



CHAPTER 1

Overview of Technologies

4 Oracle Application Server 10g Web Development



Developing applications for the Web is a complex process. The number of programming languages, Application Programming Interfaces (APIs), frameworks, and development platforms is much higher than it has been in years past and it continues to grow at a lightning pace. IT departments wanting to maintain their competitive advantage are asked to evaluate and embrace new technologies faster than ever before. This leads to shorter development times for projects and shorter times for developers to absorb new technologies. These factors, coupled with the fact that developers are increasingly called upon to have a working knowledge of and provide input on things such as database design, networking, development methodologies, code-generating tools, Java frameworks, and source code management, has led to a crisis in the development world where developers don't have enough hours in the day to learn all of the aspects of the tasks they need to know to be productive members of their IT departments. Luckily for Oracle developers, Oracle has provided a series of products that allows you to create and test applications quickly, interface with the Oracle database seamlessly, and deploy those applications to the Web securely and with minimal effort. Oracle's development and Application Server products have the added advantage of maturity: they are stable, proven technologies used in a wide range of production environments throughout the world.

At the core of your development efforts is the Oracle Application Server, sometimes incorrectly referred to as a web server (web servers are a subcomponent of the Application Server). The Application Server takes requests from a client application, most likely a web browser such as Internet Explorer or Netscape Navigator, processes the requests, and returns the results. It then, depending on the type of request it receives, can call other programs or speak to databases to satisfy the client's request. The Application Servers uses Transmission Control Protocol/Internet Protocol (TCP/IP), the protocol defining the rules for establishing connections between computers and HyperText Transfer Protocol (HTTP) to control the communication between a web browser and a web server.

Oracle Application Server 10g is a product designed to meet all of the challenges your organization faces when putting your applications on the Web. While the details of installing and administering the Application Server are beyond the scope of this book, all Oracle developers can benefit from an understanding of the basic framework of web development and Oracle's solutions for each of these framework pieces. In the language used in Oracle's documentation, the implementation of these solutions is called *services*, and services are integrated into the Application Server 10g product stack.

For more information regarding installation and administration of Oracle Application Server 10g, see the Oracle Application Server 10g Administration Handbook by John Garmany and Donald K. Burleson (Oracle Press, 2004).

The amount of tools and services provided by Application Server 10g can seem overwhelming. You are certainly not required to implement every service provided by Application Server 10g; services that are installed and configured but not used waste resources, so choose your installation options carefully. Organizations can pick and choose those services they wish to exploit. Even if a particular service is not selected during installation, it is possible to go back and add that functionality to Application Server 10g at a later date without affecting your current production setup. This provides tremendous flexibility as developers can change direction without abandoning work performed up to that point.

A good example of this would be the Oracle Forms Server that is part of Application Server 10g. This service allows you to take existing Oracle Forms and serve them up over the Web. Perhaps

your organization has begun by using Oracle Portal to quickly develop some forms and reports that are published over the Web to your employees. This initial project has proven to be so popular that the decision has been made to move Oracle Forms, Oracle Reports, and Oracle Discoverer worksheets (which have richer feature sets than their equivalent components in Portal) to the Web and to integrate some of those components with Portal. An administrator can reconfigure Application Server 10g to add the necessary functionality without affecting any of the work performed up to this point.

By doing this, your organization can move to a web-based model without losing the time and effort already invested in Forms, Reports, or Discoverer development. As your needs grow, it is possible to augment your development efforts with PL/SQL server pages (PSPs) or a tool such as JDeveloper to implement technologies like JavaServer Pages (JSPs), Enterprise JavaBeans (EJBs), Struts, or Java applets. As the site traffic grows, you can increase performance by implementing the Oracle Web Cache service to cache the most heavily used pages. Power users can create their own ad-hoc reports using Discoverer. You can expand your Portal to integrate content, web-based components (such as forms, reports, or XML-based portlets) and content from external sites with a minimal investment in development and design time. The feature set of Application Server 10g can tackle virtually any development challenge organizations face today.

The architecture of Application Server 10g allows you to scale by adding more Application Servers (called *mid-tiers* in Oracle's documentation) and more Web Cache servers. You can even allow others to incorporate your content through the use of XML and Enterprise JavaBeans. All of this is managed through a single web interface, the Oracle Enterprise Manager Application Server Control.

One could argue that as a developer, it is not necessary to understand the architecture of Application Server 10g. While we have met many developers with this philosophy, we strongly urge you to make the effort to understand the fundamentals of how the Application Server does its basic functions. The administrative details of Application Server 10g will probably be handled by a systems administrator or a DBA, but, as we'll see in later chapters, there are many steps that fall upon the developer's shoulders as you test and implement your development efforts. A basic knowledge of Application Server 10g's structure is important and will make your job easier moving forward.

What Is Oracle Application Server 10g?

Oracle Application Server 10g is an application server designed to support all major web development languages and frameworks. It is a collection of services designed together to integrate seamlessly with Oracle databases and deliver content dynamically over the Web. Some of its features include:

- **Oracle HTTP Server** Application Server 10g includes an HTTP server based on the Apache HTTP Server, version 1.3.28. The Apache server is the most popular web server in use today. It is highly customizable and its architecture fully supports enhancements via modules.
- **Oracle Portal** Portal is a complete, web-based development environment that allows you to develop and implement production-quality applications quickly and easily.

6 Oracle Application Server 10g Web Development

- **Oracle Wireless** Wireless provides complete support for enabling your applications on a multitude of wireless devices.
- **Oracle Identity Management** Identity Management has security features that can be applied across all types of applications served up by Application Server 10g.
- **PL/SQL integration** Procedural Language extension to Structured Query Language (PL/SQL) integration gives you the ability to call PL/SQL stored procedures directly from your HTML-based applications via the PL/SQL Web Toolkit and the `mod_plsql` module included with Apache.
- **Oracle Forms, Oracle Reports, and Oracle Discoverer Servers** These servers give you the ability to serve Forms, Reports and Discoverer workbooks and worksheets over the Web and integrate those components with Portal.

On top of all of these features, Application Server 10g also supports the full Java 2 Enterprise Edition technology stack, including:

- **Enterprise JavaBeans (EJBs)** EJBs enable applications to use entity, session, and message-driven beans. They come with an EJB container that provides services for you. Services include transaction, persistence, and lifecycle management.
- **Servlets** Servlets can generate dynamic responses to web requests.
- **JavaServer Pages (JSP)** Enable you to mix Java and HTML to author web applications easily. JSP also enable you to generate dynamic responses to Web requests. Servlets and JSP run within a “web container,” which also provides services similar to those provided by the EJB container.
- **Java Authentication and Authorization Service (JAAS)** JAAS enables you to authenticate users (ensures that users are who they claim to be) and authorizes users (checks that users have access to an object before executing or returning the object).
- **Java Message Service (JMS)** JMS enables you to send and receive data and events asynchronously.
- **Java Transaction API (JTA)** JTA enables your applications to participate in distributed transactions and access transaction services from other components.
- **J2EE Connector Architecture** The J2EE Connector Architecture enables you to connect and perform operations on enterprise information systems.

Out of the box, Application Server 10g also includes various programs and web pages for the administration of this complex environment. The Enterprise Manager Application Server Control provides numerous pages for monitoring the various services in Application Server 10g and gives administrators the ability to edit configuration files and review log files via a web browser. This is invaluable for remote administration of application server instances when it is difficult to provide direct access to a server and simplifies the administrator’s job by not having to make them remember the file names and directory paths of the many log and configuration files used by Application Server 10g. Certain configuration editor pages even have syntax checking functionality built into them, allowing administrators to verify their changes before attempting to

implement modifications to components on the Server. Most of the tasks that can be performed on these pages are administrative and are of little interest to the developer, but, as we'll see in later chapters, there are pages devoted to the creation and maintenance of containers, which give developers the ability to deploy their applications quickly and easily through a graphical interface.

The power of Application Server 10g lies in its seamless integration with various Internet and programmatic standards and the Oracle database. It has evolved into a product that enables you and your organization to benefit from the true power of the Internet by providing a reliable, scalable, and secure deployment platform. Figure 1-1 provides a visual representation of the Application Server 10g product.

The boxes at the top of each section represent the different categories of services. The ovals represent Oracle's implementation of those services. This chapter discusses, at a high level, the

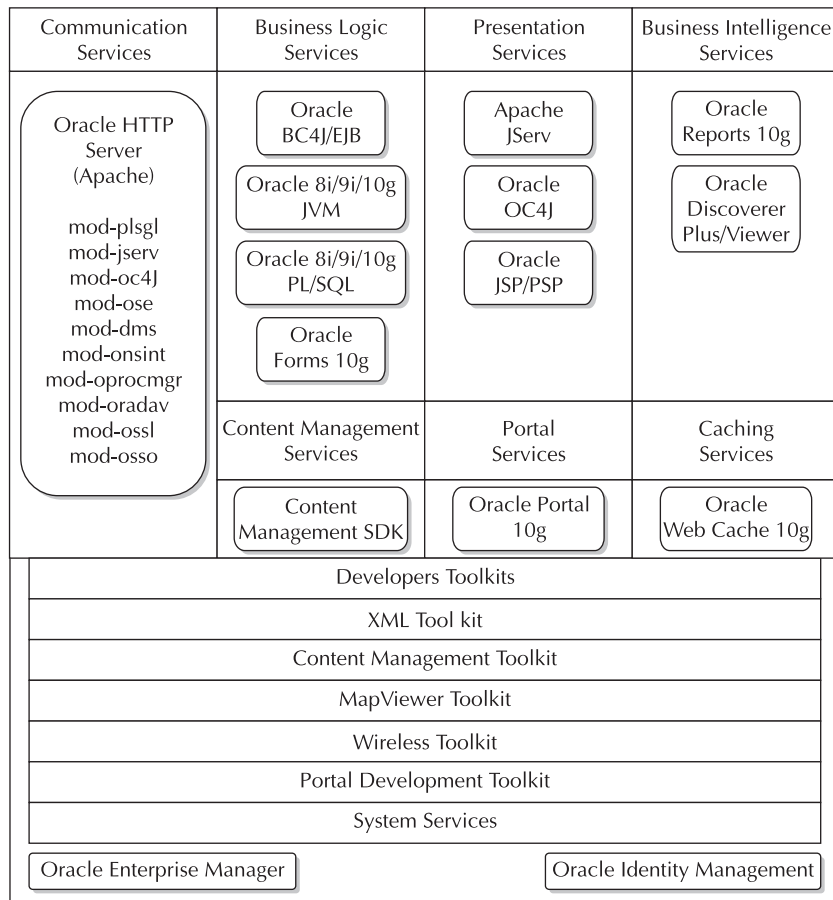


FIGURE 1-1. Oracle Application Server 10g Architecture

8 Oracle Application Server 10g Web Development

various services that Application Server 10g provides and why you, as a developer, are interested in them. Chapter 2 discusses Oracle's implementation of these services in detail. The architecture of Application Server 10g may seem overly complex, but this is necessary as it reflects Oracle's commitment to the numerous open standards and frameworks that constitute modern web development. The architecture also gives Application Server 10g a level of scalability and reliability not seen in most application servers.

There are two major pieces of Application Server 10g: the middle tier, which provides components for deploying and running applications over the Web, and the infrastructure, which maintains security and clustering information. In a one-to-many relationship, a single infrastructure can maintain security and clustering information for one or many middle tiers. When you install Application Server 10g, you must decide if you'll need an infrastructure because that piece needs to be installed first. We cover the different types of middle-tier installations, and if they are dependent on an infrastructure, next.

What is an infrastructure and a middle tier composed of? Each piece is composed of a series of programs, which provide various services to the Apache server. As an analogy, think of the services that run on a Windows machine—they are programs that provide various services to the operating system. In the same way, the programs that constitute the infrastructure and middle tiers provide services to the Application Server instance. The middle tier has three different versions that you can install, depending on your needs and licensing:

- J2EE and Web Cache
- Portal and Wireless
- Business Intelligence and Forms

Table 1-1 lists the different middle-tier components of Application Server 10g configured with each type of installation.

Based on this table, if you plan to serve Oracle Forms, Oracle Reports, and/or Oracle Discoverer workbooks and worksheets over the Web, install the Business Intelligence and Forms option. If you want to use Portal and/or develop applications to be used with wireless (handheld) devices, select the Portal and Wireless option. If you only need a web server to deploy J2EE applications, select the J2EE and Web Cache option. When you install the middle tier, you are creating an Oracle Application Server "instance."

The meaning of the word "instance" in this context is different from the more general use of "instance," which is commonly used to refer to an Oracle database instance. Figure 1-2 shows the administration page with all of the services for a middle tier installed with the Business Intelligence and Forms option.

As of version 8.1.7 of the Oracle database server, Oracle includes an HTTP server as part of its installation. It provides only the first two components in Table 1-1: an HTTP server and Oracle Application Server Containers for J2EE (OC4J) containers.

The second major piece of Application Server 10g is called the *infrastructure*. The infrastructure is an instance of Application Server 10g that maintains security and clustering information. In Oracle's documentation, the different types of security components provided by Oracle are

Component	J2EE and Web Cache	Portal and Wireless	Business Intelligence and Forms
Oracle HTTP Server	X	X	X
Oracle Application Server Containers for J2EE (OC4J)	X	X	X
Oracle Application Server Web Cache	X	X	X
Oracle Enterprise Manager web site	X	X	X
Oracle Application Server TopLink	X	X	X
Oracle Application Server Portal		X	X
Oracle Application Server Wireless		X	X
Oracle Application Server Personalization			X
Oracle Application Server Discoverer			X
Oracle Application Server Reports Services			X
Oracle Application Server Forms Services			X

TABLE 1-1. *Middle-Tier Components by Installation Type*

called “Identity Management.” The Identity Management components provide directory, security, and user management functionality. The Identity Management components are

- Oracle Internet Directory
- Oracle Application Server Single Sign-On
- Oracle Delegated Administration Services
- Oracle Directory Integration and Provisioning
- Oracle Application Server Certificate Authority

Each of the Identity Management components are discussed in Chapter 2. For both the Business Intelligence and Forms option and the Portal and Wireless option, an infrastructure is required. The infrastructure, by default, uses an Oracle 9i Database to store Identity Management information, but an Oracle9i or Oracle Database 10g can be manually specified. The Identity Management information is referenced through a Lightweight Directory Access Protocol (LDAP) server provided by Oracle called Oracle Internet Directory (OID). You can instruct the installer to use an existing Oracle9i Database or Oracle Database10g instance or have the installer create a new database instance for you. Figure 1-3 shows the administration page with all of the services for an infrastructure.

