

Oracle Designer/2000 WebServer Generator Technical Overview

VERSION 1.0: FEBRUARY, 1996

January 1996

Author: Mike Gwyer

Contributors: Rosie Harvey

Copyright © 1996 Oracle Corporation.

All rights reserved. Printed in the U.S.A.

This document is provided for informational purposes only and the information herein is subject to change without notice. Please report any errors herein to Oracle Corporation. Oracle Corporation does not provide any warrants covering and specifically disclaims any liability in connection with this document.

Table of Contents

TABLE OF CONTENTS	3
INTRODUCTION.....	4
ORACLE WEBSERVER OVERVIEW.....	5
THE DESIGNER/2000 ORACLE WEBSERVER GENERATOR	6
HOW THE ORACLE WEBSERVER GENERATOR WORKS.....	7
<i>Module Design</i>	7
<i>Generation Process</i>	7
MODULE DEFINITIONS AND DATA USAGES.....	8
GENERATED APPLICATION.....	9
GENERATED PAGES.....	10
STARTUP PAGE	11
QUERY FORM.....	12
RECORD LIST PAGE.....	13
VIEW FORM.....	14
USER PREFERENCES	15
<i>DBA</i>	15
<i>General Layout</i>	15
<i>Query Form Layout</i>	16
<i>Record List Layout</i>	16
<i>View Form Layout</i>	17
HTML FORMATTING	18
<i>Image References</i>	18
<i>Character Text Styles</i>	18
<i>Electronic Mail Addresses</i>	18
<i>Defining HTML Code Within Static Help Text</i>	18
<i>Defining HTML Code Against Column Usages</i>	19

Introduction

The Worldwide Web standards and technology were first established by computer scientists at the European Centre for Nuclear Physics (CERN) in the early 1990s to provide a hyper-text-based, multi-media information service on top of the Internet.

Today, the Worldwide Web is one of the most rapidly growing technologies in the world. It enables companies and individuals to publish information by creating Hyper-Text Markup Language (HTML) documents and publishing the Universal Resource Locator (URL) of these documents. These documents are then visible to anyone with an industry standard Web Browser such as Oracle PowerBrowser or Netscape, with document access controlled by a configuration file or the network firewall.

The technology developed at CERN was originally planned to run on top of the Internet providing wide-spread dissemination of information, the take-up of this technology has seen the greatest growth in purely internal Web Systems protected inside the firewall of a Company's network. As network performance and security improves, the use of the Web in Commercial Business Applications will grow significantly.

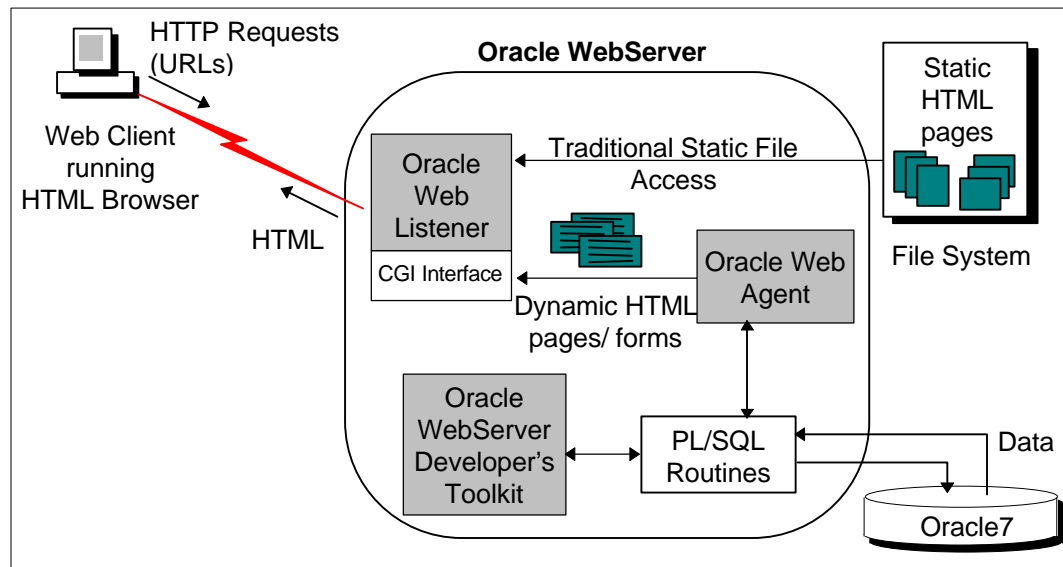
The sophistication of the Worldwide Web has developed rapidly. The advent of scripting languages such as Oracle PowerBasic and Hot Java, which can be run on the Web Server or as downline-loaded applets on the Web client, have elevated the Web from a simple, static information publishing mechanism, to an environment capable of supporting complex applications. A further attraction of this technology is that the architecture of a Web application removes the need for mass software distribution whenever the application changes - thus solving one of the problems faced by companies in the move to client-server applications.

