

Oracle Applications DBA Field Guide



Elke Phelps
Paul Jackson

Apress®

Oracle Applications DBA Field Guide

Copyright © 2006 by Elke Phelps and Paul Jackson

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-59059-644-9

ISBN-10 (pbk): 1-59059-644-7

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Jonathan Hassell

Technical Reviewers: Srinu Chavali, Sandra Vucinic

Editorial Board: Steve Anglin, Dan Appleman, Ewan Buckingham, Gary Cornell,
Jason Gilmore, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie,
Dominic Shakeshaft, Jim Sumser, Matt Wade

Project Manager: Elizabeth Seymour

Copy Edit Manager: Nicole LeClerc

Copy Editor: Andy Carroll

Assistant Production Director: Kari Brooks-Copony

Production Editor: Laura Cheu

Compositor: Dina Quan

Proofreader: Linda Seifert

Indexer: Carol Burbo

Artist: Kinetic Publishing Services, LLC

Photographer: Chris Fieldhouse

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.



Components and Architecture of Oracle Applications

The Oracle E-Business Suite is a complex Enterprise Resource Planning program that consists of many components. The primary components that comprise the E-Business Suite are the Client, Forms Server, Web Server, Concurrent Processor, and Oracle Database. Each of these plays a role in servicing Oracle Applications.

This chapter will provide an overview of the function that each of these components provides to enable end-users to access and use the application. We'll look at two main topics in this chapter:

- **Oracle Applications:** An overview of how users access the applications and the different components that service their requests. In the process, we'll look briefly at the Client, Web Node, Forms Node, Concurrent Processing Node, and Database Node.
- **Oracle Applications architecture:** A look at the architecture of the E-Business Suite from basic to complex configurations. This includes load balancing nodes, a shared APPL_TOP or Application Tier Filesystem, distributed APPL_TOP, and Secure Socket Layer (SSL) Encryption.

In addition to the primary components identified so far, there are many other components of the E-Business Suite architecture, including networking infrastructure, servers, routers, and load balancing devices, to mention only a few.

Note While we will provide an overview of complex, advanced architecture infrastructures, the details of how to implement one are beyond the scope of this guide. Oracle Discoverer, Oracle Portal, Oracle Internet Directory, Oracle Single Sign-on, Oracle Web Cache, and Oracle Integration are also out of the scope of this guide.

Servicing User Requests—Oracle Applications Components

In order to understand the primary components of the Oracle E-Business Suite, it is important to know how the user accesses the application. As the first step to accessing Oracle Applications, a user will launch a web browser and enter the URL that is the *web entry point* for the application. The Web Server then services the access request.

The first page that is displayed by the Web Server is a login screen. Once logged in, the user picks a responsibility, such as System Administrator, and then a menu option, such as Security : User ► Define, to begin his or her work. The menu option will direct the user to an HTML or JavaServer Pages (JSP) page, or to a Forms application. The Web Server will continue to service HTML or Java servlet requests; however, if a Forms application is launched, a Forms servlet or the Forms Server will service it. Throughout this process, the user is retrieving data and executing packages from within the Oracle Database.

Now that you have a very high-level overview of how users access the application, we can look at some specifics of the components that service requests. The following components will be described:

- **Client:** The requirements and processes on the user workstation
- **Web Node:** Web Server processes that run on the Web Node
- **Forms Node:** Forms Server processes that run on the Forms Node
- **Concurrent Processing Node:** Concurrent Manager processes running on the Concurrent Processing Node
- **Admin Node:** Administrative tasks executed on the Admin Node
- **Database Node:** Database services that run on the Database Node

■ **Note** A node comprises processing power (CPU) and memory. Multiple nodes servicing different functions may be hosted on the same server. For example, a Web Node, Forms Node, Concurrent Processing Node, and Admin Node may all run on the same Oracle Application Server.