10 Oracle Application Server 10g Web Development

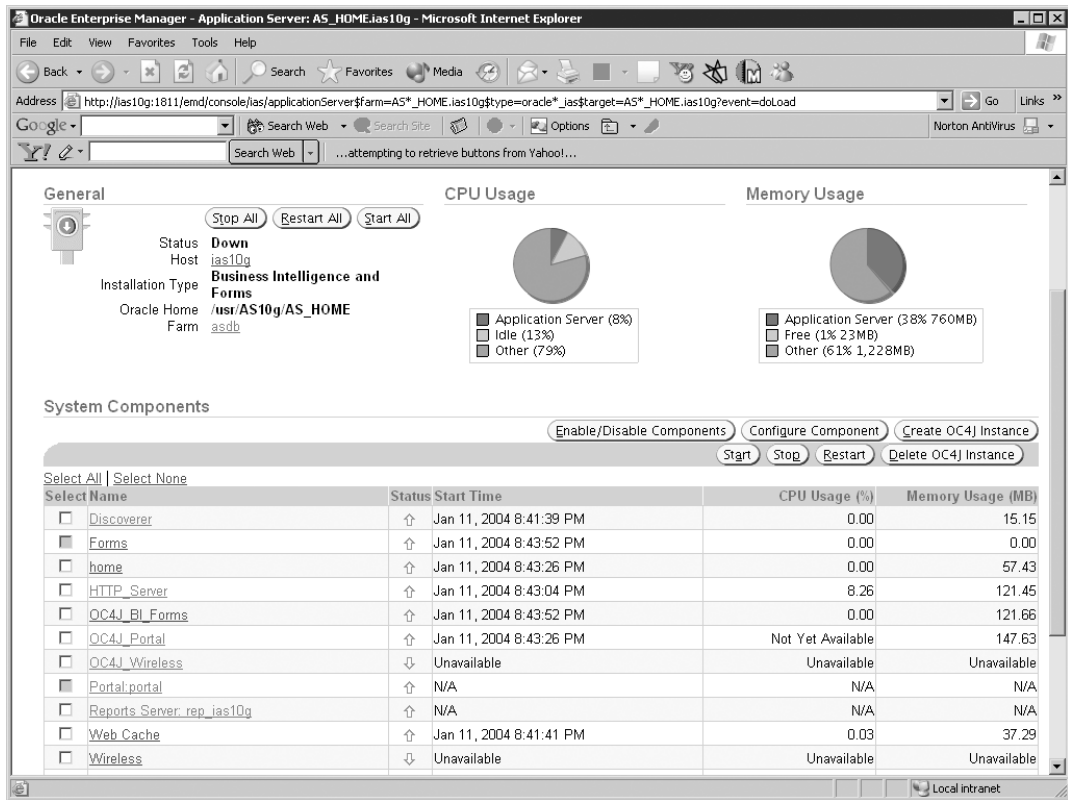


FIGURE 1-2. A Middle-tier administration screen

Technically, the Forms and Reports servers can be configured to *not* use an infrastructure in Application Server 10g, but it is far more common to use one.

“Topology” describes the layout of Application Server 10g instances installed in your organization. Application Server 10g can be installed in so many different ways that there is an entire chapter on it, entitled “Recommended Topologies,” in the installation guide. The most common topology is to install the infrastructure on one machine and the middle tier on another, although you can install both the infrastructure and middle tier on the same machine, provided you have enough disk and memory resources (as each instance needs to be installed in its own ORACLE_HOME), or numerous middle tiers on numerous servers. Figure 1-4 lists some common topologies. In an effort to provide organizations with a high amount of flexibility, the infrastructure and middle tiers do not have to be running on the same hardware or operating system platform. They do, however, have to be running the same version of the Oracle Application Server 10g software.

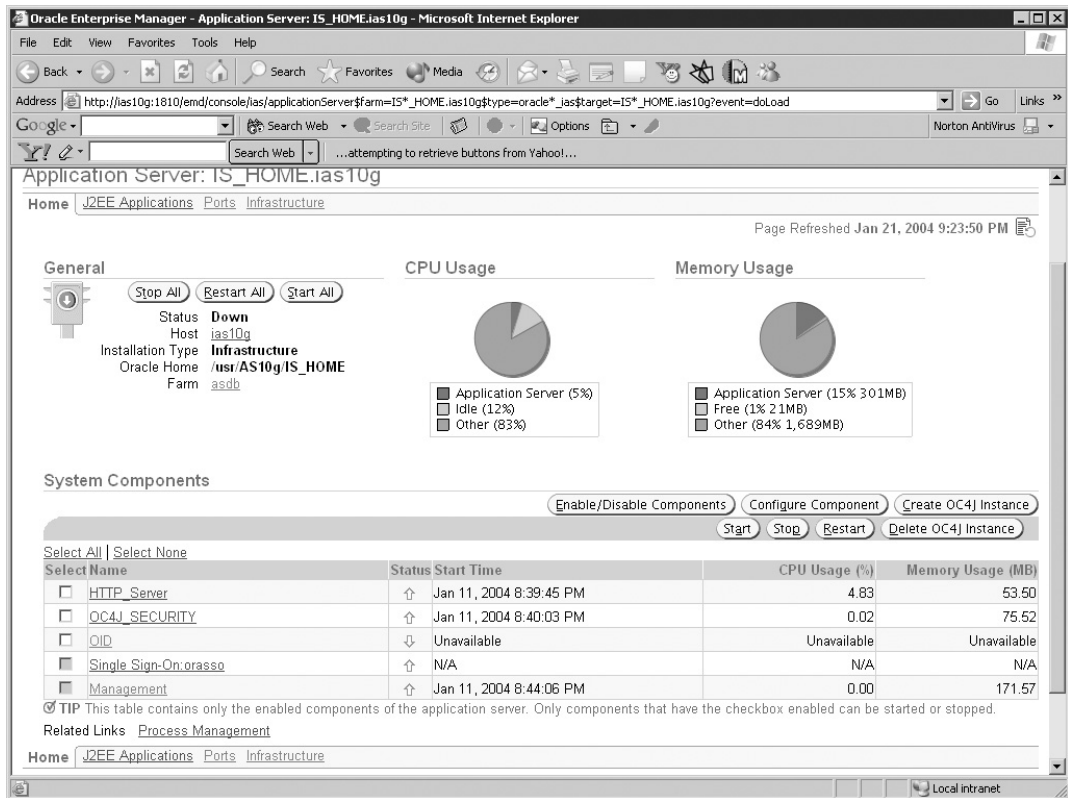


FIGURE 1-3. Administration screen for the infrastructure

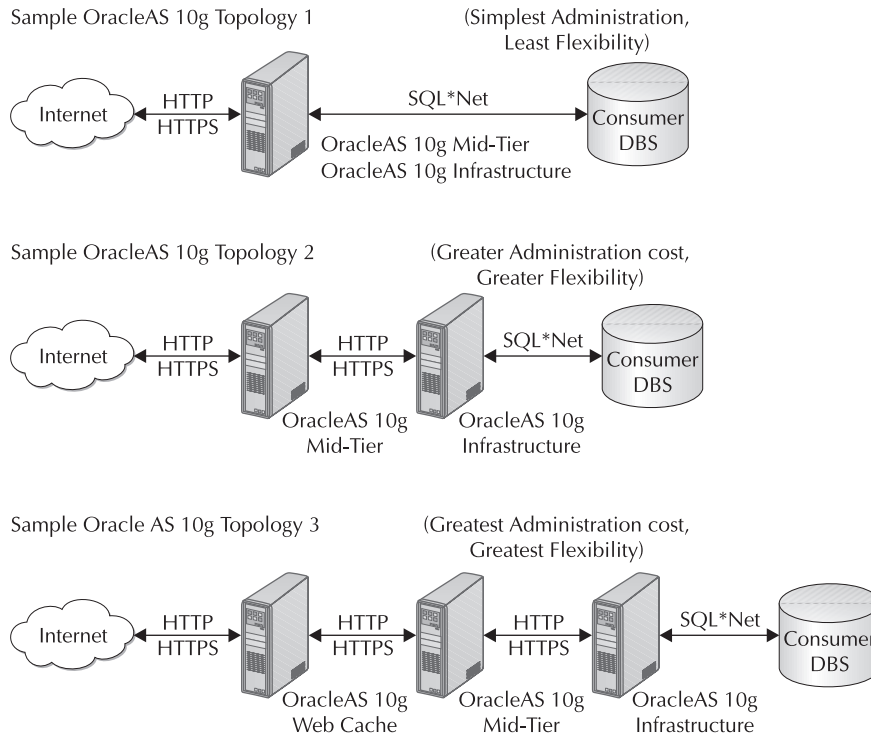
Since all instances **must** be running the same version of the software, this means that any patch applied to an Oracle Application Server 10g instance must be applied to **all** instances, even if they're running on the same machine.

The different categories of the Application Server architecture (see Figure 1-1) discussed in this chapter are grouped according to the type of function they provide:

- **Communication Services** These services handle requests to and from the Web.
- **Business Logic Services** These services are the development tools and languages for building applications.
- **Presentation Services** These service are the development tools and languages for building dynamic web pages.

12 Oracle Application Server 10g Web Development

- **Caching Services** These services are tools for improving web site performance.
- **Content Management Services** These services are tools for managing documents in the database.
- **Portal Services** These services provide publishing features for content and portlets.
- **Business Intelligence Services** These services provide reports and ad-hoc queries.
- **Database Services** These services make up the Oracle database for storing application data.
- **Persistence Layer Services** These services provide an object-relational framework.
- **Developer's Toolkits** APIs to aid in the creation of applications.



* For advanced Web Cache Topologies, see Figure 1-11

FIGURE 1-4. *Common topologies*

Communication Services

Communication services provide connectivity between clients (most likely a PC or a Macintosh running a web browser) and a server over the Internet or, more accurately, a TCP/IP connection. The most common way of interaction is through a web browser, which uses a protocol called HTTP. HTTP can handle multimedia communication services such as audio, video, and data. Tools like Oracle Portal take advantage of these features to seamlessly store and retrieve almost any type of data from an Oracle database. For the purposes of Oracle web development, the communication services are the web server.

Starting with Oracle9i Application Server Release 1, Oracle has used Apache as its web server, referred to in the Oracle documentation as the HTTP Server. Apache has a large base of knowledge available for its administration, as it is the most popular web server used today. A quick search of a popular technical book site turned up no less than 35 books related to Apache administration and configuration. By standardizing on the Apache web server, Oracle has provided a great amount of flexibility to administrators and organizations that wish to move their Oracle-based applications to the Web.

The designers of Apache were smart enough to know that there was no way they could anticipate every conceivable way people might want to use their web server, so they designed it with an open-ended interface that incorporates a modular architecture through which additional functionality can be added. These enhancements are commonly referred to as “mods” (modules) as they all begin with the prefix “mod_.” Modules may be linked statically to the web server or may be loaded dynamically at run time using Dynamic Shared Object (DSO) support on Unix or Dynamically Linked Libraries (DLLs) on Windows. The API for these modular components is based on the C programming language. Table 1-2 lists the mods that come with the Apache server provided with Application Server 10g.

Module	Functionality Provided to the Apache Server	Provided by Oracle?	Supported by Oracle?
mod_access	The directives provided by mod_access are used in <Directory>, <Files>, and <Location> sections as well as .htaccess files to control access to particular parts of the server.	No	Yes
mod_actions	The Action directive lets you run CGI scripts whenever a file of a certain type is requested. The Script directive lets you run Common Gateway Interface (CGI) scripts whenever a particular method is used in a request.	No	Yes
mod_alias	The directives contained in this module allow for manipulation and control of Uniform Resource Locators (URLs) as requests arrive at the server. The Alias and ScriptAlias directives are used to map between URLs and filesystem paths.	No	Yes

TABLE 1-2. *Apache Modules Provided in Oracle Application Server 10g*

Module	Functionality Provided to the Apache Server	Provided by Oracle?	Supported by Oracle?
mod_asis	This module provides the handler send-as-is, which causes Apache to send the document without adding most of the usual HTTP headers.	No	Yes
mod_auth	This module allows the use of HTTP Basic Authentication to restrict access by looking up users in plain text password and group files.	No	Yes
mod_auth_anon	This module does access control in a manner similar to anonymous File Transfer Protocol (FTP) sites; i.e., have a “magic” user id “anonymous” and the e-mail address as a password.	No	Yes
mod_auth_db	This module provides an alternative to DBM files for those systems that support DB (Berkeley database files) and not DBM (indexed Berkeley database files).	No	No
mod_auth_dbm	This module provides for HTTP Basic Authentication, where the usernames and passwords are stored in DBM-type database files.	No	No
mod_auth_digest	This is an updated version of mod_digest using MD5 authentication (experimental).	No	No
mod_autoindex	This module provides for automatic directory indexing.	No	Yes
mod_cern_meta	This module emulates the CERN HTTPD Meta file semantics. Meta files are HTTP headers that can be output in addition to the normal range of headers for each file accessed.	No	No
mod_cern_headers	This module allows reverse proxies that terminate Secure Sockets Layer (SSL) connections in front of Oracle HTTP Server to transfer information regarding SSL connection to Oracle HTTP Server.	No	Yes
mod_cgi	Any file that has the mime type application/x-httpd-cgi or handler cgi-script is treated as a CGI script, and run by the server, with its output being returned to the client.	No	Yes
mod_define	This module provides support for Distributed Authoring and Versioning.	No	Yes (Unix only)
mod_digest	This module implements an older version of the MD5 Digest authentication; use mod_auth_digest.	No	Yes
mod_dir	The DirectoryIndex directive sets the name of a file written by the user, typically called index.html, used as the index of a directory.	No	Yes