The final obstacle to Web technology being regarded as a viable tool in client-server development was the lack of integration between Web application development tools and the Servers used by companies. The launch of Oracle WebServer removed this obstacle offering tight integration between Web applications and the Oracle7 Server.

The release of the Oracle Designer/2000 WebServer Generator (WSG) brings all the modeling and team-working benefits familiar to those currently generating applications into Developer/2000 or Visual Basic to those sites wishing to deploy applications on the Web. The WSG uses the same modeling techniques as the other Designer/2000 Generators., Thus it offers the ability to generate and deploy Web Applications from definitions already built and deployed in Developer/2000 or Visual Basic.

Oracle WebServer Overview

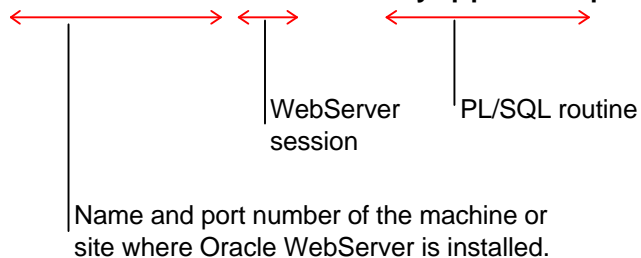
The Oracle WebServer is an HTTP server tightly integrated with the Oracle7 database. Business data stored on the Oracle7 database is formatted into Web (HTML) documents within the WebServer and then transmitted to Web Clients, on request.



Web users request HTML information by entering an appropriate Uniform Resource Locator (URL). This URL specifies the target Oracle WebServer machine or site, the WebServer session (Oracle database logon information), and the PL/SQL routine to run on that database user.

Example:

http://website:8888/fred/owa/myapp.startup



The PL/SQL routine specified within the URL fetches the data from the Oracle database and then makes use of other PL/SQL routines provided within the Oracle WebServer Developer's Toolkit to format the data within the required HTML syntax. The formatted data is then routed to the Web user originating the request, via the Web Agent.

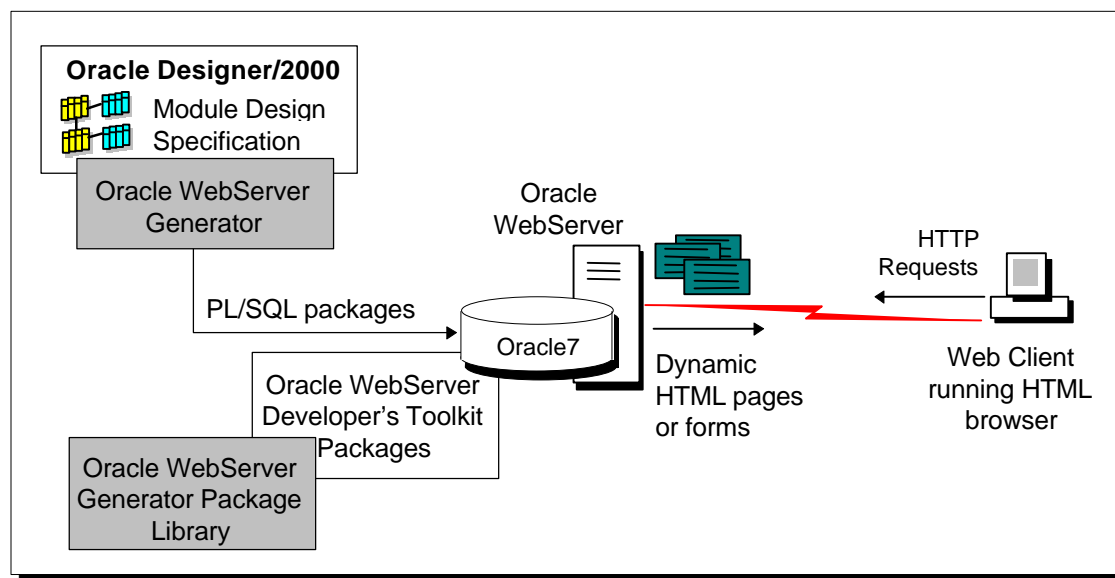
The Designer/2000 Oracle Webserver Generator