Client

Users accessing Oracle Applications are required to have an Oracle-certified web browser, such as Microsoft Internet Explorer or Netscape. Oracle Applications are served as either web applications or Oracle Forms. A user's first interaction with the application is a login screen that is presented in the web browser, and from there the user can either continue to access web pages or can access Forms applications. The Oracle JInitiator plug-in is required to run Oracle Forms as Java applets on the Client.

Web Node

The user initially accesses the application via a web browser with a URL for the web entry point. The Web Server services this web page request. For Oracle applications, the Web Server is the Oracle Application Server, which is based on Apache technology, and the Web Node is the node that runs this server. The Oracle Application Server is also called iAS, AS, Oracle HTTP Server (OHS), or simply Apache.

The iAS listens for incoming requests on a specific port. The iAS also runs the JServs that are used to service Java requests. For Oracle Applications, the iAS may also be configured to run Forms servlets. If this is the case, then the iAS will also service Forms sessions.

Forms Node

If Forms servlets are not configured for the iAS, then Forms sessions are serviced by the Forms Server. When a Forms request is initiated, the iAS hands off the Forms request to the Forms Server. Much like the iAS, the Forms Server listens for incoming requests on a specific port. The Forms Node is the node that runs the Forms Server.

Concurrent Processing Node

Concurrent processing is a special feature of Oracle Applications. It allows the user to schedule jobs, which Oracle calls *requests*. These requests may be standard Oracle requests or custom requests, they can be scheduled as one-time requests or on a repeating schedule, and they can be submitted to execute immediately or at a specific time.

Requests are scheduled with the scheduling manager, which is called the *Concurrent Manager*. The node that runs the Concurrent Manager processes is called the Concurrent Processing Node.

Admin Node

There are many administrative tasks that are executed in order to maintain the Oracle E-Business Suite, such as regenerating forms, regenerating jar files, applying application patches, and recompiling flexfields. The Admin Node is used to execute administrative tasks.

Database Node

The heart and soul of the E-Business Suite is the database. The database not only stores the data in tables under various schemas, but also stores many other objects (such as procedures, packages, database triggers, functions, indexes, and sequences) that are required for the application to function. The Database Node is where the Oracle Database instance runs and accesses the database files.

Oracle Applications Architecture

Some implementations of Oracle Applications are set up with a basic configuration. Others require advanced configuration for specific features. We will start with an overview of the basic architecture requirements and then move into advanced configuration options.

Fundamental Architecture

When a system is deployed with a basic approach to architecture, it typically does not have large transactional processing requirements or a large concurrent user base. For this environment, there are no special configuration requirements. These implementations may run on one tier, meaning that all nodes are running on one physical server, but this is a very inefficient method.

Some implementations run the application components on one server, while the database node runs on a separate server. This is a *two-tier architecture*. Multi-tier environments do not require special configuration or design effort unless multiple nodes for the same component are required (this will be described in greater detail in the following “Advanced Architecture” section). A simple, two-tier Oracle Applications environment is displayed in Figure 1-1.

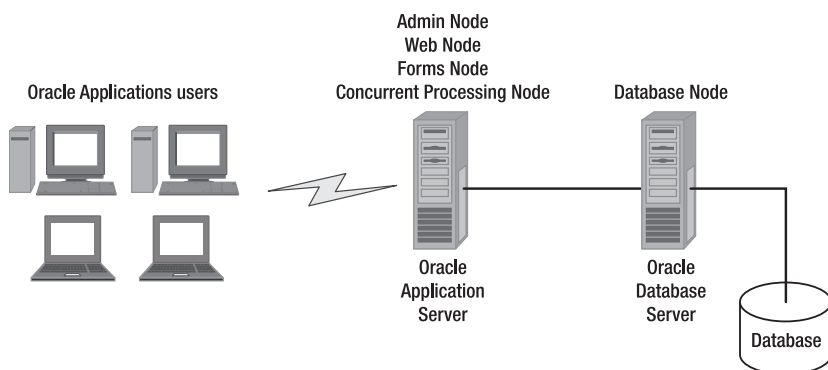


Figure 1-1. *Oracle applications: two-tier architecture*

Traditionally, Oracle recommended that the Concurrent Processing Node run the same tier as the Database Node. However, with fast network connectivity between the Concurrent Processing Node and the Database Node, it is now recommended that the Concurrent Processing Node run on the application tier.