TABLE 1-2. *Apache Modules Provided in Oracle Application Server 10g (continued)*

Module	Functionality Provided to the Apache Server	Provided by Oracle?	Supported by Oracle?
mod_dms	This module enables you to monitor the performance of site components with Oracle's Dynamic Monitoring Service (DMS).	Yes	Yes
mod_env	This module allows for control of the environment that will be provided to CGI scripts and SSI pages.	No	Yes
mod_example	The files in the src/modules/example directory under the Apache distribution directory tree are provided as an example to those that wish to write modules that use the Apache API.	No	No
mod_expires	This module provides for the generation of Expires HTTP headers (an instruction to the client about the document's validity and persistence) according to user-specified criteria.	No	Yes
mod_fastcgi	This module routes requests to fastcgi modules.	No	Yes
mod_headers	This module provides for the customization of HTTP response headers.	No	Yes
mod_ldap	This module processes .map files, thereby replacing the functionality of the imagemap CGI program.	No	No
mod_include	This module provides a handler, which will process files before they are sent to the client.	No	Yes
mod_info	This module provides a comprehensive overview of the server configuration, including all installed modules and directives in the configuration files.	No	Yes
mod_isapi	This module implements the Internet Server extension API. It allows Internet server extensions (e.g., ISAPI DLL modules) to be served by Apache for Windows.	No	No
mod_jserv	This module converts HTTP requests to servlet requests, returning HTTP responses to the client. It is disabled by default in the Oracle HTTP Server distribution; use mod_oc4j instead.	No	Yes
mod_log_agent	This module provides for logging of user agents. It's deprecated; use mod_log_config instead.	No	No
mod_log_config	This module provides for logging of the requests made to the server, using the Common Log Format or a user-specified format.	No	Yes
mod_log_referer	This module provides for logging of the documents that reference documents on the server. It's deprecated; use mod_log_config instead.	No	Yes

TABLE 1-2. *Apache Modules Provided in Oracle Application Server 10g (continued)*

Module	Functionality Provided to the Apache Server	Provided by Oracle?	Supported by Oracle?
mod_mime	This module is used to determine various bits of “meta information” about documents. The directives AddCharset, AddEncoding, AddHandler, AddLanguage, and AddType are all used to map file extensions onto the meta-information for that file.	No	Yes
mod_mime_magic	This module provides for determining the MIME type of a file by looking at a few bytes of its contents. It is intended as a “second line of defense” for cases that mod_mime can’t resolve.	No	Yes
mod_mmap_static	This module maps a list of statically configured files into memory. It is an experimental module and should be used with care.	No	No
mod_negotiation	This module provides for content selection that’s defined as the selection of a document that best matches the client’s capabilities.	No	Yes
mod_oc4j	This module routes requests to the OC4J instances.	Yes	Yes
mod_onsint	This module provides integration support with Oracle Notification Service (ONS) and Oracle Process Manager and Notification Server (OPMN).	Yes	Yes
mod_oprocmgr	This module provides process management and load balancing services to JServ processes. It is provided for legacy users of JServ. JServ is disabled by default in the Oracle HTTP Server configuration. Oracle recommends using OC4J and mod_oc4j (which are enabled by default).	Yes	Yes
mod_oradav	This module provides enhancements to mod_dav to allow Oracle DB as a backing store.	Yes	Yes
mod_oss1	This module enables strong cryptography for Oracle HTTP Server. It is very similar to mod_ssl; mod_oss1 is based on the Oracle implementation of SSL, which supports SSL, version 3.	Yes	Yes
mod_osso	This module supports single sign-on across sites and applications.	Yes	Yes
mod_perl	This module runs perl programs.	No	Yes
mod_plsql	This module routes requests to pl/sql programs in databases; it enables requests to database stored procedures to be made from the browser.	Yes	Yes
mod_proxy	This module implements proxying capability for FTP, CONNECT (for SSL), HTTP/0.9, HTTP/1.0, and (as of Apache 1.3.23) HTTP/1.1.	No	Yes

TABLE 1-2. Apache Modules Provided in Oracle Application Server 10g (continued)

Module	Functionality Provided to the Apache Server	Provided by Oracle?	Supported by Oracle?
mod_rewrite	This module uses a rule-based rewriting engine (based on a regular-expression parser) to rewrite requested URLs on the fly.	No	Yes
mod_setenvif	This module allows you to set environment variables according to whether different aspects of the request match regular expressions you specify.	No	Yes
mod_so	On selected operating systems, this module can be used to load modules into Apache at run time via the Dynamic Shared Object (DSO) mechanism, rather than requiring a recompilation; mod_so is experimental.	No	Yes
mod_speling	This module attempts to correct misspellings of URLs that users might have entered by ignoring capitalization and allowing up to one misspelling.	No	Yes
mod_status	This module allows server administrators to find out how well their servers are performing. A HTML page is presented that gives the current server statistics in an easily readable form.	No	Yes
mod_unique_id	This module provides a magic token for each request, which is guaranteed to be unique across “all” requests under very specific conditions. The unique identifier is even unique across multiple machines in a properly configured cluster of machines.	No	Yes
mod_userdir	This module provides for user-specific directories. The UserDir directive sets the real directory in a user’s home directory to use when a request for a document for a user is received.	No	Yes
mod_usertrack	This module uses cookies to provide for a clickstream log of user activity on a site. Oracle Application Server 10g contains a service called Clickstream Intelligence that enhances this functionality.	No	Yes
mod_vhost_alias	This module creates dynamically configured virtual hosts by allowing the IP address and/or the Host: header of the HTTP request to be used as part of the pathname to determine what files to serve.	No	Yes

TABLE 1-2. *Apache Modules Provided in Oracle Application Server 10g (continued)*

A page provided by Oracle in the Enterprise Manager web site called the Module Metrics page lists various Apache modules and how often they have been accessed since the startup of the HTTP Server (see Figure 1-5).

The mods provided by Oracle will be discussed at various times in this book. For more information about Apache, go to <http://www.apache.org>.

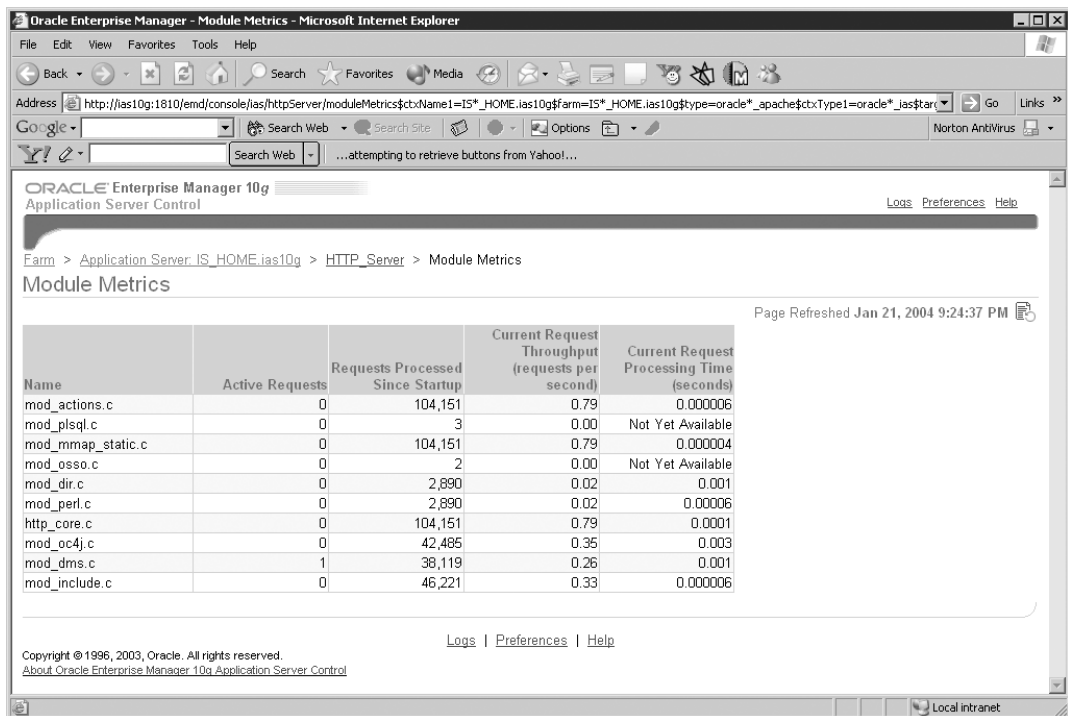


FIGURE 1-5. The Module Metrics page for the HTTP Server

Content Management Services

All organizations have different types of data that they use on a regular basis in the course of their activities. Some of this data is in spreadsheets, some in Oracle databases, some in non-Oracle databases, some is in picture and sound files—the list goes on and on. “Content Management” refers to the process of capturing, storing, sorting, codifying, integrating, updating, and protecting any and all of an organization’s information. Oracle Application Server 10g provides content management services that allow organizations to store, manage, and retrieve all different types of data in a central place; namely, an Oracle database.

Storing disparate data is not the only objective, however. For any content management service to be truly useful, it must also take into account such factors as connecting to other people’s information, approval and publication processes, indexing, search and retrieval capabilities, security, collaboration, accessibility, reuse, and digital rights management. In Oracle9i Application Server Release 1, Oracle introduced the Oracle Internet File System (iFS), which stored files in an Oracle 8i Database or Oracle 9i Database. iFS, in effect, made all content available in a file hierarchy that could be accessed through a web browser, Windows networking, or FTP, independent of platform (except for Windows networking, of course). Application Server 10g has added many new features to its content management capabilities and renamed it the Oracle Content Management Software Development Kit (SDK). The Content Manager SDK has support for

all major content management services listed above. Oracle Ultra Search can be used to search across corporate web servers, databases, mail servers, file servers, and Oracle 10g Portal instances. Ultra Search is based on Oracle 10g Text technology and is an out-of-the box solution that requires no SQL coding. It uses a crawler to index documents (the documents stay in their own repositories), and the crawled information is used to build an index that stays within your firewall in an Oracle database.

Business Logic Services

As we will see over the course of this book, the various technologies we can use for our web-based applications all have benefits and limitations. Not every technology is appropriate for every circumstance and no one tool is the “right” one to go with. Some of the technologies are good for displaying (or “presenting”) data, and are generally classified under the category “Presentation Services.” Other technologies are better for holding application logic, and are generally classified under “Business Logic Services.” That’s not to say these technologies are mutually exclusive—you could put your business logic into a technology designed for presenting data (like a JSP application) if you wanted to—but the Presentation and Business Logic Services were designed with the idea of separating the code that drives the business from the code that controls how the end user interacts with data, and the intent of making applications easier to maintain and enhance, particularly in a collaborative environment where numerous programmers with different skill sets will be working on a single project.

Traditionally, the technology used for web-based Business Logic Services were Enterprise JavaBeans. Enterprise JavaBeans technology is the server-side component architecture for the Java 2 Platform, Enterprise Edition (J2EE) platform. EJB technology enables development of distributed, transactional, secure, and portable applications based on Java technology. The EJB specification is one of the several Java APIs in the J2EE platform. The specification details how an Application Server provides server-side objects with:

- Web Services
- Persistence
- Transactions
- Concurrency control
- Events using JMS (Java messaging service)
- Naming and directory services
- Security
- Deployment of components in an application server

Might want to disclude this from the list, or lower it. CORBA support isn’t exactly the shining star of te J2EE platform. Maybe add Web Services or something else.

If you search for a definition of “persistence” on the Internet, you’ll find so many different variations that you’ll probably be more confused after searching than you were before. The one that most closely matches my idea of persistence comes from the unlikely source <http://www.posc.org>: “That quality of an instance of data related to its existence beyond and outside of

the life of its creating process or the lives of other processes that use it. In this context, persistence is normally used in describing the life of objects relative to sessions; that is, whether or not they live beyond sessions.” See the discussion of TopLink later in this chapter under the section “Persistence Layer Services” for more information on Oracle’s strategic direction regarding persistence implementation for OC4J.

Additionally, the EJB specification defines the roles played by the EJB container and the EJBs, as well as how to deploy the EJBs in a container.

Central to the understanding of EJBs is the concept of patterns; Sun Microsystems defines a pattern as “a recurring solution to a problem in a context. A context is the environment, surroundings, situation, or interrelated conditions within which something exists. A problem is an unsettled question, something that needs to be investigated and solved. A problem can be specified by a set of causes and effects. Typically, the problem is constrained by the context in which it occurs. Finally, the solution refers to the answer to the problem in a context that helps resolve the issues.” A solution to a problem is not considered a pattern unless there is some way to show that the solution can be applied to recurring types of problems in the solution’s definition. Patterns are described by five main characteristics: context, problem, solution, forces, and consequences.

EJBs, however, are notoriously difficult to program and maintain. In response to this, Oracle created a business logic service framework called Oracle Business Components for Java (BC4J), which seeks to remove much of the complexity of EJBs while it continues to provide all of the EJBs’ functionality. With BC4J, developers can author and test object-oriented business logic in components that automatically integrate with relational databases, reuse business logic through multiple views of data, and access and update those views from servlets or JSPs.

You may be asking yourself why it is even necessary to store business logic in the middle tier: Don’t most Oracle-based applications store their business logic in database packages, procedures, functions, and triggers? There are several reasons for implementing business logic via BC4J:

- You do not want to mix object-oriented and relational paradigms in your application.
- You will be accessing the database only through clients using Business Components for Java. Your programming staff is skilled in Java and not in PL/SQL.
- You will be accessing both Oracle and non-Oracle databases.

Some companies may not want to be “locked into Oracle”—that is, they want to have database vendor freedom. This is generally a bad practice (for example, you won’t maximize the performance of your Oracle application), but if you have a generic product (i.e., Peoplesoft, SAP, etc.), this may make sense.

Designing, authoring, debugging, documenting, delivering, and maintaining a J2EE design pattern framework takes a significant amount of time and effort. BC4J implements patterns for developers, giving them a significant advantage in getting their J2EE applications built, deployed, and enhanced more quickly.