The Oracle WebServer Generator creates fully functional applications that publish data retrieved from an Oracle database over the World Wide Web. These generated applications are based on module and database design specifications recorded in the Designer/2000 Repository.

The main input to the generation process is a module design specification recorded through the Module Data Diagrammer component of Designer/2000. This records the tables and columns used by the module, the links between them, and detailed information on how the module uses the data. Standard display details for the tables and columns defined within the Application System (or Project) provide default display details for generated applications, thus ensuring a consistent look and feel across applications generated by any number of developers.

Modules can be linked together using the Module Structure Diagrammer. In the generated application these module links allow navigation between the modules via hypertext links with full passing of current context via parameters.

During generation the Oracle WebServer Generator creates a set of PL/SQL packages, which are then installed into an Oracle WebServer database. Preferences determine the general look and feel of the generated application and can be customized to suit particular requirements.

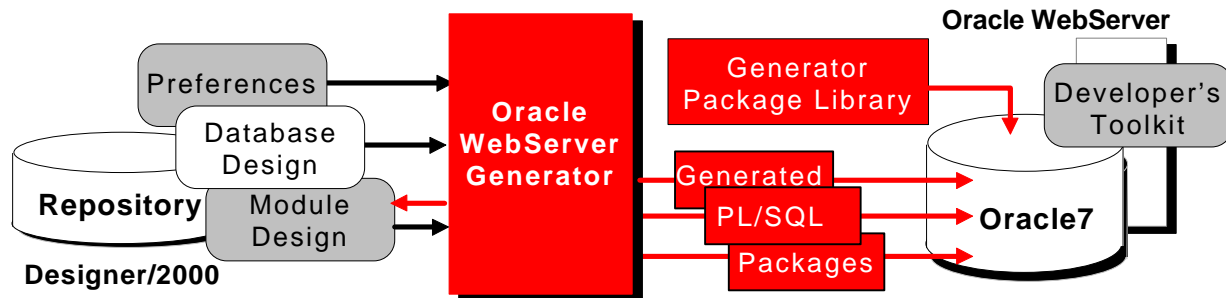


Oracle PowerBrowser, or any other HTML Browser may be used to run generated applications by calling the Oracle Web Agent which dynamically builds HTML documents from the PL/SQL packages generated by the WSG.

How the Oracle Webserver Generator works

Module Design

To generate an Oracle WebServer application, you first specify the set of module definitions that make up the application and then define any links between them. The design of these modules provide the main input to the generation process.



Generation Process

When you generate your application you choose which module you wish to generate and also whether or not to generate all the other modules called (directly or indirectly) by the chosen module.

Oracle WebServer Generator examines the design specification of each module, detects any links between modules and determines the values of user preferences.

During generation, the Generator creates a pair of PL/SQL files for each module:

- package specification file .PKS
- package body file .PKB

A PL/SQL package is created for each module component defined within a module, plus one for the module itself. The Oracle WSG builds PL/SQL procedures inside these packages to provide the database querying and parameter management functionality for the pages required.

The generation process also results in the creation of a single .SQL file which simply calls all the other generated files (.PKS and .PKB). After generation you are given the option to run this file to install the generated Oracle WebServer application onto to an Oracle database. After installation you may run the application using your chosen HTML Browser.

