnect includes several key components and features that enable application communication:

- Visual design tools aimed at business users rather than developers. These tools include:
 - iStudio for defining the applications to participate in integration and the specifications for the data to exchange
 - Oracle Workflow for creating business process collaborations between applications. Oracle Workflow includes Oracle Workflow Builder, which lets you create, view, or modify a business process with simple drag and drop operations.
 - Oracle Enterprise Manager for monitoring and managing runtime components
- Oracle9iAS Metadata Repository, which controls the runtime behavior of application integration. The repository also provides full life cycle support to simplify maintenance and enhancement of the integration environment over time.
- Extensible transformation capabilities
- Oracle9iAS InterConnect adapters for connecting applications
- Content-driven message routing
- Open software development kits (SDKs) that enable developers to build custom transformation functions and adapters

In addition to these components and features, Oracle9iAS InterConnect supports a common view application integration methodology. A common view is a logical model of the business events and objects involved in an integration environment. Each application presents an application view of business events they can produce and/or use. Application view events are mapped to common view events to satisfy the integration solution requirements. Transformation functions modify the data to and from the common view.

The common view simplifies integration of more than two applications by eliminating multiple point-to-point integration complexities. Maintenance and upgrades are simplified since the common view insulates participating applications from one another. Application changes are generally limited to the single application affected.

The following sections provide additional information on several key Oracle9iAS InterConnect features:

- [Oracle9iAS InterConnect Adapters](#)
- [Oracle Workflow](#)
- [Oracle Advanced Queuing](#)

See Also:

- *Oracle9iAS InterConnect User's Guide*
- [Chapter 8, "System Management"](#) for information on Oracle9iAS integration management with Oracle Enterprise Manager

Oracle9iAS InterConnect Adapters

Oracle9iAS InterConnect provides a set of connectivity adapters that enable third-party applications and technology environments to participate in integration. [Table 7-1](#) identifies these adapters.

Table 7-1 Oracle9iAS InterConnect Adapters

Third-Party Adapters	Technology Adapters
<ul style="list-style-type: none"> ■ SAP/R3 ■ PeopleSoft ■ Siebel ■ JD Edwards OneWorld 	<ul style="list-style-type: none"> ■ HTTP(S) ■ SMTP ■ MQSeries ■ FTP ■ Oracle Advanced Queuing ■ Database ■ CICS

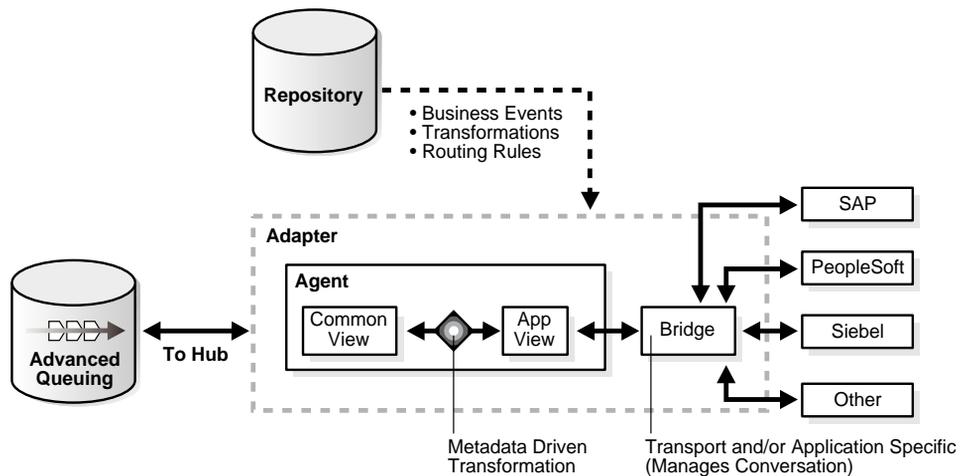
You can develop additional adapters using Oracle9iAS InterConnect's adapter SDK.

Adapters consist of two key parts:

- An agent provides functionality shared by all adapters. The agent receives application view events and transforms them to corresponding common view events, and vice versa. The event definitions and the associated transformation rules are maintained in the Oracle9iAS Metadata Repository and downloaded to the adapter at runtime.
- A bridge provides functionality unique to each adapter type. The bridge translates technology and/or application-specific interfaces to and from the Oracle9iAS InterConnect environment. For example, in the SAP/R3 adapter, the bridge translates application view events to and from biometric application programming interface (BAPI) and intermediate document (IDoc) calls in SAP/R3. When you develop additional adapters with the Oracle9iAS InterConnect adapter SDK, you essentially create your own bridge.

[Figure 7-3](#) shows the agent and bridge components of an adapter.

Figure 7-3 Agent and Bridge Components of an Adapter



See Also: See [Figure 7-2](#) on page 7-5 for the location of adapters in the integration process

Oracle Workflow

Business processes are the coordinated set of activities that companies follow when conducting business. Business process management is a key requirement for application and business integration. Oracle Workflow lets you automate and continuously improve business processes, routing information of any type according to easily-changeable business rules to users both inside and outside a company's enterprise.

Oracle Workflow provides the following capabilities in one integrated tool:

- Automated processing of business events and messages using rules defined in a business process model
- Traditional workflow functions such as sending notifications and authorizations by Web or e-mail messages to the appropriate user. For example, you may want to require manual authorization of orders or purchases over a specified dollar amount. Similarly, you may need to notify someone in the event of an exception condition in the integration process.

Oracle9iAS InterConnect and Oracle Workflow are integrated at design time and runtime. This enables you to access the Oracle Workflow Builder tool and define business processes and roles from within iStudio.

- During design, business processes and event definitions in iStudio can be deployed to Oracle Workflow Builder with one click. This enables you to create graphical integration business process diagrams.
- At runtime, Oracle9iAS InterConnect and Oracle Workflow communicate through reliable message queues using Oracle Advanced Queuing.

See Also: *Oracle9iAS InterConnect User's Guide*

Oracle Advanced Queuing

Oracle Advanced Queuing provides database-resident messaging capabilities that enable reliable, asynchronous communication between applications. Oracle9iAS integration uses Oracle Advanced Queuing as the communication mechanism between adapters, participating applications, and Oracle Workflow. (See [Figure 7-2](#) on page 7-5.)

Oracle Advanced Queuing supports the following reliable message delivery modes:

- Point-to-point
- Publish and subscribe
- Point-to-multipoint

Oracle Advanced Queuing provides several key benefits, including:

- Integrated data, message, and message history
- Recoverability (any point in time recovery)
- Auditing and tracking /visibility of in-process transactions
- Automated database logging to support business intelligence
- High performance (thousands of messages per second per queue) with scalability managed by database features such as parallel processing

Additional Integration Features

Oracle9iAS includes additional integration features that provide communication and integration capabilities for e-business applications:

- [Oracle Internet File System](#)
- [Oracle9iAS Unified Messaging](#)

Oracle Internet File System

Oracle Internet File System (Oracle 9iFS) runs in conjunction with Oracle9i Application Server and an Oracle9i database. Written entirely in Java, Oracle 9iFS is an extensible content management system with file server convenience. Oracle 9iFS is accessible through a variety of protocols, including:

- AppleTalk Filing Protocol (AFP)
- HyperText Transfer Protocol (HTTP)
- Network File System (NFS)
- Server Message Block (SMB)
- Distributed Authoring and Versioning extensions to HTTP (WebDAV)
- Internet Message Access Protocol (IMAP)
- File Transfer Protocol (FTP)

However, unlike traditional protocol servers, all content is stored in an Oracle database, not on the file system of the server machine. This means that you gain all the benefits of the Oracle9i Database Server, such as backup and recovery and full-text search capability (by means of Oracle Text). Better yet, by using Oracle9i Application Server as the basis for your middle-tier servers with Oracle 9iFS, you can gain load-balancing and numerous other benefits.

No additional client software is required: Users access Oracle 9iFS using the native client protocols available on their operating systems. For example, Mac users go to the Chooser and connect to Oracle 9iFS as if it were any other AppleShare server; Windows users map a network drive or connect using Web Folders; UNIX clients can connect using NFS or FTP. (Oracle does provide optional client applications, however, such as Oracle 9iFS FileSync and Oracle 9iFS Windows Utilities, that enhance file management for Windows users). These are just a few examples.

Development Platform

In addition to the file-system application features of the product, Oracle 9iFS exposes a set of APIs that third-party developers can use to customize and extend

the product's core functionality in numerous ways to build comprehensive content-management and collaboration systems.

Oracle 9iFS integrates with Oracle9i Application Server components, including Oracle9iAS Containers for J2EE, Oracle Internet Directory, and Oracle Enterprise Manager. For information about installing and configuring Oracle 9iFS, see the *Oracle9iFS Installation and Configuration Guide*.

See Also:

- *Oracle Internet File System Developer Reference* for information on creating custom file system applications
- *Oracle Internet File System Setup and Administration Guide* for information on administering Oracle 9iFS

Oracle9iAS Unified Messaging

Oracle9iAS Unified Messaging provides access to all message types using any access method. Voice mail, e-mail, fax, and any other mail type are available to users through their choice of access channel and device. A single message store is used for voice mail, e-mail, and fax messages that provides storage, management, and access to all types of information. The message store provides delivery, telephone processing, wireless notification, browser-based clients (both Web and wireless), and administration utilities.

The following sections describe the Oracle9iAS Unified Messaging architecture:

- [Message Store](#)
- [Open Standards-Based Messaging](#)
- [Thin Client](#)
- [Web-Based Calendar](#)
- [Telephone Applications](#)
- [Extended Server Side Rules](#)
- [Application Integration](#)
- [Administration](#)

Message Store

All messages, including voice mail, e-mail, and fax, are stored in an Oracle9i database. This eliminates the need to synchronize message stores and administer multiple stores with different message types. Users access and manage messages from their choice of interface, including Web browser, phone, personal digital assistant (PDA), and fax.

Open Standards-Based Messaging

You can access messages with Internet Message Access Protocol version 4 (IMAP4)-compliant clients or Post Office Protocol version 3 (POP3)-compliant clients, including:

- Netscape Messenger
- Microsoft Outlook Express
- Eudora Pro Lite

Oracle9iAS Unified Messaging provides directory services using the lightweight directory access protocol (LDAP) standard-compliant Oracle Internet Directory.

Thin Client

The thin client provides Internet access to Oracle9iAS Unified Messaging with a Web browser. Users access and manage their account (including calendar and to-dos) with the Web browser. Functionality is also available through the wireless Web using wireless markup language (WML) and Oracle9iAS Wireless Edition.

Web-Based Calendar

Oracle Web Calendar provides Web-based calendaring and scheduling for enterprises and service providers.

Telephone Applications

Telephone applications are created based on Enterprise Computer Telephony Forum (ECTF) standards. ECTF standards describe how to create and support platform independent computer telephony (CT) applications and enable telephone applications to integrate with enterprise and carrier class switches.

A basic dual tone multifrequency (DTMF) voice mail interface for changing user preferences with the telephone interface is provided. Actions taken on a message or on account preferences through the voice channel are visible on all channels. Voice mail and faxes are stored directly in a user's inbox as industry standard Multipurpose Internet Mail Extensions (MIME)-compliant messages.

Extended Server Side Rules

Oracle9iAS Unified Messaging extends the server side rules previously available in Oracle9iAS Email 5.2. Rules execute actions and are applied to mail events, including sending, relaying, deleting, and expunging messages. A PL/SQL API is included for users to create their own customized rule actions.

Application Integration

PL/SQL and Java programmers can create custom application interfaces to integrate Oracle9iAS Unified Messaging with other applications. Oracle9iAS Unified Messaging APIs enable applications to directly manipulate stored messages and create MIME-standard outgoing messages.

Administration

Oracle9iAS Unified Messaging integrates with Oracle Enterprise Manager. For the hosting market, support is provided for multiple domains with delegated administration on the same system. Simple Network Management Protocol (SNMP) management information bases (MIBs) for system monitoring and alerts are also supported.

See Also:

- *Oracle9iAS Unified Messaging Concepts Guide*
- *Oracle9iAS Unified Messaging User's Guide*
- *Oracle9iAS Unified Messaging Administrator's Guide*
- *Oracle9iAS Unified Messaging Developer's Guide*

System Management

This chapter provides an overview of Oracle9i Application Server system management features and benefits. The topics include:

- [System Management Overview](#)
- [System Management Architecture](#)
- [About Management](#)
- [Management Features](#)
- [About Security](#)
- [Security Features](#)

System Management Overview

In today's Internet economy, where success is directly impacted by the availability and performance of your production Web sites and applications, you need to be able to rely on a highly secure, highly available, and finely tuned production deployment platform.