■ **Tip** A two-tier architecture is the minimum number of tiers recommended to run Oracle Applications. Single-tier architectures can cause contention between application and database processing, which will result in performance degradation.

Advanced Architecture

When more nodes of the application tier are split across multiple servers, and additional nodes are defined for the same component, we begin to enter into advanced configuration topics and design.

Advanced, multi-tier configurations for Oracle Applications include combining multiple Web, Forms, Concurrent Processing, and Database Nodes. The number of nodes required is dependent upon your environmental requirements for concurrent user support and transactional processing. An advanced multi-tiered Oracle Applications environment is displayed in Figure 1-2:

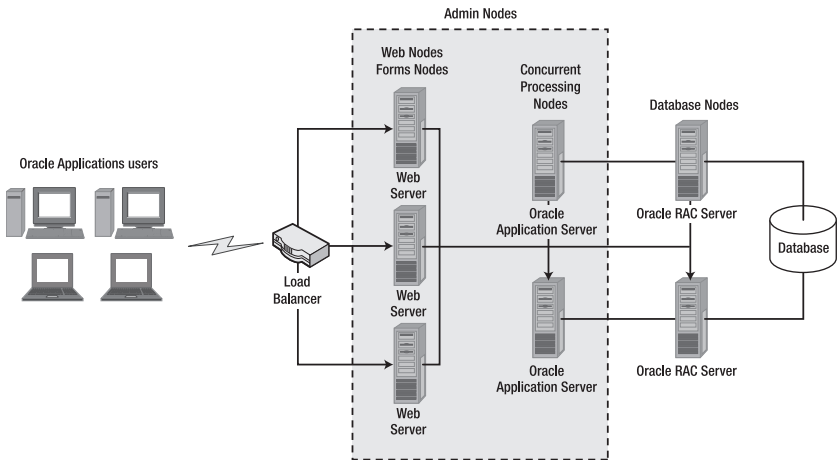


Figure 1-2. *Oracle Applications: an advanced multi-tier architecture*

This section will not provide the details required to implement a complex architecture, but it will give you the background to begin research into which advanced configuration topologies might be required by the organization you service. The following topics will be covered in this section:

- **Load balancing:** The requirements for load balancing the various nodes of the E-Business Suite
- **Shared APPL_TOP or Application Tier Filesystem:** The support of a shared applications layer, and when it should be used
- **Distributed APPL_TOP:** The support of a distributed application layer, and when it is beneficial
- **Secure Sockets Layer (SSL) Encryption:** An overview of SSL and its implementation requirements for Oracle Applications

Load Balancing

When a large number of users need to access your environment, or when the number of transactions to be processed is great, it may be necessary to create multiple nodes that service the same function. For example, if your business or customer requires the ability to support 5,000 concurrent Forms users, servicing these requests with either one Web Node or one Forms Server may cause contention in the system. This would result in users being unable to access the application. In order to resolve this problem, multiple Web or Forms Nodes would need to be put into operation.

Load balancing is the term used to describe how users or transactions are distributed to multiple nodes that service the same function. When more than one node is used, the nodes that service the same function are called a *farm*. For example, if you determine that your environment requires multiple Web Nodes, the multiple Web Nodes are collectively referred to as a *Web farm*. The Web Nodes may be further load balanced by implementing multiple JServs per Web Server. If your environment requires a large amount of Java processing, configuring additional JServs will reduce contention for its resources.