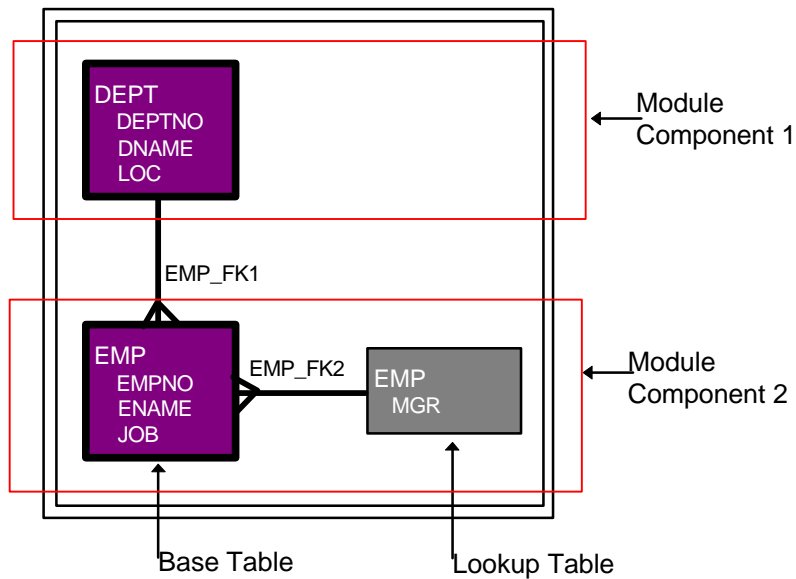
In addition to generated packages Oracle WebServer Generator makes use of its own PL/SQL package library to provide generic functionality within the application.

Module Definitions and Data Usages

A generated Oracle WebServer application comprises one or more dynamic HTML Web pages. Each generated page contains one or more sections, displaying static HTML text and formatted data sourced from SQL queries. Module components form the basis of each SQL query and they are defined visually using the Module Data Diagrammer.

A single module definition may contain one or more module components, linked together via foreign keys. Module components are based on a specific table (the base table) but may also make use of other tables to provide lookup information (lookup tables).

The module definition illustrated below has two module components, one based on a Departments table and the other on an Employee table.