Oracle9i Application Server provides a set of industry standards-based management facilities to simplify all aspects of Web site administration. It includes a comprehensive deployment infrastructure designed to streamline application deployment by leveraging a single security and directory framework for all applications and by providing administrators with a single management tool to manage, monitor, tune, and troubleshoot across all Oracle9iAS instances.

Oracle9i Application Server uses Oracle Enterprise Manager to enable Web site administrators to configure Oracle9iAS instances, to monitor and optimize them for performance and scalability, and to respond proactively to problem conditions.

Oracle9i Application Server also provides an SSL/X.509V3 Certificate-based security architecture, which provides user authentication through SSL certificates, encrypted SSL communication, and single signon across applications.

Oracle Internet Directory, an LDAP directory, provides a single repository and administration for user accounts.

System Management Architecture

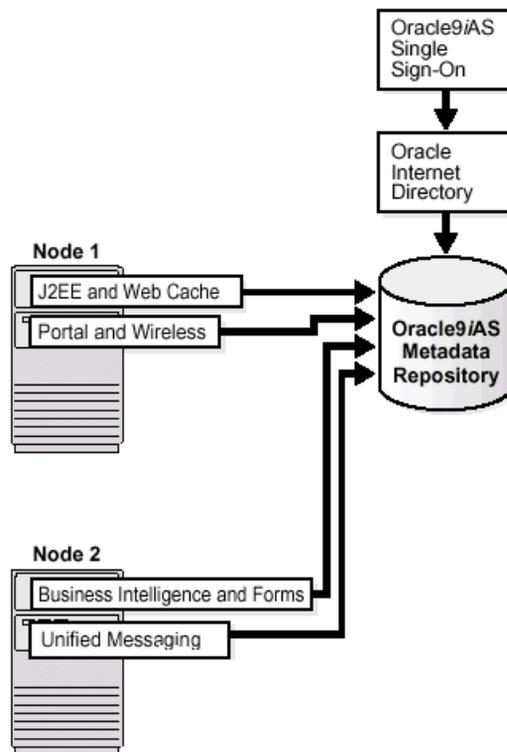
Oracle9i Application Server System Management architecture consists of the following sections:

- **Middle Tier:** All of Oracle9i Application Server applications, such as J2EE and Web Cache, Portal and Wireless, Business Intelligence and Forms, and Unified Messaging, exist and are executable on the middle tier.
- **Infrastructure:** Infrastructure installations provide a centralized location for managing data used by application servers. Most of this data is managed programmatically by application server instances and components and is invisible to users. An infrastructure contains the following components:
 - **Metadata Repository** is an Oracle9i Enterprise Edition database. It is pre-seeded with the schemas used by all Oracle9iAS components, including any schemas needed for components demos.

- **Oracle9iAS Single Sign-On** enables users to access multiple accounts and applications with a single password. Oracle9iAS Single Sign-On retrieves user information from Oracle Internet Directory.
- **Oracle Internet Directory** is Oracle's implementation of Lightweight Directory Access Protocol (LDAP), version 3. Application server instances, components, and infrastructures store security and management information in the directory.

An example of such an architecture is shown in [Figure 8-1](#).

Figure 8-1 Oracle9i Application Server System Management Architecture



See Also:

- *Oracle9i Application Server Administrator's Guide*
- *Oracle9i Application Server Installation Guide*

About Management

Oracle Enterprise Manager provides an integrated solution for centrally managing your Oracle environment. Combining a graphical console, common services, and administrative tools, Oracle Enterprise Manager provides a comprehensive system management platform for managing Oracle products.

As an Oracle9iAS administrator, you can use the Oracle Enterprise Manager Web site and the Oracle Enterprise Manager Console to manage your Oracle9iAS installations.

See Also: *Oracle9i Application Server Administrator's Guide*

Using the Oracle Enterprise Manager Web Site

For Oracle9iAS Release 2.0, Oracle introduces the Oracle Enterprise Manager Web site, which provides Web-based management tools designed specifically for Oracle9iAS. Using this Web site, you can monitor and configure the components of your Oracle9i Application Server installations. You can deploy applications, manage security, and create and manage Oracle9i Application Server clusters.

The Oracle Enterprise Manager Web site is installed on the middle tier as part of the Oracle9i Application Server installation option.

Depending on the type of Oracle9iAS installation you support, you can use the Enterprise Manager Web site independently to manage individual Oracle9iAS instances and clusters, or you can use it with the Oracle Enterprise Manager Console to manage your entire Oracle environment.

Using the Oracle Enterprise Manager Console

The Oracle Enterprise Manager console provides a wider view of your Oracle environment, beyond Oracle9iAS. Use the Console to automatically discover and manage databases, application servers, and Oracle applications across your entire network.

The Console and its related components are installed with the Oracle Management Server as part of the Oracle9iAS Infrastructure installation option. The Console is part of the Oracle Management Server component of the Oracle9iAS Infrastructure.

The Management Server, the Console, and Oracle Agent are installed on the Oracle9iAS Infrastructure host, along with the other infrastructure components.

See Also: *Oracle9i Application Server Installation Guide*

Management Features

Oracle Enterprise Manager includes the following features that enable you to manage your Oracle9iAS framework:

- [Single Point for Distributed Configuration and Management](#)
- [Performance Monitoring](#)
- [Dynamic Monitoring Service](#)
- [J2EE Deployment and Administration](#)

Single Point for Distributed Configuration and Management

Oracle Enterprise Manager Web site provides a complete management solution for administering and configuring the application server and its components. It allows you to effectively manage the Oracle9iAS environment remotely by enabling tasks such as:

- Automatically detecting all Oracle9iAS instances and their associated components on a target node for management by the Oracle9iAS administrative tools.
- Providing the ability to start and stop all Oracle9iAS components.
- Providing a single point to create and manage Oracle9iAS farms and clusters. Clusters provide a central point for managing multiple Oracle9iAS instances that host a common set of J2EE applications.
- Performing common administrative operations for all Oracle9iAS components, including configuring ports and log files.

Oracle Enterprise Manager Web site and the infrastructure repository simplifies configuration and administration of the Oracle9iAS infrastructure and provides you with all the tools you need to focus on monitoring key day and day usage and performance patterns.

Performance Monitoring

Accurate, timely, and relevant performance and troubleshooting data is absolutely essential in maintaining and monitoring the state of your production Web sites and applications. Oracle9iAS provides a host of critical performance data that are essential to tune your application server, identify resource availability issues, or help tune your application server to achieve maximum throughput and the minimum response time possible.

Oracle Enterprise Manager Web site provides a comprehensive aggregate view of the current and historical status of your Oracle9iAS environment. It allows you to monitor status, usage, and performance data for your entire Oracle9iAS environment including:

- Instance and component availability and uptime statistics
- Resource utilization including CPU and memory
- Server load and responsiveness including user volumes, active requests, request throughput and process time, error rates

Dynamic Monitoring Service

Oracle Enterprise Manager leverages an underlying service, the Dynamic Monitoring Service (DMS), to collect the performance data discussed above. Oracle9iAS has been fully instrumented with DMS to provide a comprehensive set of built-in performance metrics to automatically measure runtime performance statistics, which allows you to monitor the duration of important phases of request processing as well as status information such as the number of requests being handled at any given moment.

In addition, you can easily add monitoring to your own applications through the DMS library and API. The performance metrics are measured automatically and continuously using efficient performance instrumentation hooks, no extra configuration is required. You can view your DMS metrics through a browser as well as through the OEM console.

J2EE Deployment and Administration

Oracle9iAS enables you to easily administer OC4J and your J2EE applications. This interface supports:

- Configuring OC4J through declarative property sheets, as an alternative to manually editing XML files.
- Configuring application services and resources including Web sites (IP and URL paths for Web modules), JDBC data sources and security settings including user groups and roles.
- Deploying J2EE applications through a wizard that allows deployment to a single OC4J container or cluster and automates mapping URL paths, data sources, selecting a user manager and mapping security roles

- Browsing and maintaining applications including installing application updates or patches, and modifying application configuration.
- Monitoring the availability, usage, and performance of all applications to determine request volume, longest response time, and additionally, for Servlets and JSPs load time, service time and overhead.

About Security

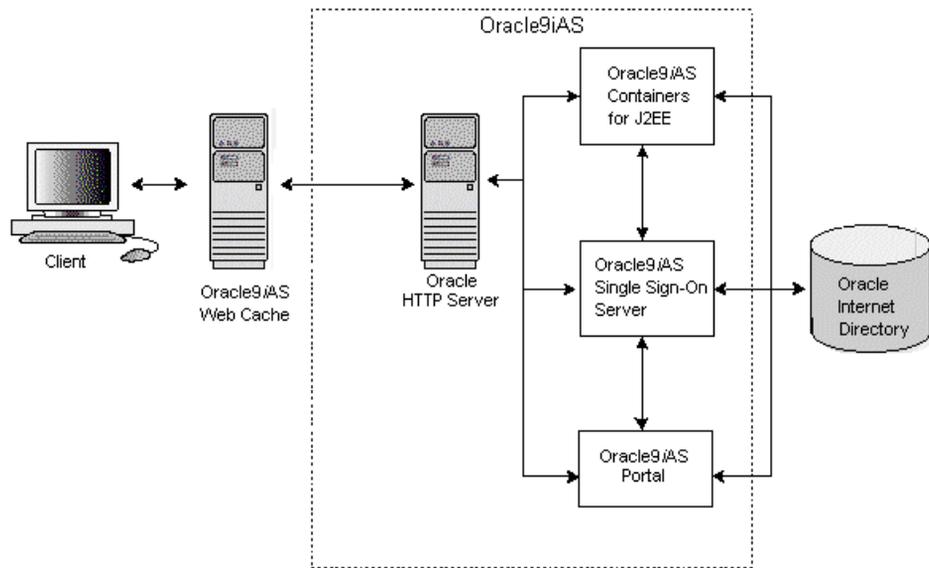
Oracle9i Application Server provides a comprehensive integrated security framework supporting all its components, as well as third party and custom application deployed on Oracle9iAS. The framework is based on Oracle9iAS Single Sign-On for authentication, the Java Authentication and Authorization Service (JAAS) for security services in J2EE, and Oracle Internet Directory for authorization and user provisioning.

See Also: *Oracle9i Application Server Security Guide*

[Figure 8-2](#), illustrates how the elements of Oracle9i Application Server function together. Following is the functionality of the various components:

- **Oracle9iAS Web Cache** is positioned in front, where it provides efficiency.
- **Oracle HTTP Server** supplies the basic Web services which are provided by Apache.
- **Oracle9iAS Portal** provides the infrastructure, including the ability to create and manage Web pages. It lets you display multiple Web pages on each Web page, with links to content through Java applications.
- **Java engine** lies underneath Oracle9iAS Web Cache and Oracle HTTP Server, supporting their ability to link efficiently.
- **Oracle9iAS Single Sign-On** enables users to login to Oracle9iAS and gain access to those applications for which they are authorized, without requiring them to re-enter a user name and password for each application.
- **Oracle Internet Directory** serves the rest of the stack by providing authentication, and a centralized user model where by users can be created and managed on an enterprise scale.

Figure 8–2 Security Architecture of Oracle9i Application Server



Security Features

Oracle9iAS includes the following security features that enable you to secure your Oracle9iAS framework:

- [Oracle9iAS Single Sign-On](#)
- [Java Authentication and Authorization Service \(JAAS\)](#)
- [Oracle Internet Directory](#)

Oracle9iAS Single Sign-On

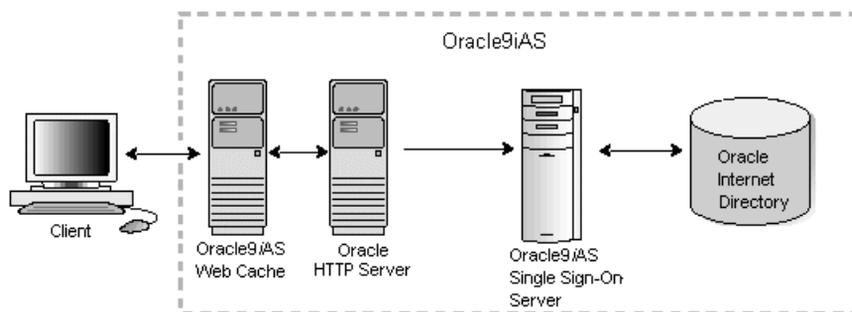
An important security feature of Oracle9iAS is support of single signon (SSO) to Web-based applications. Oracle9iAS Single Sign-On addresses the problem of “too many passwords”. With the rapid growth of the Internet, this problem has been increasingly prevalent, causing users inconvenience that typically results in poor security practices, and increased administrative costs.

Oracle9iAS Single Sign-On resolves this problem by enabling users to login to Oracle9iAS and gain access to those applications for which they are authorized, without requiring them to re-enter a user name and password for each application.

It is fully integrated with Oracle Internet Directory, which stores user information. It supports LDAP-based user and password management through OID.