It is not a requirement, of course, to store your business logic in the middle tier. Traditionally, Oracle developers have stored their business logic inside the Oracle database in the form of packages, procedures, functions, triggers, and constraints. To assist in the development of web-based applications, Oracle provides an Apache module called `mod_plsql` that allows you to make requests to stored procedures in the database directly from the browser. As of Oracle 8.1.7, a group of packages called the PL/SQL Web Toolkit has been included as part of the built-in packages in the database. By using the toolkit, you can generate web pages; query, retrieve, and

display data from an Oracle database; dynamically calculate the contents of web pages; and much more. The details of using the PL/SQL Web Toolkit are discussed in Chapter 7.

Oracle also provides an installation script that can be run against databases prior to 8.1.7 to install the PL/SQL Web Toolkit packages.

Another place business logic can be stored is within an Oracle Form. Oracle Application Server 10g, when installed with the Business Intelligence and Forms option, can serve Oracle Forms over the Web to a client's web browser where the form is displayed as a Java applet containing the user interface for the forms run-time engine. When you submit a URL to launch an Oracle Forms-based application, the web listener accepts the request and downloads the Oracle Forms applet to your browser. The Oracle Forms applet then establishes a persistent connection to an Oracle Forms run-time engine. All processing takes place between the Oracle Forms applet and the Oracle Forms Services run-time engine, which seamlessly handles any queries or commits to the database. The Oracle Forms Server can be configured to use Application Server 10g's single sign-on (SSO) and OID capabilities to secure access. Oracle Forms can also be integrated into Oracle Portal to use its publishing capabilities. SSO and OID are discussed in Chapter 2, and Oracle Forms, their implementation in a web environment, their integration with Portal, and the configuration of the Forms Server are discussed in Chapters 3, 6, and 11. Figure 1-6 shows an Oracle Form served up to a web browser.

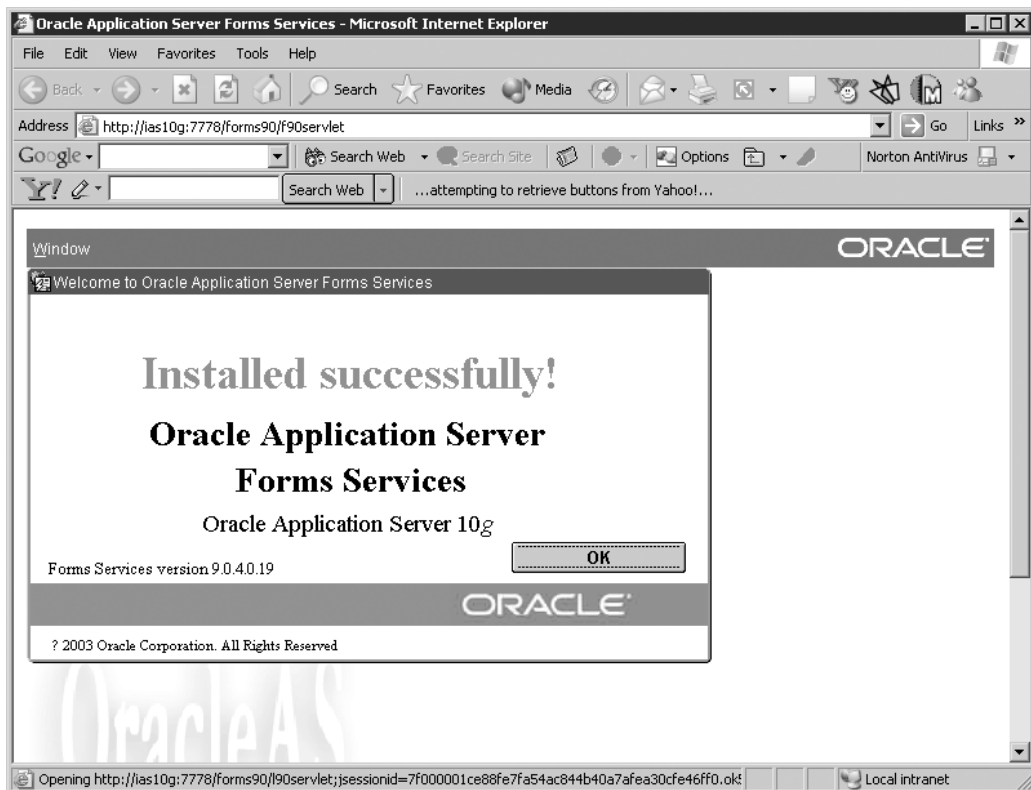


FIGURE 1-6. An Oracle Form viewed through a web browser

Presentation Services

As we have seen, technologies such as BC4J and Oracle Forms are good for writing, testing, and maintaining your business logic. There are other technologies, however, that are better suited for displaying your applications. “Presentation Services” deliver dynamic content to client browsers, supporting servlets, business intelligence, PL/SQL Server Pages, JavaServer Pages, Perl/CGI scripts, and PHP. You can use Presentation Services to build your presentation layer for your web applications. While Oracle puts these services into a separate group, you can think of them grouped with the Business Logic Services in that your application logic can reside within these components. We recommend, however, that you break your business logic into executable components, which can then be called or invoked by these services.

Apache JServ (`mod_jserv`) is a module for the Apache web server that implements Sun Microsystems’s Java Servlet API for running server-side Java code. Java servlets must be executed from within a Java Virtual Machine (JVM), and hence any servlet execution environment must include a JVM. The complexity of adding a JVM to the Apache server would degrade the Apache architecture, so the solution adopted for Apache JServ was to separate the JVM from the Apache server. This separation provides several benefits:

- It allows the use of any compatible JVM from any vendor. No modifications need be made to the web server when a change is made to the JVM (such as by an upgrade).
- It improves stability by separating the processes, allowing process-level protections by the operating system. If the JVM should crash or be mis-configured, the web server will still operate normally.
- It allows for advanced functionality, including automatic startup, separate JVMs for different configurations, and the capability to support load balancing on high traffic sites.

While the Apache module for handling servlet requests (`mod_jserv`) is still available in Oracle Application Server 10g, it is provided for backwards compatibility only. Oracle recommends converting your legacy applications to use OC4J.

OC4J is the core J2EE run-time component of Application Server 10g implemented as a module (`mod_oc4j`) in the Apache web server. OC4J is J2EE 1.3 compatible and runs on Java 2 Platform, Standard Edition (J2SE) distributions, making it easy to use and highly productive for developers, while at the same time offering outstanding performance and scalability for production environments. OC4J provides complete support for the technologies listed in Table 1-3.

OC4Js are discussed in Chapters 12, 13, and 15.

JSPs is a server-side technology that is an extension to the Java Servlet technology. JSPs have a dynamic scripting capability that works in tandem with HTML code, separating the user interface from content generation, enabling designers to change the overall page layout without altering the underlying dynamic content. Developers can benefit by using JSP technology without having to learn the Java language; they extend the JSP language through the use of simple tag handlers and easily maintain pages through the use of the JSP Standard Tag Library (JSTL). JSPs are discussed in Chapters 14 and 15.

PSPs is a technology provided by Oracle. It works on the same principles as JSPs, except that it uses PL/SQL as its scripting language. This can be beneficial for shops with minimal Java expertise. PSPs are compiled components executed as Oracle Stored Procedures. The Oracle PSP service includes the PSP Compiler and the PL/SQL Web Toolkit. Existing web pages can be made dynamic by embedding PL/SQL tags to perform database operations and display the results.

Java Technology	Version of Specification Supported in Oracle Application Server 10g
JSPs	1.2
Servlets	2.3
EJBs	2.0
JMS	1.0.2b
Java Database Connectivity (JDBC)	2.0
Simple Object Access Protocol (SOAP)	1.1
Web Services Description Language (WSDL)	1.1
Universal Description, Discovery, and Integration (UDDI)	2.0

TABLE 1-3. *Java Technologies Supported by OC4J*

Because PSPs are executed as stored procedures, it is important to note that the processing of PSPs occurs on the Oracle Database server. All of the other technologies discussed in this book execute on the Oracle Application Server. Chapter 7 discusses PSPs.

Business Intelligence Services

Business Intelligence Services can help you gain insight into internal business operations, customers, and suppliers. The basic Business Intelligence requirements usually include data quality, data analysis, and information access. The ultimate goal of Business Intelligence Services is to provide up-to-the-second business insight, most commonly provided with reporting tools. Oracle provides two enterprise-quality reporting tools: Oracle Reports and Oracle Discoverer. The choice of which tool to implement (or the decision to implement both) will be based on the needs and skill level of the employees in your organization. Both of these tools, in conjunction with Oracle Application Server 10g, provide the capability to easily make reports available over the Web.

Similar to the Forms Server discussed earlier, the Reports Server allows you to publish Oracle Reports on an internal company intranet, an external company extranet, or over the Web. In addition, the Reports Server can interface with a mail server to automatically deliver reports to the mailbox of selected users or groups. The Reports Server consists of the server component, run-time engines, and the servlet runner. When a client submits a request for a report, the Oracle HTTP Server web listener routes that request to the Oracle Reports Services server component. The server then routes the request to the Oracle Reports Services run-time engine, which runs the report. The report output is then sent back to the client through the Oracle HTTP Server web listener. Reports can be formatted as HTML, XML, PDFs (Adobe Acrobat), or as plain text for the user. Developers can also easily customize them for import into Microsoft Excel or any other commonly supported Multipurpose Internet Mail Extensions (MIME) document type. The Oracle Reports Server can be configured to use Application Server 10g's SSO and OID capabilities to secure access. Reports can also be integrated easily with Oracle Portal so that they can take advantage of Portal's publishing, security, and visual template features. Chapters 4, 6 and 11 discuss Oracle Reports and the Reports Server. Figure 1-7 shows an Oracle Report served up over the Web.

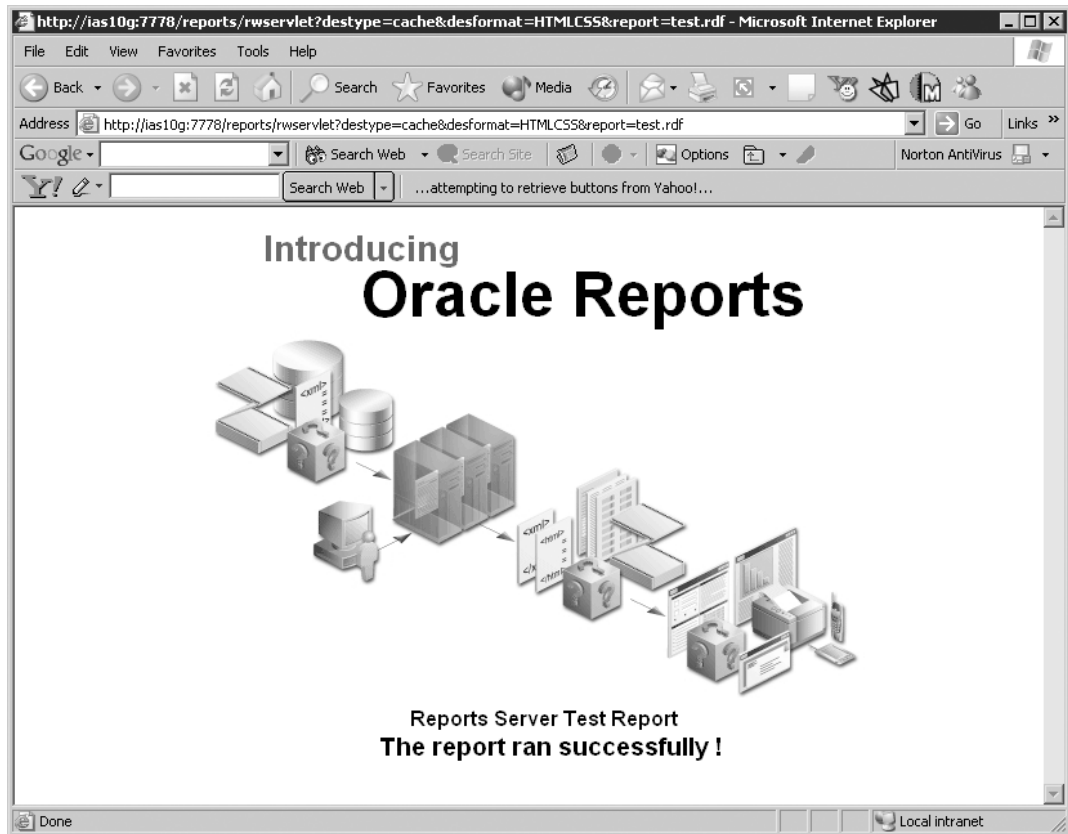


FIGURE 1-7. An Oracle Report viewed through a web browser

Oracle Discoverer is a reporting tool that gives ad-hoc query capabilities to your end users. Targeted at power users, Discoverer enables you to access information from the database and create dynamic reports that accept parameters and have drill-down or data pivoting capabilities, all without the need to learn SQL. Invoking the Discoverer workbook or worksheet over the Web can be made as simple as clicking a URL that invokes the Discoverer web component called Discoverer Viewer. This link can be constructed to invoke a particular workbook query automatically, giving the user the results in a web browser with the ability to interact with the query results to drill up or down, enter values into optional parameters, or to follow links to other workbooks or applications. Figure 1-8 shows an Oracle Discoverer workbook displayed in Discoverer Plus.