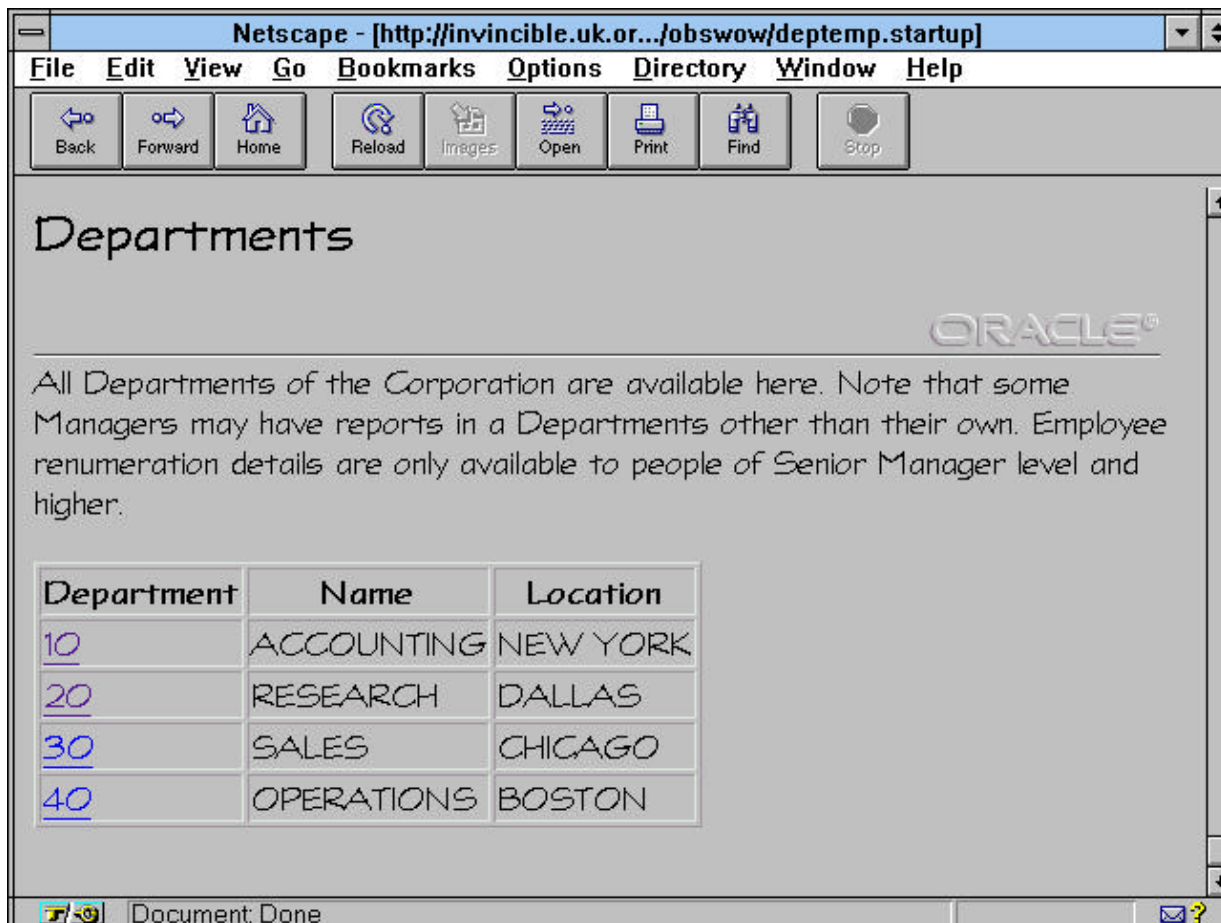


For each module component you must define which columns are to be incorporated within the application and their display characteristics e.g. caption, any special HTML formatting etc.

Generated Application

Generating the example module shown on the previous page produces an application listing the departments in a company and, when a department has been selected, listing the employees that work in that department showing the employee number, name and job title as well the name of their manager.

The following figure shows the startup page for this module containing the title, a boilerplate image, the module help text and an HTML table showing the list of departments in the Company. The figure is a screndump of the generated application viewed through an Industry standard Browser.



Generated Pages

From the module design specification, Oracle WebServer Generator generates a set of PL/SQL packages which at runtime dynamically build a series of linked Web pages. The Web pages created by the Oracle WSG break down into four types.

- Startup Page
- Query Form
- Record List
- View Form

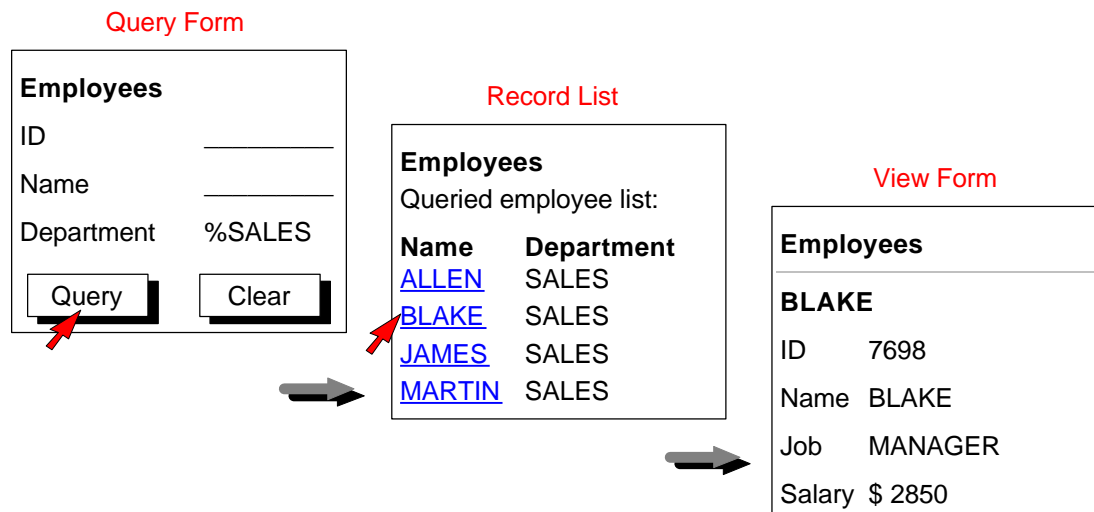
A Startup Page is created for every module within the generated application.

A Query Form allows the Web user to enter search criteria before executing the query.

A Record List displays a set of queried records in an HTML table, or in an ordered or bulleted list.

A View Form displays the full details for a chosen record.