[Figure 8-3](#) shows the architecture of Oracle9iAS Single Sign-On.

Figure 8-3 Oracle9iAS Single Sign-On Architecture



Oracle9iAS Single Sign-On provides the following functionality:

- Authenticates users and passes their identities securely to partner applications, such as Oracle9iAS Portal. It prompts users for a username and password when they access the system for the first time in a given time period, and verifies the password presented by the user.
- Uses cookies, which are formatted pieces of information stored on a browser client by a Web server. Cookies allow Web servers to store and retrieve information about the client user, effectively maintaining client state information in the otherwise stateless Web environment.
- Supports Public Key Interface (PKI) client authentication, which enables PKI authentication to a wide range of Web applications. By means of an API, Oracle9iAS Single Sign-On can integrate with third-party authentication mechanisms such as Netegrity Site Minder.

With Oracle9iAS Single Sign-On, users typically sign on to a centrally administered Single Sign On Server through a central Web portal. Once it authenticates the user, Single Sign On Server displays links to all the applications for that user.

Using a central Web portal with a centrally administered Single Sign On Server has these advantages:

- **Convenience:** The user enters the user name and password only once, at a central corporate Web portal, to access all the needed applications. From the user's perspective, authentication to each application happens transparently.
- **Increased Security:** Fewer user name and password combinations lowers the risk of unauthorized access to a user's restricted information.
- **Ease of Administration:** Oracle9iAS Single Sign-On provides centralized provisioning of user accounts, so that administrators can easily create new user accounts. Centralizing the authentication process also makes it possible to support additional authentication mechanism in a localized manner. For example, you can implement an LDAP-based authentication, or digital certificate-based authentication, and the change would be localized to the SSO Server.

Partner and External Applications

There are two kinds of applications to which Oracle9iAS Single Sign-On provides access:

- Partner Applications
- External Applications

Partner applications are integrated with the SSO Server. They contain an Oracle9iAS Single Sign-On API that enables them to accept a user's identity as validated by the SSO Server.

External applications are Web-based applications that retain their authentication logic. They do not delegate authentication to the SSO Server and, as such, require a user name and password to provide access. Currently, these applications are limited to those which employ an HTML form for accepting the user name and password. The user name may be different from the SSO user name, and the SSO Server provides the necessary mapping.

The SSO offering in Oracle9iAS is a critical differentiator for users seeking a robust, fully integrated SSO architecture. Oracle9iAS leverages JAAS and LDAP capabilities of Oracle Internet Directory, to deliver a comprehensive end-to-end security infrastructure across the entire Oracle9iAS product.

Java Authentication and Authorization Service (JAAS)

Oracle9i Application Server provides an implementation of Java Authentication and Authorization Service (JAAS) that integrates with the Oracle9iAS J2EE security infrastructure to enforce security constraints for Web (servlets and JSPs) and EJB components.

JAAS support provides the following benefits:

- Integrates Java-based applications with Oracle9iAS Single Sign-On, includes authentication, thereby giving you extensible security for Java-based applications.
- Manages access control policies centrally in Oracle Internet Directory, controls access by role, and partitions security policy by subscriber.
- Supports impersonation of a specific user, allows an enterprise bean, servlet, or JSP to run with the permissions associated with the current client or a specified user.

Oracle Internet Directory

Oracle Internet Directory (OID) is a critical component of Oracle9iAS management and security infrastructure. It ensures that user accounts and groups are managed centrally through the LDAP Version 3 standard. Oracle9iAS enables users to be created centrally in OID and shared across all components in Oracle9iAS. When users log in, they are authenticated once by Oracle9iAS Single Sign-On against their OID credentials, and can thereby access multiple applications seamlessly.

Self-Service Console

Oracle Internet Directory includes a Self-Service Console, an easy-to-use, Web-based interface which allows end-users and application administrators to search for and manage data in the directory. Combined with the new Delegated Administration Service, this console provides Oracle9iAS with means of provisioning end-users in the Oracle9iAS environment. OID also enables components of Oracle9iAS to synchronize data about users and group events, so that those components can update any user information stored in their local application instances.

Password Management

Oracle Internet Directory provides users with a very directory searches through sophisticated server-side caching capabilities. OID also provides two key features that ensure administrators can deliver seamless directory services to all users:

- **Alias De-reference:** When a user or an application searches on an alias, OID automatically de-references the alias and returns the entry to which it refers. This feature enables administrators to change the names of objects in ways that are transparent to users and applications.
- **Enhanced Proxy Capabilities:** Administrators can safely establish performant, auditable middle-tier application access to the directory on behalf of end user communities.

Synchronization with Third Party LDAP Servers

The Oracle Directory Integration Platform enables customers to synchronize data between various directories and Oracle Internet Directory. The Oracle Directory Integration Platform is a set of services and interfaces which make it possible to develop synchronization solutions with third party metadirectories and other enterprise repositories, such as iPlanet. With Oracle9iAS, Oracle Internet Directory includes an agent for out-of-the-box synchronization with Oracle Human Resources and an agent for synchronizing information with selected third party LDAP servers.

Oracle Internet Directory also provides a plug-in framework for applications that require customized functionality, such as referential integrity of data. The plug-in framework is delivered as a highly-flexible PL/SQL interface, allowing user-defined operations to be invoked by the directory server before or after LDAP commands.

Key Directory Features

Oracle Internet Directory provides the following key directory features:

- Native LDAP v3 server supporting all LDAP2000-compliant RFCs, including LDAP v2 and v3 RFCs
- Supports the X.500 information, naming, and storage model
- Extensible directory schema for online modifications with no downtime
- LDAP developers APIs in Java, C, and PL/SQL to assist with application development.

Using Oracle Internet Directory with Middle Tier Components

The middle tier components use Oracle Internet Directory in the following ways:

- Application server instances and infrastructures store security and management information in Oracle Internet Directory. Oracle Internet Directory stores users information, such as user names and privileges, required for internal operation of the application server.
- JAAS stores realms and JAAS policy in Oracle Internet Directory.
- Oracle9iAS Single Sign-On validates the user name and password against user and group profiles stored in Oracle Internet Directory.

See Also: *Oracle Internet Directory Administrator's Guide*

Glossary

A2A

A2A integration (also known as enterprise application integration) is the integration of applications and business processes within the same company (also known as an enterprise).

access control list (ACL)

See: [ACL](#)

access control list model

A method for organizing authorization information. The access control list model is resource-centered or object-centered. This means that authorization information is associated with objects.

ACID characteristics

A [transaction](#) has ACID characteristics (Atomic, Consistent results, Isolated, and Durable), if it demonstrates the following:

- If interrupted by failure, all effects are undone or rolled back (Atomic).
- The effects of a transaction preserve invariant properties (Consistent results).
- Its intermediate states are not visible to other transactions. Transactions appear to execute serially, even if they are performed concurrently (Isolated).
- The effects of a completed transaction are persistent; they are never lost (except in a catastrophic failure) (Durable).

ACL

Access Control List. A list of groups and users authorized for specific access to an object. Or a list of entities, together with their access rights, which are authorized to have access to a specified resource.

Apache

A public domain HTTP server derived from the National Center for Supercomputing Applications (NCSA).

Apache Module

An add-on to either **Oracle HTTP Server** or **Apache**. Modules extend the basic functionality of the Web server and support integration between the **Oracle HTTP Server** and other Oracle9i Application Server components.

API

Application Program Interface. A set of exposed data structures and functions that an application can use to invoke services on a component.

applet

A Java program that runs from an applet viewer or a Web page.

application integration

The ability to link different types of enterprise applications and business processes together so that they can smoothly and effectively communicate to conduct e-business. These applications can reside within a company's enterprise boundaries or across multiple company boundaries connected over the Internet.

application-to-application

See: [A2A](#)

authentication

Authentication ensures that access to static pages, CGI scripts, and applications is limited to authorized users. When access to an application or document is protected by an authentication scheme, the client sends identification information to the server, which checks if the client is authorized to access the object.

See Also: [security scheme](#), [restriction](#)

authorization

The evaluation of security constraints to send a message or make a request. Authorization uses specific criteria to determine whether the request should be permitted. The criteria are authentication and restriction.

See Also: [authentication](#), [restriction](#)

authentication broker

The portion of the authentication server that responds to and evaluates authorization requests.

authentication provider

An object that specifies all of the realms used to implement a particular security scheme. An authentication provider is a code module that runs within the authentication server and implements a particular security scheme.

authentication server

An object that encapsulates the authentication performed against applications. An authentication server consists of one authentication broker object and several authentication provider objects.

automatic deployment

A method for re-deploying applications, servlets, or JSPs after changes have been made to the application code that does not require updating the server configuration files or restarting the application server. With automatic deployment, the application server detects the changes in code and automatically re-deploys the applications.

See Also: [hot deployment](#) for comparison

availability

The percentage or amount of scheduled time that a computing system provides application service.

B2B

B2B integration is the integration of a company's applications and business processes with external business systems operated by customers, suppliers, trading partners, exchanges, and marketplaces. B2B extends A2A integration to the Internet.

basic authentication

A username/password-based authentication scheme that does not encrypt passwords when sending them over the Internet.

See Also: [digest authentication](#), [authentication](#)

bean-managed persistence (BMP) bean

A Java bean that stores all state information within itself.

browser client

A client that can access static pages, CGI scripts, and applications via a URL over HTTP, HTTPS, or IIOP.

business intelligence

Information describing your business, data, and Web site traffic.

business-to-business

See: [B2B](#)

CA

Certifying Authority. A trusted third party that vouches for the identity of an individual, company, or server and signs a certificate.

See Also: [trustpoint](#)

caching

The act of storing information that is frequently accessed in a location where it can be accessed quickly. For example, [Oracle9iAS Web Cache](#) stores dynamically generated Web pages locally and serves them to incoming requests. This reduces the total time spent handling the request by avoiding costly connections to the back-end database and other Web site bottlenecks.

capability model

A method for organizing authorization information. The capability model is principal-centered or subject-centered. This means that authorization information is associated with subjects. The [Java 2 Security Model](#) is an example of a capability-based system.

certificate

A specially formatted data item signed by a trusted third party to attest to the validity of the item's information. Public-key certificates use a **CA**'s signature to attest that the enclosed public key belongs to the principal identified by the enclosed name.

certificate authentication

An authentication method in which **clients** identify themselves using X.509 v3 certificates.

certificate authority

See: **CA**

certificate revocation list

See: **CRL**

CGI

Common Gateway Interface. 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.

character set

A set of characters used to write one or more human languages, as defined by **RFC 1521**.

clickstream

A virtual trail that a user leaves behind while moving from place to place on the Internet. A clickstream is a record of a user's activity on the Internet, including every Web site and every page of every Web site that the user visits, how long the user was on a page or site, in what order the pages were visited, any newsgroups that the user participates in and even the e-mail addresses of mail that the user sends and receives.

client

A user, software application (such as a browser), or computer that requests the services, data, or processing of another application or computer (the **server**).

cluster

A cluster is a collection of application server instances with identical configuration and application deployment. Clusters enforce homogeneity between member instances so that a cluster of application server instances can appear and function as a single instance. With appropriate front-end load balancing, any **instances** in an application server cluster can serve client requests. This simplifies configuration and deployment across multiple instances and enables fault tolerance among clustered instances.

clustering

Collecting Oracle9iAS **instances** into *clusters* for load balancing and fault tolerance. With clustering, the collection of application server instances are treated as a pool to service incoming requests. If one instance does not respond, then the request is forwarded to another instance in the cluster. The load balancer also maintains session context, so that when a client reconnects, its request is sent to the application server instance that was previously serving it.

common gateway interface

See: [CGI](#)

common object request broker architecture

See: [CORBA](#)

connect string

The set of parameters, including a protocol, that is used to connect to a specific database instance on the network. Other names for a connect string include: SQL*Net V2 service name and Net8 connect string.

container

A component that contains other components, such as a servlet. A container executes and manages a servlet. A container is either part of or associated with and used by a Web server. When a client HTTP request calls a servlet, the Web server passes the HTTP request to the container. The container translates the HTTP request into a Java method invocation and then passes the request to the servlet.

See Also: [JSP translator](#), [servlet container](#)

container-managed persistence (CMP) bean

A Java bean that stores all state information in the container.

cookie

A text string that is stored onto the client browser by the server to maintain state between [HTTP](#) calls. Cookies enable applications to store and retrieve information about a client, such as the domain, path, lifetime, and other variables. Cookies can either expire when the user exits the browser or at a date specified by the creator of the cookie.

CORBA

Common Object Request Broker Architecture. An industry standard for allowing code modules called “objects” to communicate with one another regardless of the programming language in which they are written or the operating system on which they are running.

CORBA object

This is a generic term for a server-side program that conforms to the OMG’s [CORBA](#) specification. Objects can be written in any language, deployed on any machine, and can exist locally or over a wide-area network.