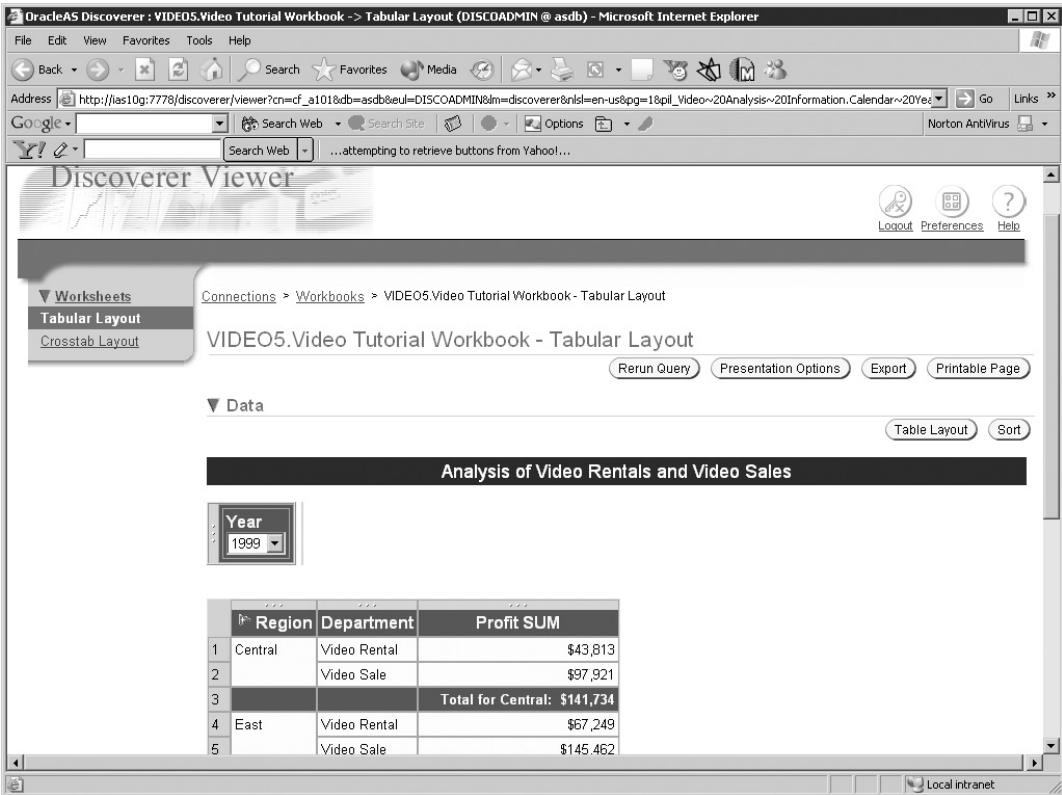


FIGURE 1-8. An Oracle Discoverer worksheet viewed through a web browser

There is no environment in Application Server 10g to develop Oracle Forms and Reports. The Forms Server and Reports Server components exist solely to take Forms and Reports and serve them up on the Web. Oracle Discoverer Server has the capability to serve Discoverer Workbooks and Worksheets over the Web, but it also has the capability to develop Workbooks and Worksheets using a web browser as its interface. This piece of the Discoverer Server, called Discoverer Plus, can be used as a replacement to the Oracle Discoverer Desktop. By using the Discoverer Plus web interface, you will no longer have to install the Discoverer Desktop software on every client that needs to create and manage workbooks and worksheets. The Discoverer Administrator program, which is used to administer End-User Layers (EULs) and Discoverer privileges, is still required. Figure 1-9 shows an Oracle Discoverer worksheet displayed over the Web in the Discoverer Plus environment.

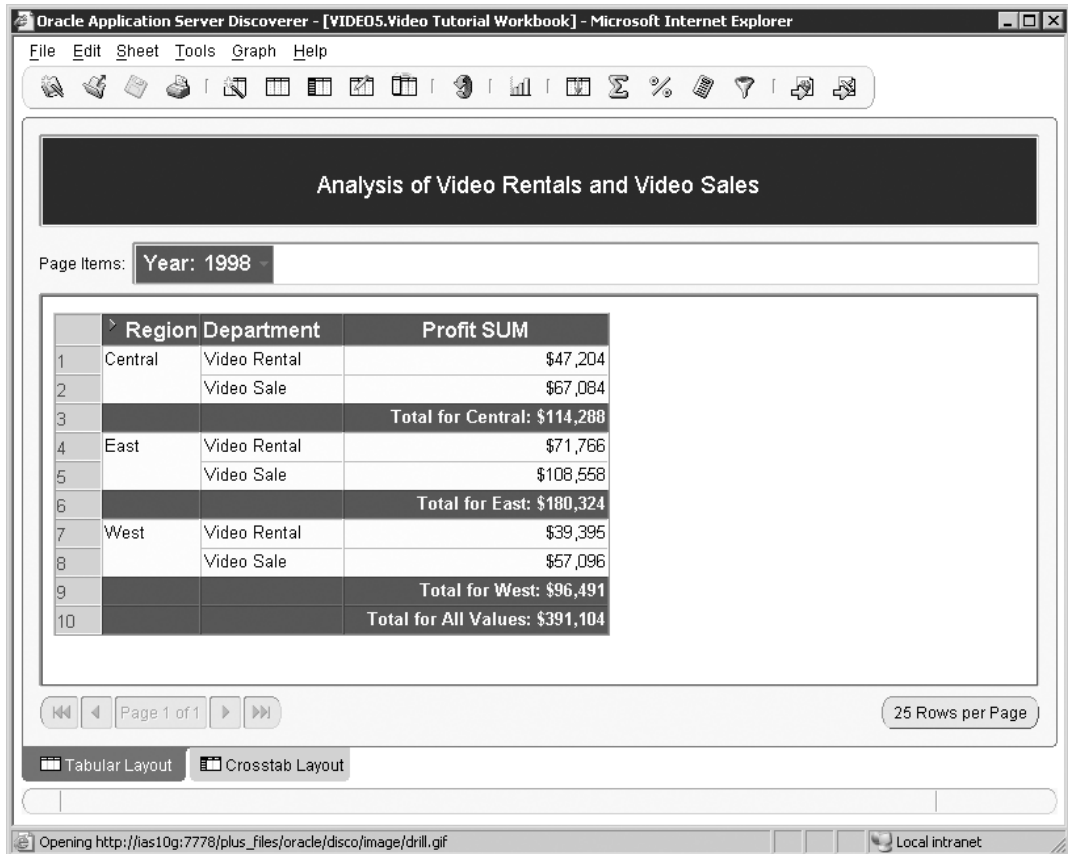


FIGURE 1-9. The Discoverer Plus development environment

Portal Services

An enterprise portal is defined as an application that enables organizations to unlock internally and externally stored information and provide users a single gateway to personalized information needed to make informed business decisions. By providing end users with relevant data from across multiple data sources and presenting that information in a context that is meaningful to the user, portals offer the promise of providing a competitive advantage through new business processes, increasing productivity by putting more power into the hands of business users, and increasing effectiveness through knowledge sharing and reduced search time.

Oracle Portal can handle and display virtually any type of data, giving you the ability to create portals that give users access to web applications, business documents, business intelligence reports, charts, and links to other forms of data that can exist both inside and outside your corporate intranet. Portal pages contain regions, which can hold items of content, such as spreadsheet or word processor files or portlets that provide access to web resources such as applications or other web pages. After portlets are written and tested, then can be published to a page, where they will be viewable by those users who have access to the page. Depending on the level of expertise of your end users and the amount of flexibility you're willing to give them, the Portal page can be set up so that the end users can modify their pages by adding, removing, or modifying content, similar to a site like <http://my.yahoo.com>. Portal is unique in the sense that its development environment is itself a Portal application. Once your developers and administrators become comfortable with the Portal environment, it will be easy for them to understand the mechanics of Portal development. Combined with the wizard-based approach to developing Portal applications, your developers can be productive in a very short period of time. Portal is discussed in Chapters 8 through 11. Figure 1-10 shows a Portal development screen.

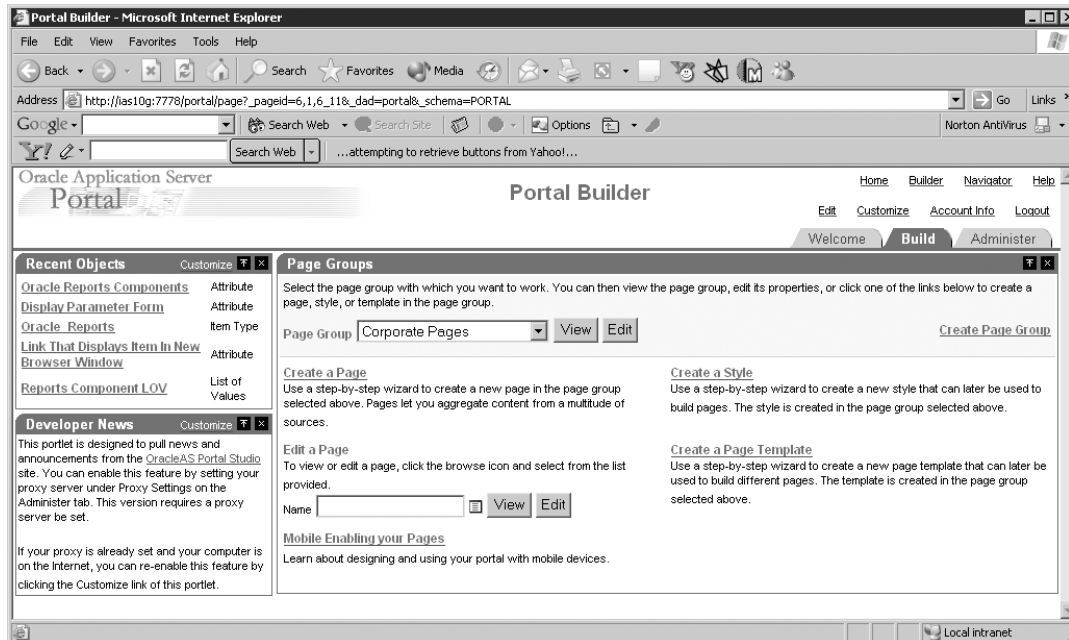


FIGURE 1-10. A screen of the Portal development environment

Web Services

Web Services are web-based applications that dynamically interact with other web applications using open standards that include eXtensible Markup Language (XML), Simple Object Access Protocol (SOAP) and Universal Description, Discovery, and Integration (UDDI). Such applications typically run behind the scenes, with one program “talking to” another (server to server). Using the UDDI discovery system, the goal of Web Services is to register the service on the Internet, allow an application to search for and find the service, and then to seamlessly exchange data with it. Web Services are either a request and response or a one-way style, and they can use either synchronous or asynchronous communication. The fundamental unit of exchange between Web Services clients and Web Services, of either style or type of communication, is a message.

Web Services enable software components to interact with each other around the world. Given Web Services use open, XML-based protocols that are lightweight and simple, their acceptance and use has gained a tremendous amount of support in recent years. A Web Service does the following things:

- *It exposes and describes itself.* A Web Service defines its functionality and attributes so that other applications can understand it. By providing a Web Services Description Language (WSDL) file, a Web Service makes its functionality available to other applications.
- *It allows other services to locate it on the Web.* A Web Service can be registered in a UDDI Registry so that applications can locate it.
- *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 (such as HTTP or SMTP).

Chapter 16 introduces and discusses XML. Chapter 17 discusses Oracle Web Services.

Developer Toolkits

To support application development and deployment, Oracle provides several toolkits containing libraries and tools. Oracle Application Server Developer Kits 10g provide APIs that enable you to develop Oracle Application Server Portal, Oracle Application Server Wireless, XML, and LDAP applications. Developer kits are used in development environments, not in staging or production environments. The code in a developer kit is deployed in production. It is possible to develop your own developer kits.

XML Toolkit

Oracle XML Developer's Kit 10g (XDK) is a set of components, tools, and utilities available in Java, C, and C++, and in Oracle Database 10g and Oracle Application Server 10g that ease the task of building and deploying XML-enabled applications. The production Oracle XDK is fully supported and comes with a commercial redistribution license. Oracle XDK consists of the following components:

- **XML parsers** XML parsers create and parse XML using Document Object Model (DOM), Simple API for XML (SAX), and Java API for XML Processing (JAXP) interfaces.

You're able to directly access XMLType in the Oracle Database 10g with unified C DOM interfaces. DOM support includes the 3.0 specification.

- **eXtensible Stylesheet Language Transformation (XSLT) processors** XSLT processors transform or render XML. They now include XSLT 2.0 Java support.
- **XSLT VM** XSLT VM and Compiler provides high performance C XSLT transformation engine using compiled stylesheets.
- **XML schema processors** XML schema processors support XML schema validation. It now includes validator interfaces for stream-based processing.
- **XML JavaBeans** XML JavaBeans parse, transform, diff, retrieve, and compress XML documents via Java components.
- **XML Class Generator** Now supporting JAXP, XML Class Generator automatically generates classes from Document Type Definitions (DTDs) and XML schemas to send XML from web forms or applications.
- **XML SQL Utility** XML SQL Utility generates XML documents, DTDs, and XML schemas from SQL queries in Java and inserts XML documents into Oracle databases.
- **XSQL Servlet** XSQL Servlet combines XML, SQL, and XSLT in the server to deliver dynamic web content and build sophisticated database-backed web sites and services.
- **XML Pipeline Processor** XML Pipeline Processor invokes Java processes through XML control files.
- **TransX Utility** TransX Utility makes it easier to load globalized seed data and messages into Oracle databases.

Chapter 16 introduces and discusses XML.

Content Management Toolkit

Oracle Content Management SDK provides a set of Java APIs for folders, versioning, check-in/check-out, security, searching, extensible metadata, and other standard operations for the development of content-oriented applications. You can access your content with your choice of tools through the following protocols:

- **File Transfer Protocol (FTP)** FTP allows the transfer of one or more files from one machine to another across the Internet.
- **Server Message Block (SMB)** SMB is a protocol for sharing files, printers, serial ports, and communications abstractions such as named pipes and mail slots between computers.
- **Web Distributed Authoring and Versioning (WebDAV)** WebDAV is a mechanism to support collaborative development of web pages.
- **Network File System (NFS)** NFS is a protocol suite developed and licensed by Sun Microsystems that allows different makes of computers running different operating systems to share files and disk storage.

- **Apple Filing Protocol (AFP)** AFP is Apple's network protocol providing file server/client access in an AppleShare network.
- **Internet Messaging Access Protocol 4 (IMAP4)** IMAP4 is a network standard that allows users to manage e-mail messages and folders from multiple locations and systems. Users can choose to store their messages on their own local computers (or clients), or on a server.
- **Simple Mail Transfer Protocol (SMTP)** SMTP is a protocol used to send e-mail on the Internet. It is a set of rules regarding the interaction between a program sending e-mail and a program receiving e-mail.