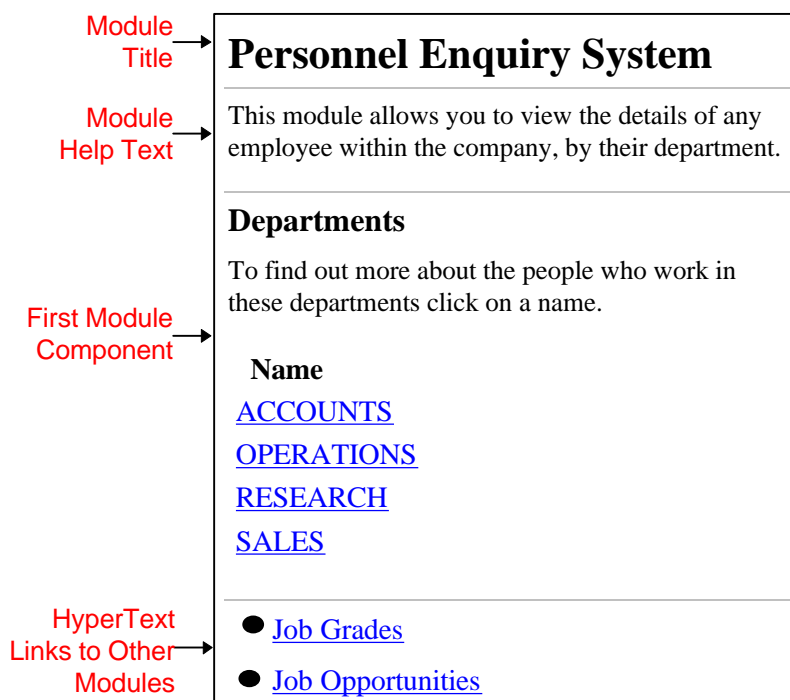
The diagram below shows the relationship between the Query Form, Record List and View Form for an example module.



The appearance and sequence of pages depends on the module design specification and user preference values. For example, a Record List for a detail module component may be placed on the same page as its master's View Form, rather than on separate pages.

Startup Page

A Startup Page is generated for every module within the Oracle WebServer application. If the module is the top level module for your Oracle WebServer application, this page will typically serve to introduce the entire application and provide a set of hypertext links to the rest of the modules within the application. For other modules, the Startup Page serves to introduce the module and may, if appropriate, display the query generated from the first module component.



Example illustrating the structure of a generated Startup page

The Help text may be added to describe the purpose of the module or give instructions. Such text is derived from the User Help Text recorded against the module definition.

Module links allow you to navigate between the modules which make up the application and they are defined using the Module Structure Diagrammer. Calls to other modules may be organized into a hierarchical structure by the inclusion of MENU type modules in the overall module hierarchy. Within generated Oracle WebServer applications, calls to other modules appear as hypertext links.

Query Form

The Query Form allows a Web user to enter search criteria to be incorporated when executing a query against a particular module component. The Query Form displays the set of column usages marked 'Select' in the module design.

In the example below four column usages had been marked as Select in the Module Definition; Name, ID, Job, and Salary.

The image shows a web form titled "Employees" with the subtitle "Enter Query criteria for Employees". It contains four input fields: "Name:" (a long text box), "ID:" (a short text box), "Job:" (a long text box), and "Salary:" (two text boxes separated by "to"). At the bottom are "Query" and "Clear" buttons. Red annotations include: "Query Title" pointing to the title, "Captions for Columns marked Select" pointing to the labels, and "Criteria Fields" pointing to the input boxes.

Example illustrating the structure of a generated Query Form

The buttons are:

- | | |
|-------|------------------------------------------------------------------------------------------------------------------------------------|
| Query | Submits the query based on the given search criteria and then dynamically builds and displays the page containing the Record List. |
| Clear | Clears all the column values, ready for the Web user to enter new query conditions. |