CRL

Certificate Revocation List. A list of [certificates](#) that have been revoked before their scheduled expiration date. CRLs only list revoked certificates. When a revoked certificate is past its original expiration date, it is removed from the CRL.

customer database

The original and primary database for storing your data. The customer database is commonly located on the database server tier in a [three-tier architecture](#).

customer relationship management (CRM)

The methodologies, software, and usually Internet capabilities that help an enterprise manage customer relationships in an organized way.

DAD

Database Access Descriptor. 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 NLS

parameters such as NLS language, NLS date format, NLS date language, and NLS currency.

database access descriptor

See: [DAD](#)

default DAD

The database access descriptor that a PL/SQL application uses when the configuration information in the application does not specify a DAD.

default MIME type

The MIME type the [HTTP listener](#) uses to interpret requested files of an unsupported MIME type.

See Also: [MIME type](#)

demilitarized zone

See: [DMZ](#)

deployment descriptors

A J2EE application and each of its modules has its own deployment descriptor (DD). A DD is an [XML](#) text-based file with an `.xml` extension that describes the deployment settings for a component. For example, an enterprise bean module DD declares transaction attributes and security authorizations for an enterprise bean. DD information is declarative so it can be changed without modifying the bean source code. At runtime, the J2EE server reads the DD and acts on the component accordingly.

digest authentication

An authentication scheme that does not send passwords over the Internet. Digest authentication is safer than basic authentication, but is not supported by most browsers.

See Also: [authentication](#), [basic authentication](#)

digital signature

A digital code attached to an electronic document that reliably identifies the author or sender, and verifies that the document has not been tampered with.

directory server

Defines a hierarchical view of an organization's employees, units, and other resources. You can protect applications using directory servers by limiting access to the virtual paths of the applications to particular branches in the directory server.

distinguished name

The unique name of an LDAP-based directory entry. A distinguished name comprises all of the individual names of the parent entries back to the root.

distributed transaction processing

See: [DTP](#)

DMZ

Demilitarized Zone. The area between outer and inner firewalls. It is normally used to protect the internal application servers from being attacked by those attempting to gain unauthorized access to a network or intranet.

document type definition

See: [DTD](#)

domain-based restriction

A restriction scheme that allows or denies access to files based on the client machine's domain name.

See Also: [restriction](#), [IP-based restriction](#)

DTD

Document Type Definition. A set of rules that define the allowable structure of an [XML](#) document. DTDs are text files that derive their format from SGML and are either embedded within an XML document or referenced by an XML document..

DTP

Distributed Transaction Processing. The protocol that guarantees a two-phase commit when multiple databases are involved in a [transaction](#).

domain name system (DNS)

The mechanism that divides the Internet into separate, hierarchical groups called domains, identified by unique alphanumeric names, such as `us.oracle.com`.

DNS identifies each computer within a domain by a unique hostname. For example, a computer named `hal` in the `us.oracle.com` domain would be uniquely identified on the Internet as `hal.us.oracle.com`.

dynamic link library (DLL)

An archive of executable functions or data that can be used by a Microsoft Windows application. Typically, a DLL provides one or more particular functions and a program accesses the functions by creating either a static or dynamic link to the DLL. A DLL can be used by several applications at the same time.

e-business

The conduct of business over the Internet, such as buying and selling products, servicing customers, and collaborating with business partners.

EAR files

A J2EE application with all of its modules is delivered in an Enterprise Archive (EAR) file. An EAR file is a standard JAR file with an `.ear` extension. An EAR file contains the JAR and WAR files that comprise a J2EE application.

EIS

Enterprise Information Systems. Systems that provide the information infrastructure for an enterprise. Enterprises run their businesses using the information stored in these systems. Examples of enterprise information systems include enterprise resource planning systems, mainframe transaction processing systems, relational database management systems, and other legacy information systems. Enterprise applications require access to applications running on enterprise information systems.

EJB

Enterprise JavaBeans. The component-based application model for Java defined by JavaSoft. It provides most of the system level services such as multi-threading to ease application programming. EJB relies on various standardized enterprise services, such as **JNDI**, **JTS**, **JDBC**, etc. to facilitate application programming and enable EJB objects to be interoperable across various EJB servers. It fulfills the write once, run anywhere paradigm.

EJB application

A framework of deploying **CORBA objects** written in Java, which adhere to the EJB specification.

EJB container

The component coordinator in an EJB application and one of the key EJB runtime components.

See Also: [JNDI](#), [EJB application](#)

EJB deployment descriptor

A serialized object that provides information, such as transaction and security policies, about how an [EJB application](#) or object should be deployed.

encapsulation

The mechanism that binds code together with the data that it manipulates. Encapsulation provides a wrapper that keeps both the code and the data safe from outside intervention.

encoding

An algorithm used to alter a file's format, such as compression.

encryption

The practice of encoding (encrypting) data in such a way that only an intended recipient can decode (decrypt) and read the data.

See Also: [public-key encryption](#), [shared secret-key encryption](#)

enterprise beans

Enterprise beans are server-side components that encapsulate the business logic of an application, which is the code that fulfills the purpose of the application.

enterprise information systems

See: [EIS](#)

Enterprise JavaBeans

See: [EJB](#)

enterprise resource planning

The set of activities supported by multi-module application software that helps a manufacturer or other business manage important parts of their business, including

product planning, parts purchasing, inventory management, supplier interaction, customer service, and order tracking.

entity beans

An entity bean is a complex business entity. An entity bean models a business entity or models multiple actions within a business process. Entity beans are often used to facilitate business services that involve data and computations on that data. For example, an application developer might implement an entity bean to retrieve and perform computation on items within a purchase order. Your entity bean can manage multiple, dependent, persistent objects in performing its necessary tasks.

failover

The ability to reconfigure a computing system to utilize an alternate active component when a similar component fails.

failure recovery

A system of failure detection and recovery. Components monitor each other continuously. When a component fails, Oracle9i Application Server detects the failure and restarts the failed component, restoring any preserved state information when possible.

farm

A collection of **clusters** and **instances** that share the same **Oracle9iAS Infrastructure**.

fault tolerance

The ability of a computing system to withstand faults and errors while continuing to provide the required services.

file protection

The practice of assigning an authentication or restriction scheme by controlling access to a specific file or group of files.

filename extension

A short alphanumeric suffix attached to a filename following a dot “.” that represents the file’s format. Oracle9i Application Server uses filename extensions to identify several kinds of file formats, including MIME types and encodings.

firewall

A machine that acts as an intermediary to protect a set of computers or networks from outside attack. It regulates access to computers on a local area network from outside, and regulates access to outside computers from within the local area network. A firewall can work either by acting as a proxy server that forwards requests so that the requests behave as though they were issued by the firewall machine, or by examining requests and attempting to eliminate suspect calls.

graphical user interface (GUI)

Sometimes referred to as UI (user interface), GUI is the graphical, as opposed to purely textual, user interface to a computer. Elements of the GUI include pull-down menus, buttons, icons, windows, and graphics.

host

A computer with a unique domain name.

hosted applications

Applications typically developed by independent software vendors and administered by application service providers. These applications are accessed on an external Web site that enables multiple companies to utilize the applications.

hosted services

Services offered by application service providers for multiple subscribers.

hosted environment

An application deployment environment in which multiple customers and companies subscribe to shared services.

hostname

A character string that uniquely identifies a computer within a DNS domain.

hot deployment

A method for deploying new applications that does not require restarting the application server. With hot deployment, changes made to the server configuration files are automatically detected, and you can deploy new applications without restarting the server.

See: [automatic deployment](#) for comparison

HTTP

The underlying format, or protocol, 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 Oracle9i Application Server and clients.

HTTP server

A server which receives HTTP requests from remote browsers, converts the requested URL to a filename, and returns the file to the requester.

HTTP header

A body of information that a browser sends along with a URL when requesting a Web page. It includes such information as the browser type and MIME types.

HTTP listener

See: [listener](#)

HTTP request information

Information requested by a client in the form of an HTTP header.

HTTP response information

Information supplied by an [HTTP listener](#) or an application.

HTTPS (secure HTTP)

A version of HTTP with provisions for secure data transmission.

See Also: [HTTP](#)

hypertext markup language (HTML)

A format for encoding hypertext documents that may contain text, graphics, and references to programs and other hypertext documents.

hypertext transfer protocol (HTTP)

See Also: [HTTP](#)

IDE

See: [integrated development environment \(IDE\)](#)

IBAC

Identity-Based Access Control. The use of digital IDs to control access to a resource.

instance

An application server instance is the set of processes required to run the configured components within an application server installation. There can be only one application server instance per application server installation. The terms *installation* and *instance* are sometimes used interchangeably, however, it is important to remember that an installation is the set of files installed into an Oracle home and an instance is a set of processes associated with those files.

integrated development environment (IDE)

A visual tool containing editors, debuggers, screen painters, object browsers etc.

identity-based access control

See: [IBAC](#)

IDL

Interface Definition Language. A standard language for interface specification primarily used for CORBA object interface definition. IDL is declarative and does not reveal the implementation of a CORBA object. CORBA defines standard mappings from IDL to various programming languages.

IIOp

Internet Inter-ORB Protocol. An Internet transport protocol used by CORBA objects to communicate with each other. In the context of Oracle9i Application Server, IIOp is used by ECO/Java and EJB objects. IIOp is also used between Oracle9i Application Server components.

inheritance

The process by which one class acquires the methods and properties of another in object-oriented programming methodology

instance

An application server instance is the set of processes required to run the configured components within an application server installation. There can be only one application server instance per application server installation. The terms *installation* and *instance* are sometimes used interchangeably, however, it is important to

remember that an installation is the set of files installed into an Oracle home and an instance is a set of processes associated with those files.

interface definition language

See: [IDL](#)

Internet inter-ORB protocol

See: [IIOP](#)

interoperable object reference

See: [IOR](#)

Internet Information Server (IIS)

Microsoft's Web server that runs on Windows NT/2000 platforms.

IOR

Interoperable Object Reference. This is a unique string for each CORBA object and is created when an object reference is passed among different ORBs. You can pass this string to a method to determine the actual object reference.

See Also: [object reference](#)

IP address

A four-part number separated by periods that uniquely identifies a computer on the Internet; the number format is defined by the Internet Protocol (IP).

IP-based restriction

A restriction scheme that allows or denies access to files based on the client machine's IP address.

See Also: [restriction](#)

J2EE

Java 2 Platform, Enterprise Edition. This platform enables application developers to develop, deploy, and manage multi-tier, server-centric, enterprise level applications.

J2EE Server

The runtime portion of a J2EE product, which provides EJB containers, Web containers, or both. J2EE servers are usually located in the middle tier of a three-tier architecture.

J2SE

Java 2 Platform, Standard Edition. This platform enables application developers to develop, deploy, and manage Java applets and applications on a desktop client platform such as a personal computer or workstation.

JAAS

Java Authentication and Authorization Service. A Java package that enables services to authenticate and enforce access control upon users. JAAS implements a Java version of the standard Pluggable Authentication Module (PAM) framework, and extends the access control architecture of the [Java 2 Security Model](#) to support user-based authorization.

JAR files

There are different types of Java Archive (JAR) files. An EJB JAR file contains its [deployment descriptors](#) (DD), related files, and the `.class` files for the enterprise bean. An application client JAR file contains its DD, related files, and the `.class` files for the application client. JAR files are packaged together with WAR files into EAR files.

Java

A language developed by Sun Microsystems. This language is fully object-oriented, extremely portable, and optimized for creating distributed applications on the Internet or other computer networks.

Java Authentication and Authorization Service

See: [JAAS](#)

JavaBeans

A portable platform-independent component model that enables developers to write reusable components once and run them anywhere.

See Also: [Enterprise JavaBeans](#)

JavaServer Pages

See: [JSP](#)

Java interpreter

A program that interprets and executes Java bytecode independently of a Web browser.

Java Naming and Directory Interface

See: [JNDI](#)

Java Transaction Service

See: [JTS](#)

Java Virtual Machine

See: [JVM](#)

Java 2 Platform, Enterprise Edition

See: [J2EE](#)

Java 2 Platform, Standard Edition

See: [J2SE](#)

Java 2 Security Model

Provides developers and administrators with increased control over many aspects of enterprise applet, component, servlet, and application security. The Java 2 Security Model is capability-based and enables you to establish protection domains, and set security policies for these domains.

Permissions are the basis of the Java 2 Security Model. All Java classes (whether run locally or downloaded remotely) are subject to a configured security policy that defines the set of permissions available for those classes. Each permission represents a specific access to a particular resource.

Javadoc

A tool for generating API documentation in HTML format from documentation comments in Java source code. These HTML pages describe the classes, inner classes, interfaces, constructors, methods, and fields.