Developers can also create custom/proprietary protocol servers. Oracle Content Management SDK is fully integrated with Oracle Text, Oracle Workflow, Oracle *interMedia*, Oracle Advanced Queueing, and other Oracle API products. It is also integrated with Oracle platform infrastructure products such as Oracle Internet Directory. It was formerly known as Oracle Internet File System.

Oracle Application Server MapViewer Toolkit

Geographic and location data are managed in a native type within Oracle Database 10g. Oracle provides the following technologies to make use of geographic information:

- **Oracle Locator** Oracle Locator provides core location functionality to support a variety of Location Based Services (LBS) and third-party Geographic Information Systems (GIS) solutions.
- **Oracle Spatial** Oracle Spatial is a database option for Oracle9*i* and Oracle Database 10g Enterprise Edition that provides advanced spatial features to support high-end GIS and LBS solutions.
- **Oracle MapViewer** Oracle MapViewer is an Oracle Application Server Java component and JDeveloper extension used for map rendering and viewing geospatial data managed by Oracle Spatial or Locator.

Wireless Toolkit

Oracle Application Server Wireless is a comprehensive and flexible wireless and voice platform. Enterprises can deploy wireless browser-based applications, voice applications, notifications, Java 2 Micro Edition (J2ME), and two-way messaging applications. It helps enterprises and service providers efficiently build, manage, and maintain wireless and voice applications. Some of the main features of Application Server Wireless include:

- **Multichannel server** A multichannel server enables applications to be accessed through multiple delivery methods such as Short Message Service (SMS), voice access, Wireless Application Protocol (WAP), Pocket PCs, etc. Developers can focus on creating mobile applications for any channel in one, future-proof open standards language.
- **J2ME support** J2ME provides a lightweight run-time environment for mobile devices enabling client-side development based on industry standards instead of proprietary device interfaces.

- **J2ME Developer's Kit** The J2ME Developer's Kit offers the capability to extend Web Services to J2ME devices.
- **J2ME Provisioning System** The J2ME Provisioning System is a web-based application manager that allows users to upload J2ME applications to the database repository for efficient management and secure storage.
- **Notifications and multimedia messaging** Notifications and multimedia messaging enhance intelligent messaging with new functionality for actionable alerts, message adaptation, and failover delivery control.
- **Web clipping** Web clipping allows clipping and scraping of existing web content to create wireless applications that reuse existing PC browser-based applications.
- **Location services** Location services give access to the full LBS functionality, such as user positioning, geocoding, mapping, driving directions, and business directory lookup.

To develop mobile applications, Oracle has created the Wireless Developer's Kit, a download that provides developers with documentation, samples, code templates, and wizards for common mobile user interface constructs. The Wireless Development Kit can be used on any PC or laptop, connected or disconnected, to build and test wireless and voice applications. Developers can use any IDE to develop their mobile applications, including Oracle JDeveloper. The JDeveloper Wireless Extension includes built-in simulators allowing developers to preview the mobile application, code templates, wizards, code insight, and automatic deployment to Oracle Application Servers.

Wireless applications can be deployed in the following ways:

- **Hosted services** Oracle provides a hosting service that maintains logins and allows access to a secure repository. This is the fastest way to implement your wireless development.
- **In-house deployment** Corporations deploy and maintain their own firewall-based security infrastructure.
- **In-house with hosted Web Services** Corporations install the Oracle9i Application Server Wireless infrastructure within their own premises while using the desired features as a Web Service.

Security is provided through a number of different protocols depending on the type of wireless application:

- **Browser-based applications** 802.11: HTTPS, WAP 1.2, WAP 2.0.
- **Short Message Service (SMS)** General Packet Radio Service (GPRS) security; applications can use symmetric shared encryption keys.
- **E-mail** Security is usually guaranteed in the domain if the e-mail server and application server are located at the mobile operator. If the servers reside at the enterprise, a secure channel—SSL, Transport Layer Security (TLS)—needs to be used.

- **Voice** HTTPS can secure the channel between the voice gateway and the application server.

Oracle Application Server Wireless supports the following messaging capabilities:

- **Push messaging** Push messaging sends messages to mobile users using the notifications engine or a custom application.
- **Mobile-initiated pull messaging** Mobile users send a message to invoke a server-side application, which replies by sending a message to the mobile user.
- **Server-initiated pull messaging** The server sends a message to which the mobile user can respond.

Out-of-the-box, the following channels and protocols are supported in Oracle Application Server Wireless:

- **SMS, Enhanced Messaging Systems (EMS), SmartMessages** Short Message Peer to Peer or SMPP (Logica, CMG, Comverse), UCP (CMG), CIMD (Nokia), Nokia Global System for Mobiles (GSM) phone modems with data cable, Mobileway V-SMSC, Vodafone VVSP
- **MMS** SMTP (Ericsson, LogicaCMG), External Application Interface or EAIF (Nokia), MM7 (Ericsson, LogicaCMG)
- **E-mail** IMAP, POP3, SMTP
- **Fax** Captaris RightFax
- **Voice notifications** VoiceGenie Voice Gateway
- **Pagers** Wireless Communications Transfer Protocol (WCTP)
- **Instant Messaging (IM)** Jabber (also as Gateway to AOL, MSN, Yahoo!, ICQ, etc.)

Portal Development Kit

The wizard-based approach to developing Portal components allows developers to create and deploy web-based components and applications extremely quickly. There are, however, serious limitations to any type of wizard-based development environment. The ability to customize and extend the functionality of components generated through a wizard-based development environment is arduous at best, and impossible in many cases.

The Portal Development Kit (PDK) provides developers with tools and articles that provide advanced techniques for extending the functionality of Portal components beyond what is available through the component wizards in Portal. The PDK is a framework that enables development of portlets for seamless integration with Oracle Application Server Portal. Using the PDK, developers can create portlets either as PL/SQL stored procedures (database providers) or in any other web language, including Java, Web Services, XML, Application Server Pages (ASP), Perl, etc. (web providers). The PDK includes the following:

- **PDK Services for Java (JPDK)** JPDK provides APIs for creating Java-based and Web Services-based portlets.
- **Java Portlet Container** Java Portlet Container provides a run-time environment for portlets coded to the JPDK Portlet API.
- **OmniPortlet** OmniPortlet provides wizards that enable Portal page designers to publish SQL, XML, Web Service, spreadsheet, and existing web page data directly to a Portal page.
- **Web Clipping Portlet** The Web Clipping Portlet enables page designers to collect existing web content into centralized Portal pages as a means of content consolidation. Page designers can clip page content from an existing web site and deliver it as a portlet to a Portal page.
- **PL/SQL Development Kit** The PL/SQL Development Kit provides APIs for developing database providers.
- **Utilities** Utilities simplify the development and testing of portlets.
- **Java Portlet Wizard** The Java Portlet Wizard is a JDeveloper Add-in that provides a wizard-based utility for creating JPDK-based portlets.
- **PL/SQL Generator** The PL/SQL Generator is a hosted utility that simplifies creation of PL/SQL-based portlets.
- **Test suite** The test suite includes the Provider Test and Test Harness utilities for testing web providers without having an installation of the Oracle Application Server Portal.
- **Samples** This includes various code samples highlighting development solutions.
- **Articles** This includes development white papers explaining development issues in depth.

Go to <http://portalstudio.oracle.com> for more information about downloading and installing the PDK.

Persistence Layer Services

When developing Java applications that interface with Oracle, the integration between Java's object-oriented features such as object references, business rules, complex relationships and inheritance, and Oracle's relational database features such as stored procedures, tables, rows, columns, and foreign keys (referred to as object-relational[O-R] mapping) requires an effort that is commonly underestimated by both developers and project managers. The problem of bridging object-oriented and relational technologies is referred to as object-relational impedance mismatch and the fundamental differences in modeling, design, and skill sets can prolong development work. This can lead to deployed applications that are, at best, difficult to maintain and, at worst, unreliable and impossible to enhance. In addition to these issues, developers also need the ability to translate object-oriented data into relational data, referred to as O-R mapping. A product that easily enables the integration of relational database and Java technologies, while allowing

34 Oracle Application Server 10g Web Development

database designers and Java developers to maintain database and application design principles, would be invaluable. Oracle recognized the developer's need for these features and has created a product that not only addresses all of the complex issues mentioned above, but also adds features such as:

- A query framework that allows developers to define queries using various technologies
- A transaction framework that provides object-level transaction support
- Performance enhancements that includes a fully-configurable cache that ensures object-identity

Oracle Application Server TopLink integrates the object and relational worlds. It allows applications and application developers to manage Java objects using relational databases. TopLink is robust enough to work with any Java-supporting database and any application server that supports Java and J2EE containers. Some of the major features of Oracle Application Server TopLink include:

- **The Oracle Application Server Mapping Workbench** The Oracle Application Server Mapping Workbench is a visual tool that creates and maps metadata describing the relationship between Java classes and relational tables.
- **Advanced mapping support** Advanced mapping support includes support for the following mappings:
 - Direct-to-field
 - One-to-one
 - Variable one-to-one
 - One-to-many
 - Many-to-many
 - Aggregate Object
 - Aggregate-collection
 - Transformation
 - Object type
 - Type conversion
 - Serialized object
 - Direct collection
 - Nested tables
 - VArrays and OREFS
 - Multitable
 - Inheritance

- **Object caching** Object caching stores data returned as objects from the database for future use, minimizing database and network access. The following cache types are supported:
 - **Soft** Objects are maintained through garbage collection sweeps until memory is at a premium.
 - **Hard** Objects are not garbage collected.
 - **Weak** Objects exist in the cache as long as they are being used.
 - **Full** Objects are cached but not removed.
- **Query Flexibility** Developers can define queries using:
 - An object-oriented expression framework
 - Query By Example (QBE)
 - Enterprise Java Beans Query Language (EJB QL)
 - SQL
 - Stored procedures
- **Object-level transaction support** Transactions are supported through the “Unit of Work,” a Java-based transaction. During a Unit of Work, the application modifies business objects; when the Unit of Work is committed, Oracle Application Server TopLink updates the database, based on what has changed, thereby executing the minimally required SQL.

Oracle Application Server TopLink consists of three main components:

- **Development tools** Development tools consist of the Oracle Application Server Mapping Workbench (which creates and manages the mapping metadata that describes the relationship between Java classes and relational tables) and the Oracle Application Server Sessions Editor (which describes how TopLink communicates with the datasource at run time by creating the session.xml file).
- **Run-time tools** Development tools consist of the web client, an interface to interact with any Oracle Application Server TopLink server session, and the Performance Profiler, providing performance tuning and run-time diagnostics.
- **Class libraries** Class libraries consist of the TopLink API, which is called at run time to retrieve and store Java objects.

Caching Services

Web caching is the process of keeping frequently used pages in memory so that the HTTP server can display them quickly instead of repeatedly processing requests for those URLs. By implementing caching, clients experience faster page retrievals and the load on the web server is greatly reduced. It is one of the most effective ways to scale your application without the need for r-coding or purchasing new hardware.

Oracle Application Server 10g contains a component called the Web Cache. It is a content-aware service that improves the performance, scalability, and availability of web sites. Application Server Web Cache 10g uses caching and compression technologies to optimize application performance and more efficiently utilize low-cost, existing hardware resources. Built-in workload management features ensure application reliability and help maintain quality of service under heavy loads. And new in this release, end-user performance monitoring features provide unparalleled insight into end-user service levels. The real power of Application Server Web Cache is its capability to cache both static and dynamically-generated pages.

The Web Cache can be configured to run on its own server or on the middle-tier server (see Figure 1-11). In either configuration, the Web Cache is placed in front of the web server to cache pages and provide content to those browsers that request it. If the Web Cache, acting as a virtual server or virtual request router, can satisfy the request, it will provide content to the client. If the requested content is not cached by the Web Cache or has been marked invalid for any reason, the content is retrieved from the web server and cached in the Web Cache. Application Server Web Cache allows you to define invalidation rules, which can be used to control the amount and types of cached content in your server. Some of the key benefits of Application Server Web Cache can be measured by dramatic improvements in the following areas:

- **Resource usage** Higher throughput and scalability
- **User experience** Faster response times without sacrificing personalization
- **Availability** Intelligent workload management
- **Productivity** No need to roll your own cache means faster time-to-market
- **Bottom line** Reduced infrastructure load translates into cost savings
- **Intelligence** Better visibility into end-user service levels

Some of the key features of Application Server Web Cache include:

- Efficient use of low-cost hardware
- Fine-grained cache control
- Workload management and reliability
- End-user experience management
- Advanced networking
- Single-vendor manageability and integration
- Flexible deployments

The new features of Application Server Web Cache include enhancements in the following categories:

- **End-user performance management** This is the most significant new Web Cache feature. Administrators can configure Application Server Web Cache to measure

end-user response times for individual URLs, sets of URLs, or even entire web-based applications, regardless of whether the URLs are cached. The Analyze functionality lets you view detailed reports in context by group, URL, domain, visitor, or application as well as in a daily, weekly, or monthly context. Further drill-downs provide administrators with response time and load distribution information to help balance web server resources.