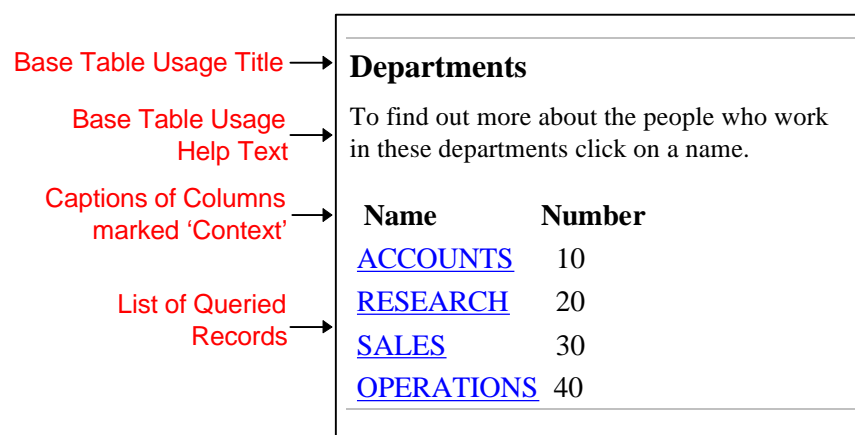
For column usages based on non-key DATE or NUMERIC columns, a pair of From/To column values is created to allow for range searching as illustrated in the example above.

A Query Form is only generated if one or more of the column usages are marked 'Select' in the module design. If no such column usages are defined, then no Query Form is created and an unrestricted query is performed instead. This may be appropriate for a detail query that is automatically restricted by the chosen master record.

Record List Page

The Record List Page displays a set of records for a particular module component (i.e. base table usage and its associated lookups). The list normally displays a subset of the table's columns, sufficient for the user to identify the records. A separate View Form is generated to display the full details of any record selected from the list.

The Record List displays the set of column usages marked 'Context' in the module design. In the example below two columns are marked Context; Name and Number.



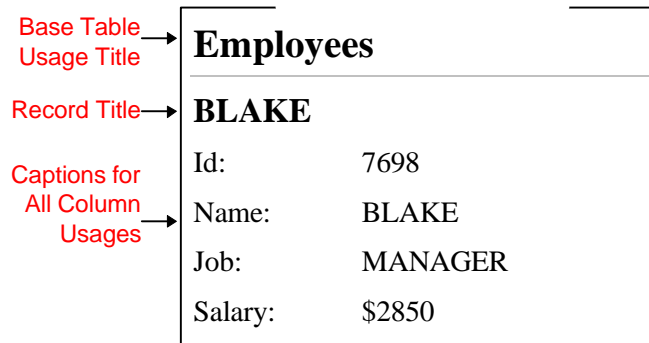
Example illustrating the structure of a generated Record List

The Help text may be added to describe the Record List or give instructions. Such text is derived from the User Help Text recorded against the module component's base table usage.

View Form

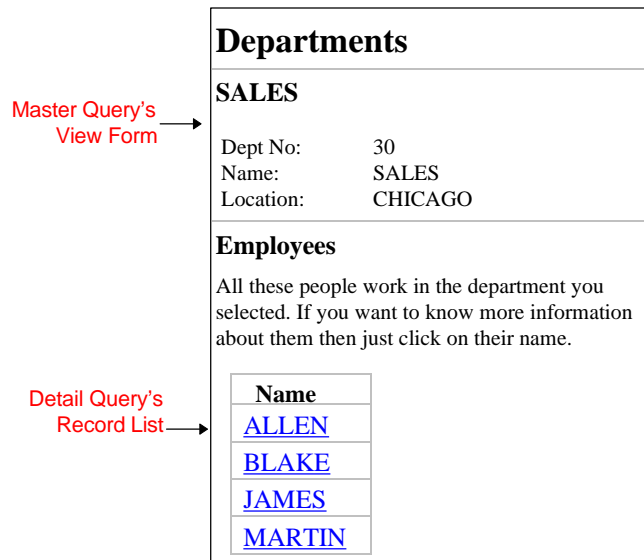
The View Form displays the details of any one record. It contains the full set of column usages defined against a particular module component (i.e. base table usage and its associated lookups).

When a Web user chooses a hypertext entry in the Record List the View Form is displayed on a new Web page.



Example illustrating the structure of a generated View Form

If the module design contains module components acting as details, the Record List (or Query Form) for the detail may be displayed on the same page as the master's View Form, as shown in the following example.



Example illustrating the structure of a generated View Form with detail records in a Record List on the same page

The placement of master-detail components on the same, or separate Web pages is defined visually, using the Module Data Diagrammer.

User Preferences

Following installation, the Oracle WebServer Generator can be used 'as is' to generate applications with a consistent look and feel. However, the appearance and behavior of a generated application may be configured, or customized by user preferences to suit your particular requirements.