JDBC

Java DataBase Connectivity. A Java package that provides connectivity to databases from within Java.

JIT compilation

Just-in-time compilation is the process by which the Java Virtual Machine keeps a copy of native code that it generates from bytecode the first time a method is encountered. Subsequently, when the method is run, the JIT uses the native code without having to interpret the method, resulting in a boost in performance.

JNDI

Java Naming and Directory Interface. JNDI consists of a standard set of APIs that provide directory and naming services. Oracle9i Application Server has a JNDI naming server which clients can use to obtain object references to ECO/Java objects or Enterprise JavaBean objects.

JServ process

An open source servlet engine that can communicate with Oracle HTTP Server by configuring it to accept a servlet request from mod_jserv, which is an Oracle HTTP Server plug-in.

JSP

JavaServer Page. An extension to the servlet functionality that enables a simple programmatic interface to Web pages. JSPs are HTML pages with special tags and embedded Java code that is executed on the Web or application server providing dynamic functionality to HTML pages. JSPs are actually compiled into servlets when first requested and run in the [servlet container](#).

JSP translator

A JSP translator is an entity that translates, executes, and processes JSP pages and delivers requests to them. The exact architecture of a JSP translator varies from implementation to implementation, but it consists of a servlet or a collection of servlets. The JSP translator, therefore, is executed by a [servlet container](#).

JSP engine

See: [JSP translator](#)

JSP tag

Tags that are used in JavaServer Pages (JSP). These tags use the `<jsp:` syntax and enclose action elements in the JSP with "begin" and "end" tags similar to [XML](#) statements.

JTS

Java Transaction Service. It provides the services necessary for applications and databases to become part of a transaction. JTS is the Java version of [OTS](#).

just-in-time compilation

See: [JIT compilation](#)

JVM

Java Virtual Machine. A virtual machine is an abstract specification for a computing device that can be implemented in different ways, in software or hardware. You compile to the instruction set of a virtual machine much like you would compile to the instruction set of a microprocessor.

The Java Virtual Machine is part of the Java runtime environment responsible for interpreting Java bytecode. It consists of a bytecode instruction set, a set of registers, a stack, a garbage-collected heap, and an area for storing methods. Java bytecode is executable by any JVM running on any machine.

key pair

A pair of mathematically related keys (a [public key](#) and a [private key](#)) associated with a user and used in public-key encryption.

language identifier

A two-character alphanumeric string that identifies a human language, as defined by [RFC 1766](#).

LDAP

Lightweight Directory Access Protocol. A protocol that allows clients to access information from a directory server. This protocol enables corporate directory entries to be arranged in a hierarchical structure that reflects geographic and organizational boundaries.

light weight directory access protocol

See: [LDAP](#)

listener

Listeners are HTTP servers; they handle incoming requests and route them to the dispatcher.

Login Module

The authentication modules configured for a particular application. The `LoginContext` class decouples the application code from the authentication services, and different login modules can be plugged in under an application without affecting the application code.

Login Server

Authenticates the username and password of a client user attempting to access an application. Once authenticated, the Login Server passes the client's identity to various applications. An encrypted login cookie identifies the client as being authenticated. The Login Server provides sign-on (SSO) authentication. This enables a user to access multiple accounts and applications with a single username and password.

manifest file

A text file that describes the contents of a JAR file. For EJB JAR files, the manifest file identifies it as an "ejb-jar" file. The manifest file also indicates which elements in the JAR file are the EJB deployment descriptors for the components to be deployed.

message

The smallest unit of information inserted into and retrieved from a queue. A message consists of the following:

- Control information (metadata)
- Payload (data)

The control information represents message properties used by the queue to manage messages. The payload data is the information stored in the queue. A message can reside in only one queue.

message-driven beans

Message-driven beans (MDB) provide an easier method to implement asynchronous communication than using straight JMS. MDBs were created to receive

asynchronous JMS messages. The container handles much of the setup required for JMS queues and topics. It sends all messages to the interested MDB.

metric

A performance statistic, such as uptime or queue.

MIME type

Multipurpose Internet Mail Extension type. A message format used on the Internet to describe the contents of a message and defined by the MIME standard. MIME is used by HTTP servers to describe the type of content being delivered. Several [RFCs](#) define MIME.

See Also:

<http://www.oac.uci.edu/indiv/ehood/MIME/MIME.html>

mods

See: [Apache Module](#)

mod_access

A module, or plug-in, of Oracle HTTP Server that provides access control based on client host name or IP address.

mod_osso module

Provides communication between the SSO-enabled Login Server and the Oracle HTTP Server listener.

module

See: [Apache Module](#)

multiport

A single listener that can respond to requests directed at more than one address/port combination. A multiport can be configured to respond to requests differently if they are made to a different address/port combination on the same listener.

Multipurpose Internet Mail Extensions

See: [MIME type](#)

node

See: [host](#)

noun

In the scope of performance monitoring tools, a noun is a component of an Oracle9i Application Server site. Examples of a noun are component, object, class, bean, process, piece of code, computer, process, etc.

national language support (NLS)

The set of mechanisms used to translate data between various languages and character sets.

Object Management Group

See: [OMG](#)

object-oriented programming

A method of programming that organizes a program around its data (objects) and a set of well-defined interfaces to that data. It is a revolutionary new way of looking at computer programming. A generalization of data object along with its possible data variables and methods (what to do with variables) in a class of data objects.

object reference

A unique identifier that is used to represent an object instance in a distributed system.

See Also: [IOR](#)

Object Transaction Service (OTS)

See: [OTS](#)

OC4J

See: [Oracle9iAS Containers for J2EE \(OC4J\)](#)

OCI

Oracle Call Interface. An API or low-level tool to access Oracle databases and execute SQL and PL/SQL statements.

object

Generally, any item that can be individually selected or manipulated. This includes buttons and pull-down menus. In object-oriented programming, an object is a self-contained entity that consists of both data and procedures to manipulate the data.

OMG

Object Management Group. A consortium with a membership of more than 700 companies. The organization's goal is to provide a common framework for developing applications using object-oriented programming techniques. OMG is responsible for the CORBA specification.

See Also: <http://www.omg.org>

ORACLE_HOME

An environment variable (UNIX) or registry key (Windows) that indicates the directory where Oracle software is installed.

Oracle9i platform

A server deployment platform consisting of Oracle9i Application Server, Oracle9i Database Server, and Oracle9i Developer Suite.

Oracle9iAS Containers for J2EE (OC4J)

A complete set of **J2EE** containers written entirely in Java that execute on the **Java Virtual Machine** (JVM) of the standard Java Development Kit (JDK).

Oracle9iAS Discoverer

Oracle9iAS Discoverer is a **business intelligence** tool for analyzing data. Using Oracle9iAS Discoverer's award-winning user interface, users can access and analyze database data. There are two Oracle9iAS Discoverer products:

- Oracle9iAS Discoverer Plus is the Internet version of the award-winning Windows version of Discoverer. With Discoverer Plus, business professionals can get and analyze data in a company's database without having to understand complex database concepts. Using Wizard dialogs and menus, Discoverer Plus guides users through the steps to get and analyze data to support their business decisions.
- Oracle9iAS Discoverer Viewer is a tool for viewing workbooks created by Discoverer Plus. Oracle9iAS Discoverer Viewer can also be used to integrate database output into a Web site and portal. It is easy to customize Discoverer

Viewer to both conform to a particular Web site look and feel and to build custom Discoverer applications for the Web. Discoverer Viewer is optimized for performance and designed to minimize network traffic.

Oracle9iAS Forms Services

Oracle9iAS Forms Services deploys Forms applications with database access to Java clients in a Web environment. Oracle9iAS Forms Services automatically optimizes class downloads, network traffic, and interactions with Oracle database. Applications are automatically load-balanced across multiple servers and, therefore, can easily scale to service any number of requests.

Oracle9iAS Infrastructure

A single, consistent combination of the **Oracle9iAS Metadata Repository**, a directory server, management server, and single signon server. Most Oracle9iAS components use the infrastructure to provide a more integrated environment for application developers and system managers.

Oracle9iAS InterConnect

The integration hub that coordinates the communication and transformation of messages between two or more heterogeneous applications. Oracle9iAS InterConnect defines business events, their associated data, and any transformations required to map one application's view of a business object to another's view.

Oracle9iAS InterConnect Adapters

Provides a set of connectivity adapters that enable third-party applications and technology environments to participate in integration.

Oracle9iAS Metadata Repository

A pre-seeded database containing metadata required by Oracle9i Application Server **instances**.

Oracle9iAS Portal

Oracle9iAS Portal is a complete solution for building, deploying and monitoring Web database applications and content-driven Web sites. Oracle9iAS Portal enables you to create and view database objects through an easy-to-use HTML-based interface, and provides tools for creating HTML-based interfaces. It also allows you

to resolve performance problems using performance tracking facilities, and enables you to manage database security through its interface.

See Also: [portal](#), [portlet](#)

Oracle9iAS Portal Login Server

See: [Login Server](#)

Oracle9iAS Reports Services

Oracle9iAS Reports Services provides an easy-to-use, scalable, and manageable solution for high-quality database publishing and reporting by creating dynamic reports for the Web and across the enterprise. It enables you to implement a multi-tiered architecture for running your reports.

Oracle9iAS Unified Messaging

Oracle9iAS Unified Messaging provides access to all message types using any access method. This means that e-mail, fax, voice mail, and any other mail type are available to users through the access channel and device of their choice.

Oracle9iAS Web Cache

Oracle9iAS Web Cache is a server accelerator caching service that improves the performance, scalability, and availability of frequently used e-business Web sites that run on the Oracle9i platform. By storing frequently accessed URLs in virtual memory, Oracle9iAS Web Cache eliminates the need to repeatedly process requests for those URLs on the Web server, and it caches both static and dynamically-generated HTTP content from one or more applications Web servers.

Oracle9iAS Wireless

Oracle9iAS Wireless is a portal service for delivering information and applications to mobile devices. Using Oracle9iAS Wireless, you can create custom portal sites that use different kinds of content, including Web pages, custom Java applications, and **XML**-based applications. Oracle9iAS Wireless sites make this diverse information accessible to mobile devices without you having to rewrite the content for each target device platform.

Oracle Advanced Queuing

Provides database-resident messaging capabilities that enable reliable, asynchronous communication between applications. Oracle9iAS integration uses

Oracle Advanced Queuing as the communication mechanism between adapters, participating applications, and Oracle Workflow.

Oracle Advanced Security

Oracle Advanced Security provides a comprehensive suite of security features to protect enterprise networks and securely extend corporate networks to the Internet. It provides a single source of integration with network encryption and authentication solutions, single signon services, and security protocols. By integrating industry standards, it delivers unparalleled security to the Oracle network and beyond.

Oracle Business Components for Java (BC4J)

Oracle Business Components for Java is a 100% Java-compatible, [XML](#)-powered framework that enables productive development, portable deployment, and flexible customizations of multi-tier database applications from business components.

Oracle Call Interface (OCI)

See: [OCI](#)

Oracle Database Client Developer Kit

The Oracle Database Client Developer Kit contains the following client libraries:

- Oracle Java Database Connectivity (JDBC) Drivers
- Oracle Java Messaging Service (JMS) Toolkit
- Oracle SQLJ Translator

Oracle Enterprise Manager Client

Oracle Enterprise Manager Client provides an integrated solution for centrally managing your Oracle environment. Combining a graphical console, Oracle Intelligent Agents, common services, and administrative tools, Oracle Enterprise Manager Client provides a comprehensive systems management platform for managing Oracle9i Application Server. To use this client, you must have a previously installed Oracle Management Server on your network.

Oracle HTTP Server

Oracle9i Application Server uses the Oracle HTTP Server, which is built on Apache Web server technology. Oracle HTTP Server offers scalability, stability, speed, and extensibility. It also supports Java Servlets, JavaServer Pages, Perl, PL/SQL, and CGI applications.

Oracle Internet Directory

A general purpose directory service that enables retrieval of information about dispersed users and network resources. Oracle Internet Directory combines LDAP version 3 with the high performance, scalability, robustness, and availability of the Oracle database.

Oracle Internet Directory runs as an application in the Oracle database. It communicates with the database, which may be on the same or a different operating system.

Oracle Internet File System

A file system that stores its contents in an Oracle database. The contents can range from files to metadata that defines the file system (including foldering, security, and users). Oracle Internet File System indexes textual content in files as they are added or updated. This enables you to search the contents of more than 150 file formats.

Oracle LDAP Developer Kit

LDAP (Lightweight Directory Access Protocol) is the emerging Internet standard for directory services. Oracle LDAP Developer Kit supports client interaction with any LDAP-compliant directory server, for example, Oracle Internet Directory. The toolkit provides tools and development libraries to support client calls to directory services, encrypted connections, and enables you to manage your directory data.