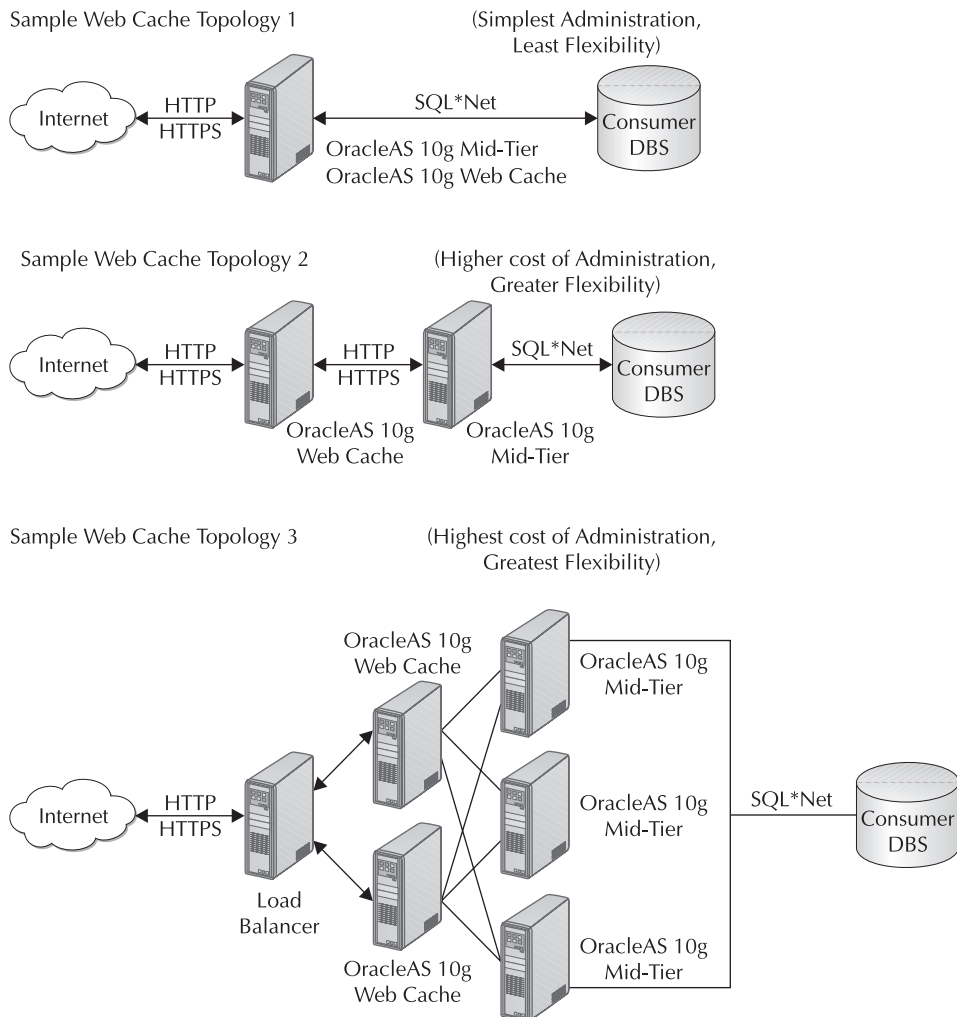


FIGURE 1-11. Possible Web Cache topologies

- **Security** Web Cache now supports applications that require client-side SSL certificates for PKI-based authentication. Oracle Application Server now supports nCipher's BHAPI-compliant hardware for deployment on servers running Web Cache and/or Oracle HTTP Server.
- **Caching** Previously, administrators could either cache one version of a page for all browsers, or they could cache one version for each browser type and version. Now, administrators can customize the caching rules to define groups of browsers that will share a cached version of a page.
- **Invalidation** The 9.0.4 release of Web Cache introduces an inline invalidation mechanism as an additional means of managing content freshness. Inline invalidation provides a useful way for origin servers to "piggyback" invalidation messages on transactional responses sent to Web Cache. In previous releases of Web Cache, the URL-based cache key was the unique identifier for a cached document. Invalidation requests needed to specify either exact URLs or a set of URLs and headers matching a regular expression in order to invalidate cached objects. Because it can be difficult for applications to map URLs to the underlying data used to generate those URLs, Web Cache invalidation has been extended in 9.0.4 to support search keys. Cached objects can now be associated with multiple application-specified search keys, with the URL-based key being the primary key.
- **Compression** The Web Cache compression engine now supports self-describing compression policies and more compressible content by compressing documents containing session-encoded URLs, Edge Side Includes (ESI) tags, or the `<!--WEBCACHETAG-->` and `<!--WEBCACHEEND-->` tags.
- **Load balancing and request routing** Support for session binding in a cache cluster allows Web Cache to bind a user session to a particular origin web server. This feature is used to consecutively route requests for a unique session to the same origin server, allowing stateful load balancing.
- **HTTP protocol support** Web Cache now supports chunked transfer encoding. HTTP 1.0 (with connection keep-alive) was used in prior releases of Web Cache, which caused problems for servers that generated dynamic content with unknown content length. Application Servers's support for HTTP 1.1 allows for persistent connection to the Web Cache, even when content length is undetermined.
- **Usability and manageability** Web Cache now includes improved access logging, event logging, and diagnostics, reporting on popular cache misses, integration with Oracle Process Manager and Notification (OPMN), and dynamic configuration for select parameters.

Figure 1-12 shows the administration screen for the Web Cache server. The Web Cache is an administration function, and is therefore only mentioned briefly in this book. The Database Cache, introduced in Oracle9i Application Server Release 1, has been discontinued.

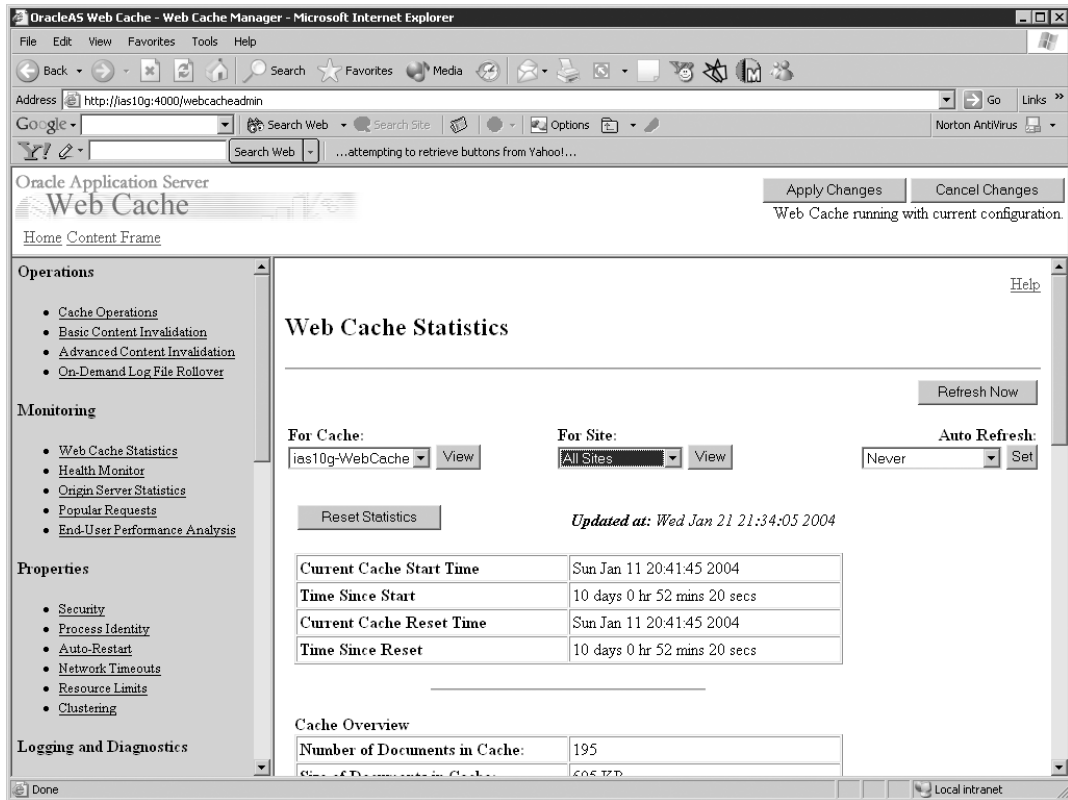


FIGURE 1-12. The Web Cache administration screen

System Services

Oracle provides numerous system services to help make your interaction with Oracle Application Server 10g easier. Most of the system services made available through the Enterprise Manager Application Server Control web pages are of interest mainly to administrators, but developers will benefit from an understanding of the Application Server Control pages when the deployment of applications is discussed in Chapter 15.

Enterprise Manager 10g Grid Control

Oracle Application Server 10g is supported by a web-based management system, Oracle Enterprise Manager 10g. Enterprise Manager provides both a simple, out-of-box application server administration interface and a comprehensive system for managing and monitoring large-scale Oracle systems.

The Application Server administration interface, known as Application Server Control, is automatically installed with Application Server 10g. It is designed to help you manage your individual application server instances, application server farms (an Application Server grouped with its corresponding infrastructure), and Oracle Application Server clusters. From the Application Server Control, you can monitor the entire Oracle Application Server platform—from J2EE to Portal and Wireless to Business Intelligence components—and perform administrative tasks, such as configuration changes to your application server components and real-time monitoring of all services configured in your Application Server 10g instances. With Enterprise Manager 10g Application Server Control, administrators can:

- Manage and configure application server components
- Monitor server performance and application server logs
- Create and configure J2EE services
- Deploy and monitor J2EE applications

The complete Enterprise Manager 10g Grid Control product is provided as an optional installation with the application server. The two can be used together as an integrated system for complete application server system management. Grid Control is a web-based system for central management of Oracle products, host systems, and applications. It provides a central console for monitoring distributed application servers and is integrated with the Application Server Control for performing administration operations. With Grid Control, application server administrators can:

- Monitor the status and performance of servers across the business enterprise
- Maintain service-level requirements for web applications
- Maintain system and software configurations
- Manage the entire system stack employed by the application server system

Figures 1-2 and 1-3 are screens for administering the middle tier and infrastructure instances, respectively, via the Enterprise Manager Application Server Control web site.

Development Tools

Does your web application have to be written in Java? Many developers believe that to be true, but as we will see, there are many options available to us as Oracle developers. Oracle provides numerous tools that allow us to leverage existing (or legacy) code into applications that can be deployed over the Web with minimal effort. In general, there are two types of web applications:

- **HTML-based** Application code (residing on the server or in an Oracle database) producing a series of HTML pages. There is no limitation as to what language is used to generate these pages.

- **Applet-based** Java code downloaded to the client's browser and executed there. Applets give the developer full control of the interface, but at a price—the time needed to download the applet may be prohibitive for all but the most trivial of applications.

Except for Discoverer Plus and Portal, there is no development environment provided with Oracle Application Server 10g. You will have to make a decision as to what technology to use when beginning your development. There is no “one” tool that is perfect for solving all of your development needs and virtually all organizations use a combination of different tools and technologies. The most valuable developers in modern IT shops are those that are fluent in numerous technologies.

Choosing the right development tool is not an easy task and it is not a decision to enter into lightly. Human nature being what it is, people tend to gravitate to things they know and feel comfortable with. Some hard questions need to be answered before deciding on a technology, or a combination of technologies:

How Will Your Application Be Accessed?

It is not enough to deploy an application over the Web and not worry about how users will access it, although this was the original promise of browser-based applications. There are three basic ways of delivering your application, and each one of these will affect the development decisions you will need to make:

- **Intranet** In this environment, users access the application over a network inside a company or organization, which uses software like that used on the Internet, but the intranet is for internal use only and is not accessible to the public. This option gives you the most flexibility in your development decisions as there is a good likelihood that all users of your application will have fast network connections and that your organization has standardized on a single browser.
- **Extranet** In this environment, users access the application over a private network that uses the Internet protocols and the public telecommunication system to share a business's information, data, or operations with external suppliers, vendors, or customers. An extranet can be viewed as the external part of a company's intranet. Here, your development options begin to decrease, as there will be questions regarding browser types and firewall issues, which may block certain types of communications between the clients and server.
- **Internet** In this environment, users can access the application from anywhere in the world. This option has the least amount of flexibility regarding your development decisions as the application will have to be “generic” enough to support all types of browsers and all types of access (dial-up, broadband, etc.). It will be almost impossible to deliver any applet-based application via this method.

How Complex Is the User Interface?

Applets give the developer the greatest control over the look and feel of the user interface. As mentioned before, this comes at a great price: namely, the time required to download the

applet to the browser. In an intranet or extranet setting, this may be acceptable, but it will be virtually impossible to serve applets over the Internet. If the decision is to go with an HTML-based application, whether users can be productive with a standard interface or if sophisticated HTML tools (such as Dreamweaver) will need to be utilized to create something more complex must be determined.

What Types of Users Will Be Using Your Application?

In general, users can be broken into three categories:

- **Casual users** For these users, speed is the most important factor. Casual users will probably demand that an application start with minimal load time and provide data access immediately. These users will usually be satisfied with HTML-based applications.
- **“Heads-down” users** These users work with an application for an extended period of time, and a startup period of a minute will probably be acceptable in a tradeoff to the productivity gains realized throughout the day. These users may benefit from an Applet-based application.
- **Power users** These users demand applications they can customize: either the interface of the application or the types queries and data used in the application. For these users, flexibility and interaction are the most important features, which will probably require an Applet-based application.

Oracle Forms

Oracle Forms is one of Oracle’s oldest and most mature development products. It has been traditionally used to create client/server applications. It is similar in design and philosophy to Microsoft’s Visual Basic development environment. Countless Forms-based applications are still in production environments around the world today.

As development efforts started moving towards web and browser-based deployments, the demand to create an environment that still supported Forms-based development (preserving legacy code and knowledge) and introduced web deployment (eliminating the high cost of maintaining clients with multiple operating systems, nonstandard configurations, etc.) began to grow. Oracle began developing technologies that allowed existing Forms to be served over the Web. These initial efforts were called “cartridges” and were designed to “plug into” the web server, enhancing its functionality.

Oracle Application Server 10g includes a service called the Forms Server that is installed with the Business Intelligence and Forms option. It handles serving Oracle Forms over the Web, gathering metrics to assist in administration and tuning, and integration with Oracle’s Identity Management features. There are numerous configuration parameters to modify virtually every aspect of the Forms environment on the Web. These parameters are discussed in detail in Chapter 6.

Starting with Oracle Forms 9i (the current version is Oracle Forms 10g), Oracle no longer provided the Forms run time to deploy Forms in a traditional client-server environment. The only way to test an Oracle Form you have developed is to run it as a web form through a forms server provided with the Oracle Forms 10g development environment. It was the final indication that

Oracle believed (and still firmly believes) that web-based applications is where organizations will be putting their development resources in the future. Chapter 3 discusses the Oracle Forms development environment. Chapters 6 and 11 discuss integrating Forms with Oracle Application Server 10g and Oracle Portal, respectively.