Preferences for Oracle WebServer Generator can be set at an application-wide level to be inherited by all modules in that application, or on a module-by-module basis. Certain preferences can also be set on individual module components within a module. Preference settings can be grouped into 'named' preference sets and these sets can then be applied to different modules or module components.

Oracle WebServer preferences are grouped into these categories:

- DBA
- General Layout
- Query Form Layout
- Record List Layout
- View Form Layout

DBA

These are preferences for referencing database objects. Such preferences include

- Name of account against which generated Application should be run
- Scope of reference code tables

General Layout

These preferences are options that govern the user interface; for example

- Name of the background image which should be used as a background for all pages in the module.
- Name of graphic which should be used as a Custom List bullet in a Record List Page.
- Whether the first zone should be included on the Module Startup Page.
- Item prompt separator
- Split prompt separator
- Default Character display length
- Default Data display length
- Default Number display length

Query Form Layout

These preferences determine the appearance of the Query Form. They include

- Zone alignment
- HTML table border
- Horizontal alignment of fields
- Layout style, one of:
 - TABLE (if Browser supports them)
 - PREFORMAT (like a TABLE but displayed in fixed font, visible on all browses)
 - BULLET list
 - NUMBER list
 - CUSTOM (Build a list which begins with the HTML defined in the MODCLH preference)
 - WRAP (No formatting)
- Horizontal alignment of prompts
- Vertical alignment of rows

Record List Layout

Options that determine the appearance of the Record List. Such preferences include

- Zone alignment
- HTML table border
- Horizontal alignment of Character columns
- Horizontal alignment of Date columns
- Horizontal alignment of Number columns
- Layout style, one of:
 - TABLE (if Browser supports them)
 - PREFORMAT (like a TABLE but displayed in fixed font, visible on all browses)
 - BULLET list
 - NUMBER list
 - CUSTOM (Build a list which begins with the HTML defined in the MODCLH preference)
 - WRAP (No formatting)
- Vertical alignment of rows

View Form Layout

These preferences determine the appearance of the View Form. Such preferences include

- Zone alignment
- HTML table border
- Horizontal alignment of Character columns
- Horizontal alignment of Date columns
- Horizontal alignment of Number columns
- Display of null values
- Layout style, one of:
 - TABLE (if Browser supports them)
 - PREFORMAT (like a TABLE but displayed in fixed font, visible on all browses)
 - BULLET list
 - NUMBER list
 - CUSTOM (Build a list which begins with the HTML defined in the MODCLH preference)
 - WRAP (No formatting)
- Alignment of horizontal prompts
- Vertical alignment of rows
- Location of images

HTML Formatting

Additional HTML formatting can be incorporated within the module design specification to enhance the appearance and add functionality to the generated application. HTML code can be recorded directly against column usages or embedded within help text. Some examples of HTML formatting:

Image References

Through the use of image references embedded within the module design, image files stored on the Oracle WebServer file system may be used in various ways to enhance the appearance, or clarify the purpose of a generated application. Such images can provide a background to Web pages, be embedded within static text or referenced dynamically based on a column values. The most commonly supported image formats are GIF and JPEG.

Character Text Styles

A range of standard HTML functions may be embedded within help text to highlight aspects of the text you define, for example **bold** or *italic*. The same functions may also be applied to column usages.

Electronic Mail Addresses

Electronic mail addresses can be embedded within generated applications

Defining HTML Code Within Static Help Text

Static help text may be recorded against module definitions and individual module components. Within this static text you may reference any of the HTML functions provided by the HyperText Functions (HTF) package within the Oracle WebServer Developer's Toolkit.

For example the following static help text in the Designer/2000:

```
Welcome to the htf.bold('ACME Corporation') home page. Send any
comments on this page to htf.MailTo('JBLOGGS@ACME.COM', 'Joe
Bloggs').
```

Would be generated into HTML

Welcome to the ACME Corporation home page. Send any comments on this page to [Joe Bloggs](#).

Defining HTML Code Against Column Usages

Oracle WebServer Generator supports a set of standard built-in formatting options that may be set against the column usage, such as BOLD, ITALIC, IMAGE and MAILTO. For more flexible formatting, calls to PL/SQL Toolkit functions may also be defined.

These formatting requirements are defined against the HTML Formatting property of the column usage.