Oracle Management Server

Oracle Management Server provides distributed control between the database and Oracle9i Application Server in the network. As a central engine for notifications, it processes all system management tasks and administers the distribution of these tasks across the enterprise. Ensure that you do not have multiple Oracle Management Servers installed on a single machine.

Oracle Wallet Manager

A Java-based application that security administrators use to manage public-key security credentials on clients and server. Security credentials consist of a public/private key pair, a certificate, and a **trustpoint**.

See Also: [public-key cryptography](#), [wallet](#)

Oracle Workflow

Lets you automate and continuously improve business processes, routing information of any type according to easily-changeable business rules to users both inside and outside a company's enterprise.

Oracle XML Developer Kit

The Oracle XML Developer Kit (XDK) contains the necessary **XML** components libraries and utilities to give developers the ability to easily XML-enable applications and Web sites. Oracle XDK supports development in Java, C, C++, and PL/SQL with a collection of libraries, command-line utilities, and tools.

overloading

In programming, the ability to use the same name for more than one variable or procedure, requiring the compiler to differentiate them based on context. Overloaded procedures and functions (in PL/SQL) or methods (in Java) have the same name but take different parameters and do similar but not identical things.

ORB

Object Request Broker. In CORBA, an ORB acts as a “broker” between a client request for a service from a distributed object or component and completion of that request. Having ORB support in a network means that a client program can request a service without having to understand where the server is in the distributed network or what language the server object is written in.

OTS

Object Transaction Service. It provides the services necessary for applications and databases to become part of a transaction.

See Also: [JTS](#)

package

A group of PL/SQL or Java functions and procedures.

PL/SQL

Oracle's proprietary extension to the SQL language. PL/SQL adds procedural and other constructs to SQL that make it suitable for writing applications.

policy

An authorization rules repository for [JAAS](#). A policy contains the rules with which a user is authorized to use resources, such as a database. The primary function of a policy implementation is to answer the question:

Given a grantee, return the granted permissions for the grantee.

policy file

Grants permissions to the trusted codes or applications that you run. These codes or applications are not allowed to access resources unless they are explicitly granted permission to do so by the security policy in effect. In Java environments, the permission must be granted by an entry in a policy file.

policy store

A store that provides secure, centralized storage, retrieval, and administration of [JAAS](#) policies (the roles and privileges of a user). Oracle Internet Directory (OID) is the policy store used for storing this information.

polymorphism

In Object-Oriented Programming, polymorphism (from the greek meaning “heaving multiple forms”) is the characteristic of being able to assign a different meaning to a particular symbol or “operator” in different contexts.

port

A number that TCP uses to route transmitted data to and from a particular program.

portal

Portal sites give users a single, centralized, personalized view of relevant applications and data.

See Also: [Oracle9iAS Portal](#), [portlet](#)

portlet

A re-usable information component that summarizes or provides access to an information source. Portlets are the fundament building blocks of an [Oracle9iAS Portal](#) page.

See Also: [portal](#)

principal

A principal is a specific authenticated identity, such as a user named `frank`. A principal is essentially an identity associated with a subject. A principal can be associated with a subject (such as a user) upon successful authentication to a computing service.

private key

A key used by a limited number of communicating parties to decrypt data encrypted with a public key.

See Also: [public-key encryption](#)

processor sets

Introduced in Solaris 2.6, processor sets allow a group of processors to be allocated for the exclusive use of one or more applications. This gives a system administrator control over the creation, management, and binding of processes into processor sets.

protection domain

A fundamental building block of system security. The protection domain concept serves as a convenient mechanism for grouping and isolation between units of protection. For example, it is possible to separate protection domains from interacting with each other so that any permitted interaction must be either through trusted system code or explicitly allowed by the domains concerned.

proxy server

A proxy server typically sits on a network firewall and allows 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 allows an organization to create a secure firewall by preventing Internet access to internal machines, while allowing Web access.

privilege

The right to perform an action on the database. These can either be general (system privileges) or specific to particular database objects (object privileges). They can also be grouped into roles.

See Also: [role](#)

public key

A key known to all users, used to encrypt data in such a way that only a specific user can decrypt it.

See Also: [private key](#), [public-key encryption](#)

public-key cryptography

A set of well-established techniques and standards that ensure that data passes securely over a network, from the sender to the receiver, without any interference from a third party. Public-key cryptography facilitates [encryption](#) and decryption, tamper detection, [authentication](#), and nonrepudiation.

public-key encryption

A form of [encryption](#) that uses a key pair (a [public key](#) and a [private key](#)) to encrypt and decrypt data.

query string

The optional portion of a URL that specifies parameters to be passed to an application.

RC4, RC5

[RSA](#) encryption algorithms.

realms

Provides access to a policy store of users and roles (groups) and optionally provides administrative functionality. Oracle provides a proprietary Realm API that implements the notion of a user community. Essentially a user community instance is a realm that is maintained internally by the authorization system.

redundant

Duplicate or extra computing components that protect a computing system.

reliability

The ability of a computing system to operate without failing. Reliability is measured by mean-time-between-failures (MTBF).

remote method invocation

See: [RMI](#)

request latency

The time required to process a request.

request throughput

The number of requests processed per unit of time.

restriction

A security scheme that restricts access to files provided by the server to client machines within certain groups of IP addresses or DNS domains.

See Also: [domain-based restriction](#), [IP-based restriction](#)

reverse mapper

A utility that converts a CORBA server object's interfaces to IDL (for example, Java to IDL). These IDL interfaces can then be converted to CORBA stub and skeleton code by a IDL compiler (e.g. IDL-to-Java) for use by clients.

RFC

Request for Comments. A collection of Internet-related articles.

See Also: <http://www.rfc-editor.org>

RMI

Remote Method Invocation. An interaction scheme for distributed objects written in Java. It enables a Java program running on one computer to access the methods of another Java program running on another computer.

role

A group of database privileges that can be granted and revoked as a unit. Specific privileges can be granted and revoked from the role dynamically, and the role can be enabled or disabled dynamically for specific users.

role activation

A user is granted multiple roles, but not all roles are enabled by default. The user can selectively enable the required roles to accomplish a specific task in a user session, thus ensuring the principle of least privilege. That is, the user is not enabling permissions or privileges unnecessary for the task. This limits the damage that can result from an accident or error.

role-based access control

A process that simplifies the administrative process of authorization management based on direct assignment of permissions to users. Instead of directly assigning permissions to users, permissions are assigned to roles, and users are granted their permissions by being made members of appropriate roles. Multiple roles can be granted to a user, and a role can also be granted other roles, thus forming a role hierarchy that provides security administrators with a tool to model enterprise security policies.

routing

The process of directing data from one machine on the Internet to another by way of intermediate machines.

RSA

An Oracle partner supplying encryption algorithms for the HTTP server and Web Request Broker.

scalability

The ability to handle increasing numbers of hardware requests without adversely affecting latency and throughput.

schema

A collection of database objects, including logical structures such as tables, views, sequences, stored procedures, synonyms, indexes, clusters, and database links.

A schema has the name of the user who controls it.

secure sockets layer

See: [SSL](#)

security scheme

It prevents unauthorized users from accessing protected files or applications. The Oracle9i platform supports both **authentication** schemes and **restriction** schemes.

The Oracle9i platform supports the following types of authentication:

- **basic authentication**
- **digest authentication**
- **certificate authentication**

The following restriction schemes are supported:

- **IP-based restriction**
- **domain-based restriction**

serializable

An interface that a Java class implements to enable itself to be persistent through the Java Serialization service. This service allows a bean's state to be stored and retrieved.

server

There are two types of servers relevant to this product. The first is Oracle9i Database Server, which is a relational database server dedicated to performing data management duties on behalf of clients using any number of possible interfaces. The other is Oracle9i Application Server which is a collection of middleware services and tools that provide a scalable, robust, secure, and extensible platform for distributed, object-oriented applications. Oracle9i Application Server supports access to applications from both Web clients (browsers) using **HTTP** and Common Object Request Broker Architecture (**CORBA**) clients, which use the CORBA and the Internet Inter-ORB (**IIOP**) protocols.

servlet

A small Java program that runs on a Web server, as opposed to an applet, which is a Java program that runs in a client browser. Servlets take client HTTP requests from a browser, generate dynamic content (such as through querying a database), and provide an HTTP response back to the browser.

servlet container

The runtime environment for a Java Servlet. The servlet container calls the methods of a servlet and provides services to the servlet while it is executing.

servlet zones

Servlet zones are analogous to virtual directories in a Web server. Each zone can have its own set of servlets and can use a dedicated Java virtual machine (JVM). All of the servlets in one servlet zone share the same servlet repository, which is a location where Java class files are loaded.

session beans

A session bean implements one or more business tasks. A session bean might contain methods that query and update data in a relational table. Session beans are

often used to implement services. For example, an application developer might implement one or several session beans that retrieve and update inventory data in a database.

session key

A secret key used by SSL to encrypt data transmitted over a secure connection. The client generates the session key after Oracle9i Application Server authenticates itself and sends it to Oracle9i Application Server using public-key encryption.

shadow process

This is a very simple process on every host that detects if an Oracle HTTP Server listener process on that host is alive. If the listener process dies, then the shadow process brings a new instance up to handle requests.

shared secret-key encryption

A form of encryption that uses a single key both to encrypt and to decrypt a document. Shared secret-key encryption is much faster than public-key encryption, but is more vulnerable to attack.

Simple Network Management Protocol

See: [SNMP](#)

Simple Object Access Protocol

See: [SOAP](#)

single signon

See: [SSO](#)

skeletons

In [CORBA](#), skeletons are classes that are generated as part of the IDL compilation process. They receive calls from client stubs, unmarshal parameters, and call the method implementation.

See Also: [IDL](#), [skeletons](#)

SNMP

Simple Network Management Protocol. A set of protocols for managing complex networks. The versions of SNMP were developed in the early 1980s. SNMP works by sending messages, called protocol data units (PDUs), to different parts of a network. SNMP-compliant devices, called agents, store data about themselves in Management Information Bases (MIBs) and return this data to the SNMP requesters.

SOAP

Simple Object Access Protocol. A lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment. SOAP supports different styles of information exchange, including: Remote Procedure Call style (RPC) and Message-oriented exchange. RPC style information exchange allows for request-response processing, where an endpoint receives a procedure oriented message and replies with a correlated response message. Message-oriented information exchange supports organizations and applications that need to exchange business or other types of documents where a message is sent but the sender may not expect or wait for an immediate response.

SQL

See: [structured query language \(SQL\)](#)

SSL

Secure Sockets Layer. 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.

SSO

Single Signon. The ability for a user to authenticate once, combined with strong authentication occurring transparently in subsequent connections to other databases or applications. SSO lets a user access multiple accounts and applications with a single password.

stateful application

An application that maintains its own state information.

stateful session bean

In a stateful session bean, instance variables represent the state of a unique client-bean session. Because the client interacts with its bean, this state is often

called the conversational state. This conversation state is retained for the duration of the client-bean session. If the client removes the bean or terminates, the session ends and the state disappears.

stateless application

An application that does not store state information. Requests to a stateless application must include all information that the application needs to run properly.

stateless session bean

A stateless session bean does not maintain a conversational state for a particular client. When a client invokes the method of a stateless bean, the bean instance variables may contain a state, but only for the duration of the invocation. When the method is finished, the state is no longer retained. Except during method invocation, all instances of a stateless bean are equivalent, allowing the EJB container to assign an instance to any client. Because stateless session beans can support multiple clients, they can offer better scalability for applications that require large numbers of clients.

stored procedure

A set of PL/SQL instructions that are stored in a database.

stubs

In **CORBA**, stubs are classes that are generated as part of the IDL compilation process (e.g. using an IDL-to-Java compiler). The compiler creates the stubs from the IDL interfaces of a server object. The stubs contain code that handles operations needed to access remote objects. A client application uses the generated stubs to access methods in the remote server object.

See Also: [IDL](#) and [skeletons](#)

structured query language (SQL)

The internationally accepted standard for relational systems, covering not only query but also data definition, manipulation, security, and some aspects of referential integrity.

See Also: [PL/SQL](#)

subagent

A process that receives queries for a particular managed element from the master agent and sends back the appropriate answers to the master agent. The SNMP subagent detects unexpected events and forwards them to the master agent.

subject

A subject refers to any user of a computing service. Since users, roles, and computing services use computing services, all are subjects. A subject is the source of a service request. To identify the subjects with which it interacts, a computing service typically relies on names. Subjects can have different names for each individual service with which they interact. These individual names are known as principals. For example, a user can have a name principal (user `frank`), a role principal (role `hr`), and a social security number principal (`123-45-6789`).

subscribers

Customers (such as a company) that use (subscribe to) an application deployed in a hosted environment.

subscription

In the scope of the performance monitoring tools, a subscription is a request by a user to track a particular **metric**.

three-tier architecture

A client-server computing architecture where systems are separated into three layers: the interface layer, the application logic layer, and the database layer. Clients in this model are thin clients which do not contain as many resources as clients in the two-tier architecture. The application logic layer which is also known as the middle-tier provides the computing power and resources for the client. Oracle9i Application Server fulfills this role ideally as a development and deployment platform for database enabled applications.

topology management request

This is an HTTP request that is handled by the Oracle HTTP Server module, `mod_oprocmg`. It is used for server process management and administration.