Oracle Reports

Much of what has been said about Oracle Forms also applies to Oracle Reports. It is a mature and stable product that allows developers to create reports quickly and easily. It also suffers from the high administration costs of deployment as Reports run-time libraries need to be installed and maintained on every machine that needs to run an Oracle Report. In response to this, Oracle developed cartridges for the Oracle Application Server (before Oracle standardized on Apache as its web server), which eventually became the Reports Server engine that is included with Oracle Application Server 10g today. The Reports Server can be configured to integrate with Oracle's Identity Management features for security. Like Oracle Forms 9i, every version of Oracle Reports since 9i (the current version is Oracle Reports 10g) no longer provides the Reports run-time environment to run Oracle Reports in a traditional client/server environment. The only way to test a Report you have developed is to run it as a web report through a reports server provided with the Oracle Reports 10g development environment.

Multiple Reports servers can be defined for each middle tier in Oracle Application Server 10g. This gives another layer of scalability for your environment. If you wish to integrate Reports with Oracle Portal, calendars can be created in Portal to limit the access times for certain reports, Report servers, and printers. You can also use the Portal security model to limit access to any of these components. Chapter 4 discusses the Oracle Reports development environment. Chapters 6 and 11 discuss integrating Reports into Oracle Application Server 10g and Oracle Portal, respectively.

Oracle Discoverer

As mentioned earlier in this chapter, Oracle Discoverer server contains a component to view and interact with Discoverer worksheets over the Web (Discoverer Viewer) and a component to create and modify Worksheets (Discoverer Plus). The ad-hoc capabilities of Discoverer can allow your development staff to off-load report design and generation to power users. By putting the full capabilities of the Discoverer desktop on to the Web, Oracle makes the implementation of Discoverer a simple experience for most organizations.

Discoverer also integrates seamlessly with Oracle Portal via an internal portlet provider provided by Oracle called the Discoverer portlet. This portlet allows you to define the workbook and worksheet you wish to display, any parameters to be passed to the worksheet, and a refresh options setting that allows you to specify AUs: Okay to make active here? the length of time a particular report will be cached (in Portal's Cache, not the Web Cache). The Discoverer worksheet becomes a portlet and can have any of Portal's security and display properties applied to it. Chapter 5 discusses viewing and creating Discoverer workbooks and worksheets via the Web, and Chapters 6 and 11 discuss how to integrate Discoverer with Oracle Application Server 10g and Oracle Portal.

Java Tools

An Integrated Development Environment (IDE) is a development environment in which the tools have been integrated to collaborate with each other (e.g., the output of one tool can be used as

44 Oracle Application Server 10g Web Development

the input to another tool). Attempting to build Java applications without an IDE using the free Java compiler provided in the Java Software Development Kit from Sun Microsystems is possible, but it is a less than optimal environment. Oracle's Java IDE, JDeveloper, has so many essential tools, only the hardiest of Java programmers would ever attempt to build production-quality Java applications without it. JDeveloper will be covered in Chapter 13, but a quick list of its prominent features include:

- Three profiling modes that enables you to create a statistical analysis of the performance of your application with respect to its functionality both at compile time and run time, its use of memory in the Java heap, and the occurrence and duration of various events.
- A public Extension SDK that enables its development environment to be extended and customized.
- Native support for SQL, PL/SQL, and XML. This support includes syntax highlighting and code insight, as well as PL/SQL development, PL/SQL debugging, and SQL tuning. Additionally, JDeveloper provides direct access to the database, allowing you to view, create, modify, and delete tables, views, triggers, indexes, sequences, and more.
- Robust debugging support for both Java and PL/SQL. Debugging in these two environments is seamlessly integrated when using an Oracle9i database, providing the capability to step from Java code directly into PL/SQL code within the same debugging session.
- Class diagrams. The goal of class modeling is to visualize classes or components, and the relationships between them, that comprise all or part of a system design. Classes and components modeled on a class diagram are synchronized to ensure that changes made to modeled elements are reflected in their implementation files, and that changes to the implementation files are visualized on the diagram.
- Activity diagrams. Processes performed in a business or system can be visualized using modeled activities, flows, and states on activity diagrams. An activity can represent a single process, or it can represent a subactivity model (that is, the activity can be broken down into a set of subactivities which themselves form an activity model).
- All aspects of EJB development from conception to implementation. EJB support includes: modeling and wizard-based development of EJB session, entity, and message-driven beans, the ability to add, edit, and delete EJBs and EJB properties using the EJB Module Editor, reverse-engineering of database tables and foreign key relationships as Container-Managed Persistence (CMP) entity beans and Container Managed Relationships (CMRs), the ability to test EJBs locally in the IDE, and the ability to check EJBs for deployment errors and inconsistencies using the EJB Verifier.

- J2EE deployment, which allows developers to package and assemble J2EE modules into standard archives such as EJBs, JARs, WARs, and EARs. These archives can be deployed with a single-click to Oracle Application Server Containers for J2EE (OC4J).
- A built-in J2EE Applications Framework. Oracle Business Components for Java (BC4J) is a standards-based, server-side framework for creating scalable, high-performance J2EE applications. The framework provides design-time facilities and run-time services to simplify the task of building, debugging, customizing, and reusing business components.
- An application Development Framework (ADF). This is the most exciting new feature of JDeveloper for developers. The ADF is Oracle's solution to the ever-increasing complexity of the J2EE platform. Based on the Model-View-Controller (MVC) architecture, Oracle ADF lets application developers focus on the business domain rather than on the underlying technologies. By using visual, declarative, and guided-coding techniques, the framework allows application developers who are not J2EE experts to quickly become productive. The framework is based on industry standards allowing applications built with ADF to be deployed on any J2EE server and connect to any SQL database.
- Integration of the TopLink Mapping Workbench. Oracle JDeveloper 10g seamlessly integrates with the Mapping Workbench (see Figure 1-13), giving developers the ability to:
 - Automatically map descriptors
 - Generate database schemas from object models
 - Generate object models from database schemas
 - Generate both CMP and Bean-Managed Persistence (BMP) entity beans
 - Import an object model from any IDE or UML modeling tool
 - Connect and interact with any relational database with a JDBC-compliant driver

HTML DB

HTML DB occupies a unique space in the world of Oracle web development. It is a web development tool, but it is not part of the Oracle Application Server 10g product stack. The Oracle HTML DB engine is stored in the database, and its accessible via the HTTP server that comes standard with the Oracle 10g database. It is intended primarily for simple web-based applications that do not require the sophisticated features provided with Application Server 10g. HTML DB is composed of three main pieces:

- **Application Builder** Application Builder helps you assemble an HTML user interface on top of database objects. The Oracle HTML DB engine takes care of presenting your application using templates and UI elements.

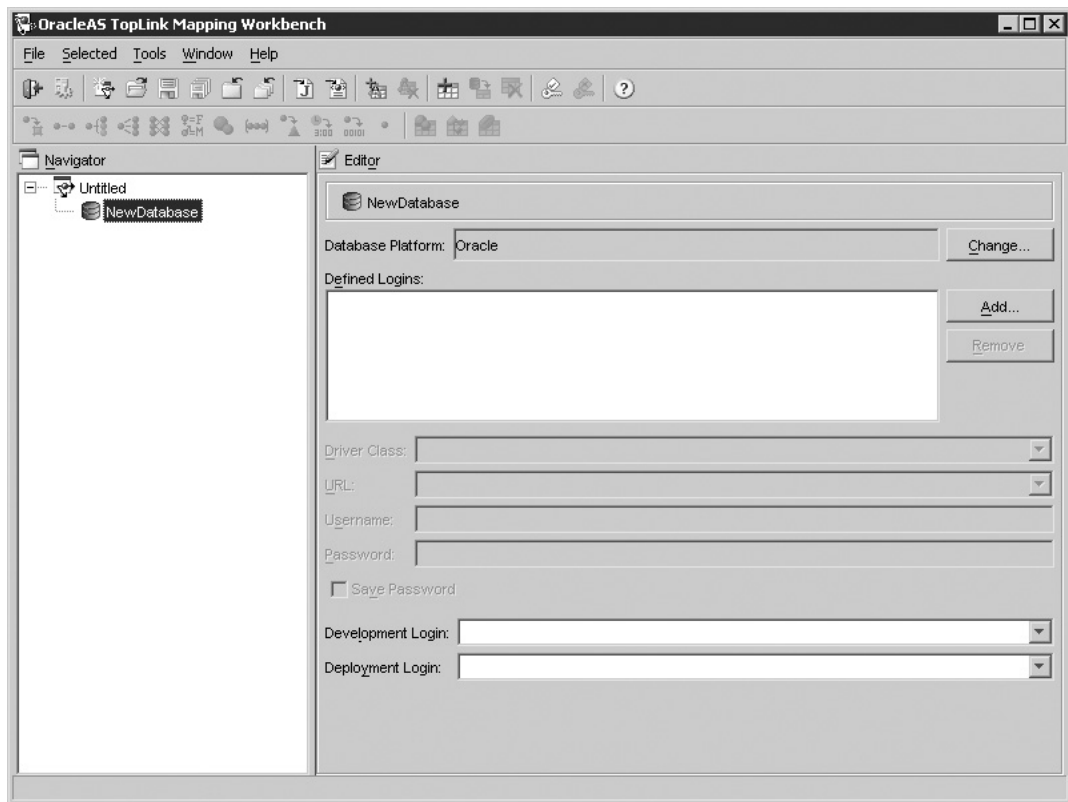


FIGURE 1-13. The TopLink Mapping Workbench

- **SQL Workshop** SQL Workshop enables you to interact with the database through a web browser. With it, you can view or create database objects, run SQL commands, and query by example.
- **Data Workshop** Data Workshop helps you import plain text and spreadsheet data into database tables and export data from database tables.

All HTML DB applications are displayed in your browser from data queried from the database, called *application definitions*. This metadata provides information about your application to the HTML DB engine, and therefore no code is generated (or needed). The Oracle HTML DB rendering engine reads the application definition and displays the application accordingly. Figure 1-14 shows an example of an HTML DB page.

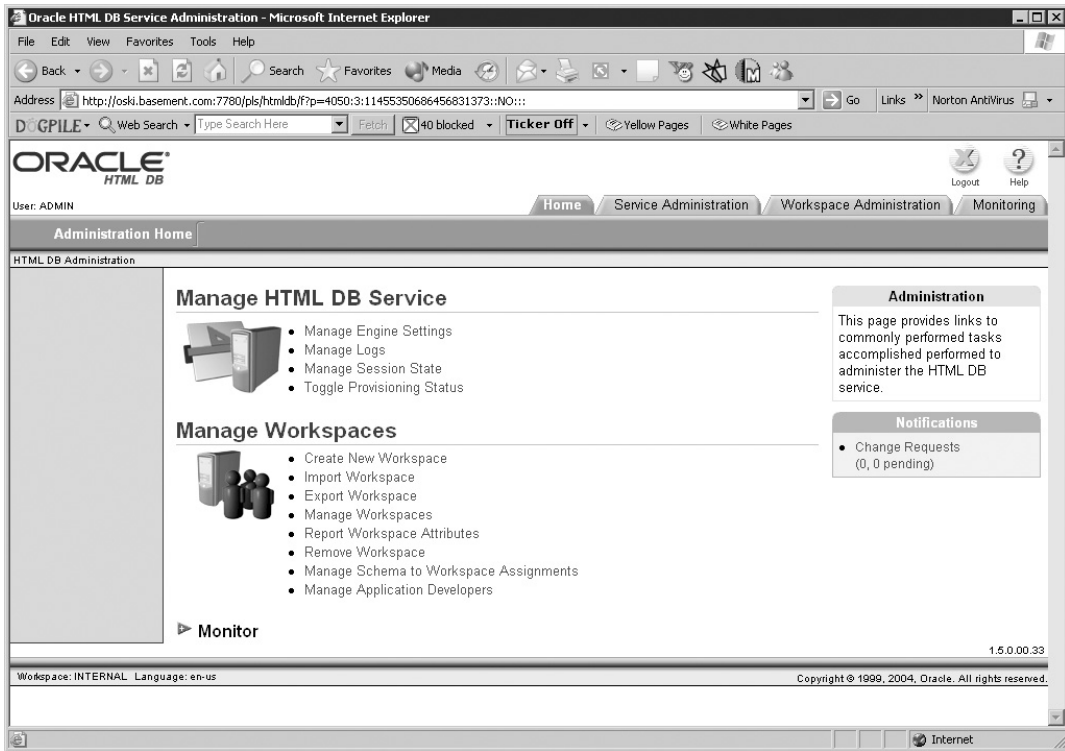


FIGURE 1-14. An HTML DB page

Database Tools

The PL/SQL Web Toolkit, a set of packages, procedures, and functions that directly interface with the `mod_plsql` module included with Apache is discussed in Chapter 7, but there are many enhancements to the PL/SQL language and Oracle's support of SQL in Application Server 10g that are mentioned here. Some of the major SQL enhancements include:

- Operators, functions, and constraints for new, native floating point datatypes
- The introduction of regular expression support
- Performance and size limit of large objects (LOBs) has been enhanced

Some of the major PL/SQL enhancements include:

- A rewritten, optimizing PL/SQL code generator for the PL/SQL Virtual Machine (PVM)
- Support for `binary_float` and `binary_double` datatypes
- Built-ins to support regular expression manipulation: `regexp_like`, `regexp_instr`, `regexp_substr`, and `regexp_replace`
- Multiset operations on nested table instances, supporting operations like equals, union, intersect, except, member
- Support for a user-defined quote character
- `INDICES OF` and `VALUES OF` syntax for `FORALL`

Summary

Web applications are applications based on open, public web standards that use a web browser as the client. They fall into one of two categories: HTML-based or applet-based. How you decide to code your application depends on numerous factors, as there is no one “right” way to code an application. Various technologies exist that excel at different aspects of web application development. In all but the smallest IT departments, you will be called upon to use a combination of these technologies.

Oracle offers all the software you need to build production-quality web applications. At the core of your Oracle-based web development is Oracle Application Server 10g, an Application Server designed to support all major web development languages, APIs, and frameworks, and to provide unparalleled flexibility, security, scalability, and reliability. Application Server 10g supports the full J2EE technology stack as well as Oracle-specific technologies such as Forms, Reports, and Discoverer. Oracle is committed to open standards, allowing developers to use the capabilities of Application Server 10g to solve virtually any development challenge.