TNS String

Transparent Network Substrate string, a foundation technology, built into Net8, Oracle Connection Manager and Oracle Names, that works with any standard network transport protocol. The TNS string contains destination service and network route information. Users initiate a connect request by passing a user name

and password along with a connect identifier in a TNS String for the service to which they wish to connect.

transaction

A transaction is a collection of operations that work toward a goal of satisfying a single user's need. A transaction guarantees that all operations within it either succeed or fail.

As an example, a transfer from a savings account to a checking account has a transaction that debits the savings account and credits the checking account. If both operations succeed, the transaction is committed, otherwise the transaction is rolled back.

transaction service

An OTS service that enables you to perform transactions that span several HTTP requests. The transaction service is based on the XA open model transactions defined by the X/Open Company.

transactional DAD

A database that has been configured to be involved within a transaction. Once configured, more than one database can be managed by Oracle9i Application Server in committing resources on all databases involved in the transaction. A database cannot be involved in a transaction unless the **DAD** is configured to be transactional.

trustpoint

Also referred to as a trusted certificate authority (**CA**), a trustpoint is an issuer of **certificates** whom you can trust. You can use **Oracle Wallet Manager** to manage trustpoint information associated with your application.

two-tier architecture

The traditional client-server computing architecture with application front-end and logic executing on a client that accesses a database.

UDDI

Universal Description, Discover, and Integration. An online electronic registry that serves as electronic Yellow Pages, providing an information structure where various business entities register themselves and the services they offer through their WSDL definitions. There are two types of UDDI registries, public UDDI registries that serve as aggregation points for a variety of businesses to publish their services, and private UDDI registries that serve a similar role within organizations.

UNICODE

A 16-bit character encoding scheme supporting a large subset of characters found in both Western and Asian languages.

uniform resource identifier

See: [URI](#)

uniform resource locator

See: [URL](#)

Universal Description, Discover, and Integration

See: [UDDI](#)

URI

Uniform Resource Identifier. The generic term for all types of names and addresses that refer to objects on the World Wide Web. A URL is one kind of URI.

URL

Uniform Resource Locator. A URL, a form of URI, is a compact string representation of the location for a resource that is available via the Internet. It is also the text-string format Web clients use to encode requests to Oracle9i Application Server.

virtual host

See: [multiport](#)

virtual machine

See: [JVM](#)

virtual pathname

In a virtual file system, a virtual pathname is a synonym that the virtual file system maps to a file stored in the file system maintained by the host machine's operating system.

VM

See: [JVM](#)

wallet

A wallet is an abstraction used to store and manager security credentials for an individual entity. It implements the storage and retrieval of credentials for use with various cryptographic services.

WAR Files

Web Archive (WAR) files contain the Web component modules of J2EE applications. A WAR file contains its **deployment descriptors** (DD), related files, and the `.class` files for the servlet or the `.jsp` files for a JSP page.

watcher

A thread in an Oracle HTTP Server parent process. It is spawned by the parent `init` process of `mod_oprocmgr`. Its purpose is to periodically check the status of each process running on the server. If a process is dead or if it fails to start up, then the watcher thread will issue an HTTP request to start a new process.

Web component

J2EE Web components can be either JavaServer Pages (**JSPs**) or **servlets**. Static HTML pages, applets, or server-side utilities can be bundled with Web components during application assembly, but are not considered Web components by the J2EE specification.

Web Services Description Language

See: [WSDL](#)

wireless portal

A portal accessible from wireless devices such as a cellular telephone.

See Also: [Oracle9iAS Wireless, portal](#)

WSDL

Web Services Description Language. An XML format for describing network services containing RPC-oriented and message-oriented information. Programmers or automated development tools can create WSDL files to describe a service and can make the description available over the Internet. Client-side programmers and development tools can use published WSDL descriptions to obtain information about available Web Services and to build and create proxies or program templates that access available services.

X.500

A standard for developing an electronic directory of users in an organization so that it can be part of a global directory available to anyone in the world with Internet access. Such a directory is sometimes called a global White Pages directory. The idea is to be able to look up people in a user-friendly way by name, department, or organization.

XML

eXtensible Markup Language. XML is a set of rules for defining data markup in a plain text format.

Index

A

access control
 portals, 3-12

adapters
 Oracle9iAS integration feature, 7-7
 supported types, 7-7

advanced queuing
 integration feature, 7-9

agent
 definition, 7-7

Apache HTTP Server, 2-23

applications
 deploying, 2-2
 developing, 2-2
 distributed, 2-5
 FastCGI, 2-23
 forms, 2-21
 integration support, 7-2
 J2EE clients, 2-7
 J2EE packaging, 2-8
 packaging, J2EE, 2-8
 portals, 3-2, 3-9, 3-11
 types, 2-2

application-to-application (A2A)
 integration support, 7-4

architecture
 OC4J, 2-3
 Oracle HTTP Server, 2-25
 Oracle9iAS Forms Services, 2-22
 Oracle9iAS Web Cache, 5-4
 portals, 3-9
 Web services, 2-16
 wireless, 4-2

automatic content compression, 5-9

availability, 1-5, 2-28

B

BC4J, 2-14

bean-managed persistence, 2-11

bridge
 definition, 7-7

business challenges, 3-3

Business Components for Java
 See BC4J

business intelligence, 1-9, 6-2

business logic
 Java, 2-4

C

C++
 Oracle HTTP Server, 2-23
 XML, 2-18

caching, 1-5, 1-9
 clustering, 5-11
 Java, 2-13
 overview, 5-2
 portals, 3-8

- categories, 3-6
- CGI
 - Oracle HTTP Server, 2-27
 - PL/SQL, comparison, 2-20
- challenges
 - deployment, 1-4
 - development, 1-2
- clickstream data, 6-3
- Clickstream Data Warehouse, 6-5
- clickstream loader, 6-7
- clustering, 2-28
 - application server instance
 - role, 2-34
 - benefits, 2-28
 - caches, 5-11
 - DCM, 2-36, 2-37
 - farms, 2-29
 - hardware load balancer, 2-33
 - instances, 2-29
 - Oracle Enterprise Manager, 2-31
 - Oracle9iAS Web Cache, 2-35
 - terminology, 2-28
 - using infrastructure, 2-34
- collector agents, 6-7
- collector server, 6-7
- commit
 - two-phase, 2-12
- connection rerouting, 2-40
- connectivity, remote, 2-10
- container managed transactions, 2-11
- container-managed persistence, 2-11
- content
 - delivery, 3-7
 - personalization, 3-7
 - sharing, 3-4
 - subscribers, 3-7
- content publishing, 3-5
- content syndication, 3-7
- Convenience, 8-10
- cookies, 6-4
- core, wireless, 4-5
- corporate portals, 3-3
- customization, wireless, 4-11

D

- data packets, 6-7
- database events, 6-14
- database providers, 3-9
- DCM
 - clustering, 2-38
 - role in clustering, 2-36, 2-37
- dcmctl, 2-37
- death detection, 2-39
- deployment challenges, 1-4
- development challenges, 1-2
- Discoverer reports
 - portals, 3-10
- distributed applications
 - Java, 2-5
- Distributed Configuration Management, see DCM
- DMS, 2-25
- Document Object Model
 - See DOM
- DOM, 2-2
 - APIs, 2-17
- DTDs, 2-17
- dynamic
 - pages, 2-20
- Dynamic Configuration Management, 2-33
- dynamic HTML, 2-7
 - PL/SQL, 2-19
- dynamic monitoring service, 2-25

E

- e-business
 - definition, 7-2
 - integration, 1-9
 - portals, 3-3
 - solution, 1-6
- e-business dashboard, 6-22
- edge side includes (ESI), 5-7
- Edge Side Includes for Java (JESI), 5-7

EJBs, 2-2

- bean-managed persistence, 2-11
 - container, 2-8, 2-10
 - container-managed persistence, 2-11
 - JDBC, relationship, 2-11
 - managing, 2-10
 - persistence, 2-11
 - running, 2-10
 - types, 2-10
 - version supported, 2-3
- e-mail
- notifications, Java, 2-12
 - wireless, 4-8
- end user layer, 6-6
- enhanced report bursting and distribution, 6-15
- enterprise beans, 2-7
- enterprise integration, 1-3
- Enterprise JavaBeans
- See EJBs
- enterprise warehousing environment, 6-5
- entity beans, 2-10
- ETL scripts, 6-7
- Extensible Stylesheet Language
- See XSLT

F

- failover, 2-28, 2-39, 5-10
- farms, 2-29
- FastCGI, 2-23, 2-27
- fault tolerance, 2-39
- features
 - wireless, 4-5
- file management, 2-23
- flexible deployment options, 5-12
- FormFiller, wireless, 4-10
- forms
 - applications, 2-21
- forms applications, 2-2
- full-page static and dynamic content caching, 5-5

G

- Gateway Provider, wireless, 4-2

H

- hardware load balancer, 2-33
- high availability
 - benefits
 - connection rerouting, 2-40
 - death detection and restart, 2-39
 - failover, 2-39
 - fault tolerance, 2-39
- HTML, 3-5
 - dynamic, 2-7
 - PL/SQL, 2-19
 - forms, 2-19
- HTTP Adapter
 - wireless, 4-3

I

- IDAP, 2-2
- information sharing, 3-4
- instances, 2-29
- integration
 - of applications, 7-2
 - Oracle9iAS Business Intelligence, 6-22
 - Oracle9iAS Portal and Oracle9iAS Clickstream Intelligence, 6-22
- internet applications, 1-8
- Internet Data Access Presentation
 - See IDAP
- intranet, 3-3
- introduction
 - wireless, 4-2
- invalidation for cache consistency
 - management, 5-8
- items, 3-5, 3-6

J

- J2EE, 1-2, 1-8
 - application clients, 2-7
 - application components, 2-7
 - applications, 1-3, 2-2, 2-5
 - applications, distributed, 2-5
 - BC4J
 - caching, 2-13
 - clients, 2-7
 - components, 2-7
 - Connector API
 - version supported, 2-3
 - connector API, 2-12
 - containers, 2-7
 - distributed applications, 2-5
 - Java Object Cache, 2-13
 - modelling, 2-14
 - packaging, 2-8
 - security, 2-13
 - services, 2-12, 2-13
- J2SE, 2-4
- JAAS, 2-2, 2-13
 - Oracle Internet Directory, 2-13, 8-13
 - Oracle9iAS Single Sign-On, 2-13
 - version supported, 2-3
- Java
 - Object Cache, 2-13
 - Oracle HTTP Server, 2-23
 - See J2EE*
 - XML, 2-18
- Java API for XML
 - See JAXP*
- Java Authentication and Authorization Service
 - See JAAS*
- Java Authentication and Authorization Service (JAAS), 8-11
- Java Database Connectivity
 - See JDBC*
- Java Message Service
 - See JMS*
- Java Naming and Directory Interface
 - See JNDI*
- Java servlets
 - See servlets*
- Java Transaction API
 - See JTA*
- Java2 Platform
 - Enterprise Edition
 - See J2EE*
 - Standard Edition
 - See J2SE*
- JavaBeans
 - Activation Framework
 - version supported, 2-3
- JavaMail, 2-12
 - version supported, 2-3
- JavaServer Pages
 - See JSPs*
- JAXP, 2-12
 - version supported, 2-3
- JDBC, 2-2, 2-12
 - DataDirect drivers, 2-12
 - EJBs, relationship, 2-11
 - non-Oracle connectivity, 2-12
 - version supported, 2-3
- JDeveloper
 - See Oracle9i JDeveloper*
- JMS, 2-2, 2-8, 2-12
 - version supported, 2-3
- JNDI, 2-2, 2-12
 - EJBs, relationship, 2-10
- JSPs, 2-2, 2-7
 - HTML, 2-9
 - JDBC, 2-12
 - managing, 2-9
 - request flow, 2-10
 - running, 2-9
 - servlets, relationship, 2-9
 - translator, 2-10
 - version supported, 2-3
 - XML, 2-9
- JTA, 2-2, 2-12
 - version supported, 2-3

L

- listener, 2-23
- load balancing, 5-10

M

- management, 1-9
- m-Commerce
 - drivers, wireless, 4-10
 - wireless, 4-9
- message-driven beans, 2-10
- Microsoft Internet Information Server, 2-25
- mobile wallet, wireless, 4-9
- MobileXML, wireless, 4-7
- mod_fastcgi, 2-23
- mod_oc4j, 2-23
- mod_oradav, 2-23
- mod_oss1, 2-24
- mod_osso, 2-24
- mod_perl, 2-23
- mod_plsql, 2-23
- Model-View Controller, 2-14
- modules, 2-23
- monitoring
 - dynamic monitoring service, 2-25
- MVC, 2-14
- MVC framework for J2EE, 2-14

N

- namespace
 - XML APIs, 2-18
- Netscape iPlanet, 2-25

O

- OC4J
 - architecture, 2-3
 - monitoring, 2-25
 - Web container, 2-8
- off-line management, wireless, 4-12
- OLTP, 6-17
- online, 6-3
- Online Studio, wireless, 4-10
- online transaction processing systems, 6-3, 6-17
- OPMN
 - role in clustering, 2-38
- Oracle Advanced Queuing
 - Oracle9iAS integration feature, 7-9
- Oracle Enterprise Manager, 8-4

- Oracle Forms, 2-2
- Oracle HTTP Server, 2-23
 - architecture, 2-25
 - CGI, 2-27
 - components, 2-23
 - FastCGI, 2-27
 - features, 2-25
 - modules, 2-23
 - monitoring, 2-25
 - proxy plug-in, 2-25
 - request flow, 2-24, 2-27
- Oracle Internet Developer Suite, 1-7
- Oracle Internet Directory, 8-12
 - JAAS, 2-13, 8-13
 - Oracle9iAS Single Sign-On, 8-13
- Oracle Internet File System
 - Oracle9iAS integration feature, 7-10
- Oracle Process Management and Notification, 2-33
- Oracle Process Management Notification, *see* OPMN
- Oracle Ultra Search, 3-6
- Oracle Workflow
 - Oracle9iAS integration feature, 7-8
- Oracle9i, 1-6
- Oracle9i Application Server
 - Overview, 1-7
- Oracle9i Database, 6-6
- Oracle9i JDeveloper, 2-19
- Oracle9i Lite, wireless, 4-12
- Oracle9i Reports Builder, 6-12, 6-14, 6-16
- Oracle9i Warehouse Builder, 6-5
- Oracle9i XML Developer Kit
 - See* XML
- Oracle9iAS, 1-7
 - solutions, 1-8
 - Wireless, 1-9
- Oracle9iAS Clickstream Intelligence, 6-2
 - features, 6-3
- Oracle9iAS Discoverer, 6-3, 6-17
- Oracle9iAS Forms Services, 2-21
 - architecture, 2-22

- Oracle9iAS integration
 - application integration, 7-2
 - Oracle Advanced Queuing, 7-9
 - Oracle Internet File System, 7-10
 - Oracle Workflow, 7-8
 - Oracle9iAS InterConnect, 7-3
 - Oracle9iAS InterConnect Adapters, 7-7
 - Oracle9iAS Unified Messaging, 7-11
- Oracle9iAS InterConnect
 - adapters, 7-7
 - Oracle9iAS integration feature, 7-3
- Oracle9iAS InterConnect adapters
 - supported types, 7-7
- Oracle9iAS Metadata Repository, 2-29
- Oracle9iAS Personalization, 6-3, 6-16
 - features, 6-9
- Oracle9iAS Portal Developer Kit, 3-9, 3-11
- Oracle9iAS Reports Services, 6-3, 6-12, 6-14
 - features, 6-13
- Oracle9iAS security, 8-7
- Oracle9iAS Single Sign-On, 8-9
 - features
 - convenience, 8-10
 - ease of administration, 8-10
 - increased security, 8-10
 - JAAS, 2-13
 - Oracle Internet Directory, 8-13
 - partner and external applications, 8-11
- Oracle9iAS Syndication Server, 3-7
- Oracle9iAS Unified Messaging
 - Oracle9iAS integration feature, 7-11
- Oracle9iAS Web Cache
 - architecture, 5-4
 - automatic content compression, 5-9
 - cache clustering, 5-11
 - edge side includes (ESI), 5-7
 - Edge Side Includes for Java (JESI), 5-7
 - features, 5-5
 - flexible deployment options, 5-12
 - full-page static and dynamic content
 - caching, 5-5
 - invalidation for cache consistency
 - management, 5-8
 - overview, 5-2
 - page assembly components, 5-6
 - partial-page caching and personalized content
 - assembly, 5-6
 - performance assurance heuristics, 5-8
 - security, 5-14
 - certificate, 5-16
 - certificate authority, 5-15
 - restricted administration, 5-14
 - secure sockets layer support, 5-15
 - wallet, 5-16
 - web server load balancing and failover, 5-10
- Oracle9iAS Web Cache
 - clustering, 2-35
- Oracle9iAS Web Services
 - See Web services
- Oracle9iAS Wireless
 - introduction, 4-2
- overview
 - caching, 5-2
 - Oracle9iAS Web Cache, 5-2
 - cost savings, 5-3
 - high availability, 5-3
 - performance, 5-3
 - reduce network traffic, 5-3
 - scalability, 5-3

P

- pages, portals, 3-2
- parallel page engine, 3-8
- parsing
 - XML, 2-17
 - XML APIs, 2-18
- partial-page caching and personalized content
 - assembly, 5-6
- partner and external applications, 8-11
- partner portlets, 3-11
- PDK, 3-9, 3-11
- performance, 1-5
 - assurance heuristics, 5-8
- Perl
 - interpreter, 2-24
 - Oracle HTTP Server, 2-23, 2-27
- persistence
 - EJBs, 2-11
- personal information management, wireless, 4-8
- personalization, 1-4
 - portal views, 3-4
 - portals, 3-7
- perspectives, 3-6
- PL/SQL, 2-19
 - applications, 2-2
 - CGI, comparison, 2-20
 - Oracle HTTP Server, 2-23
 - Server Pages
 - See PSP
 - Web toolkit, 2-2, 2-20
- portals, 1-4, 1-8, 3-3
 - access control, 3-12
 - applications, 3-2, 3-9, 3-11
 - architecture, 3-8, 3-9
 - building blocks, 3-5
 - caching, 3-8
 - categories, 3-6
 - content syndication, 3-7
 - developer kit, 3-9, 3-11
 - features, 3-4
 - introduction, 3-2
 - pages, 3-5
 - parallel page engine, 3-8
 - perspectives, 3-6

- providers, 3-9
- request flow, 3-9
- security, 3-12
- single signon, 3-4, 3-12
- user authentication, 3-12
- wireless access, 3-8
- portlets, 3-4, 3-5
 - applications, 3-9, 3-11
 - partner, 3-11
 - providers, 3-10
- programming languages, 2-2
- protocol, wireless, 4-6
- providers
 - portlets, 3-10
- proxy plug-in, 2-25
- PSP, 2-2, 2-20
- public Key Interface (PKI), 8-10
- publishing content, 3-5

Q

- query management, 6-8

R

- regions, 3-5
- registries
 - UDDI, 2-16
- reliability, 2-28
- remote connectivity, 2-10
- Remote Method Invocation
 - See RMI
- Remote Procedure Call
 - See RPC
- reports
 - portals, 3-10
- request flow
 - Java
 - JSPs, 2-10
 - servlets, 2-9
 - Oracle HTTP Server, 2-24, 2-27
 - Oracle9iAS Forms Services, 2-22
 - portals, 3-9

Request Processor, wireless, 4-2
RMI, 2-8
RPC, 2-16
runtime administrator, 6-6, 6-7

S

SAX, 2-2
 APIs, 2-17
scalability, 1-6, 2-28
schemas
 XML, 2-2
 XML processors, 2-18
security features, 1-9, 8-9
 Java, 2-13
 Java Authentication and Authorization Service (JAAS), 8-11
 Oracle Internet Directory, 8-12
 features, 8-13
 password management, 8-12
 self-service console, 8-12
 synchronization with Third Party LDAP servers, 8-12
 Oracle9iAS Single Sign-On, 8-9
 portals, 3-12
security management, 1-6
self-service, 3-4, 3-5
server push, wireless, 4-11
services
 content specific, 4-8
 J2EE, 2-12
 location-based, 4-8
 wireless, 4-6
servlets, 2-2, 2-7
 container, 2-8
 JDBC, 2-12
 JSPs, relationship, 2-9
 managing, 2-8
 request flow, 2-9
 running, 2-8
 version supported, 2-3
 Web services, relationship, 2-16
 XSQL, 2-19
session beans, 2-10
short message service, wireless, 4-11
Simple API for XML
 See SAX
Simple Object Access Protocol
 See SOAP
single signon, 2-24, 3-4
 portals, 3-4, 3-12
SOAP, 2-2, 2-16
solutions
 Oracle9iAS, 1-8
SQL
 JDBC, 2-12
 XML, 2-19
string data, 6-4
styles, 3-6
subscribers, content, 3-7
syndication server, 3-7
system management, 1-6
 architecture, 8-2
 infrastructure, 8-2
 metadata repository, 8-2
 middle tier, 8-2
 Oracle Internet Directory, 8-3
 Oracle9iAS Single Sign-On, 8-3
 features, 8-5
 dynamic monitoring service, 8-6
 J2EE deployment and administration, 8-6
 Oracle Internet Directory, 8-12
 Oracle9iAS JAAS, 8-11
 Oracle9iAS Single Sign-On, 8-9
 performance monitoring, 8-5
 single point for distributed configuration and management, 8-5
 Oracle Enterprise Manager, 8-2
 Oracle Internet Directory, 8-2
 overview, 8-2
 security, 8-2

T

technologies
 supported, 2-2
third-party
 portlets, 3-11
transactions
 two-phase commit, 2-12
transcoding, wireless, 4-11
transviewer beans
 XML, 2-19
two-phase commit, 2-12

U

UDDI, 2-2, 2-16
Ultra Search, 3-6
Universal Description, Discovery, and Integration
 See UDDI
user authentication
 portals, 3-12
user interfaces
 Java, 2-4
user profiles, 3-7, 6-4
using
 Oracle Internet Directory with middle tier
 components, 8-13

V

version control, 2-23

W

W3C
 XML schemas, 2-18
 XML specifications, 2-17
Web Cache, 1-9
Web listener, 2-23
Web logs, 6-3
Web pages, 3-5
Web providers, 3-9
Web servers, 2-23
 non-Oracle servers, 2-25

Web services, 1-3, 2-2, 2-15
 architecture, 2-16
 Description Language
 See WSDL
 framework, 2-16
 servlets, relationship, 2-16
 SOAP, 2-16
 UDDI, 2-2, 2-16
 WSDL, 2-16
wireless, 1-4, 1-9
 architecture, 4-2
 content specific, 4-8
 Core, 4-5
 core, 4-5
 customization, 4-11
 description, 4-2
 device, 4-6
 device/network adapter, 4-4
 e-mail, 4-8
 features, 4-5
 FormFiller, 4-10
 HTTP Adapter, 4-3
 introduction, 4-2
 location-based, 4-8
 m-Commerce, 4-9
 m-Commerce drivers, 4-10
 mobile wallet, 4-9
 MobileXML, 4-7
 network, 4-6
 off-line management, 4-12
 Online Studio, 4-10
 Oracle9i Lite, 4-12
 Payment Processing, 4-10
 personal information management, 4-8
 PIM, 4-8
 portal connectivity, 3-8
 protocol, 4-6
 server push, 4-11
 services, 4-6
 short message service, 4-11
 transcoding, 4-11

- transformers, 4-4
 - device-specific, 4-4
 - generic, 4-4
- WAP device, 4-3
- XML, 3-8
- XML object, 4-3
- WML, 2-7
- WSDL, 2-2, 2-16

- UDDI, 2-2
- W3C specifications, 2-17
- wireless, 3-8
- WSDL, 2-2
- XPath, 2-2
- XSLT, 2-2
- XSLT APIs, 2-18
- XPath, 2-2
- XSLT, 2-2
 - APIs, 2-18
 - parsers, 2-17
- XSQL, 2-2
 - servlet, 2-19

X

XDK

See XML

XML

- applications, 2-2
- BC4J
- C++, 2-18
- class generators, 2-18
- database, 2-19
- datagrams, 2-19
- descriptor files, 2-5
- developer kit, 2-17
- DOM, 2-2
- DTDs, 2-17
- IDAP, 2-2
- Java, 2-18
- JAXP, 2-3
- JSPs, 2-7
- namespace
 - APIs, 2-18
- navigating, 2-17
- pages, 2-7
- parser APIs, 2-18
- parsers, 2-17
- parsing, 2-17
- Path Language
 - See* XPath
- portals, 3-5
- SAX, 2-2
- schema processors, 2-18
- schemas, 2-2
- SOAP, 2-2
- SQL, 2-2
- stylesheets, 2-17
- transviewer beans, 2-19