Web Node load balancing may be achieved by employing a hardware load balancing device or with DNS load balancing. Forms load balancing is implemented with either the Web Node as the load balancer for Forms servlets or as multiple Forms Nodes. If multiple Forms Nodes are implemented, one of the Web Nodes is designated as the primary Web Node and serves as the entry point for access to the Forms Nodes. The Forms Metrics Server runs on the primary Web Node and serves as the load balancer for sending requests to the multiple Forms Nodes. Information regarding advanced configuration for the Oracle E-Business Suite can be found in MetaLink Note 217368.1.

When the Concurrent Processing Nodes are load balanced, this configuration is referred to as *Parallel Concurrent Processing*. Parallel Concurrent Processing is load balanced by the Internal Concurrent Manager. If Parallel Concurrent Processing is required, then a shared filesystem implemented with either Network Filesystem (NFS) or a shared disk array is required to share log and output files that are generated by the Concurrent Managers. Additional information regarding Parallel Concurrent Processing may be found in MetaLink Note 185489.1.

For the database, a multiple-node implementation may be achieved by implementing Oracle Real Application Clusters (RAC). In a RAC environment, multiple Database Nodes function as one database instance, accessing the same database files. Additional information for implementing Oracle RAC with 11i may be found in MetaLink Note 312731.1.

Note Oracle has not certified and does not support load balancing of the same types of nodes on the same physical server.

Shared APPL_TOP or Application Tier Filesystem

Each implementation of Oracle Applications contains an APPL_TOP and a COMMON_TOP directory on each node. The APPL_TOP directory comprises all product files and directories, all core technology files and directories, as well

as the application context file and environment files. Details regarding the context file and environment files are provided in Chapter 2 of this guide. The `COMMON_TOP` directory contains files and directories that are used by all application products.

While not necessary, it is recommended that you investigate implementing a shared `APPL_TOP` or Application Tier Filesystem for a multiple-node installation. In a shared `APPL_TOP` implementation, a shared filesystem (either NFS or a disk array) is used to store the `APPL_TOP` and `COMMON_TOP` structures. Because the `APPL_TOP` and `COMMON_TOP` directories contain application code and binaries, placing them on a shareable filesystem will reduce maintenance downtime, since only one copy of the `APPL_TOP` and `COMMON_TOP` sources exist.

As of version 11.5.10 of Oracle Applications, a shared Application Tier Filesystem may be implemented. A shared Application Tier Filesystem not only includes the `APPL_TOP` and `COMMON_TOP` directories, but also the Applications Technology Stack components of the iAS and Developer Tools (Forms, Reports) installation. This provides even greater manageability of the application environment.

Just imagine having an implementation on ten nodes without a shared `APPL_TOP` or Application Tier Filesystem. You would need to maintain the application and Applications Technology Stack code for all ten nodes! This exemplifies the benefit of a Shared `APPL_TOP` or Application Tier Filesystem. MetaLink Note 233428.1 provides details for implementing a shared `APPL_TOP` or Application Tier Filesystem.

Distributed `APPL_TOP`

A distributed `APPL_TOP` is yet another advanced configuration feature of Oracle Applications. With this configuration, you can use some or all of the servers in your implementation to serve as Admin Nodes. An administrative task will distribute workers on multiple servers that are configured as Admin Nodes.

This feature may assist in reducing downtime by expediting administrative functions, such as when a patching session spawns multiple workers across multiple nodes. Details for implementing a distributed `APPL_TOP` are outlined in MetaLink Note 236469.1.

Secure Sockets Layer Encryption

Secure Sockets Layer (SSL) is a method of encrypting transactions and data over a network. Securing transactional data is often a requirement when said transactions contain sensitive data or information, such as credit card data.

If encryption is required, it may be implemented with Oracle Applications. SSL may be implemented for the Oracle HTTP Server, Forms Server, and Database Server. SSL may be implemented with software or with a hardware device known as an SSL accelerator. Details for implementing SSL are given in MetaLink Note 123718.1.

Architecture Best Practices

When designing the infrastructure of your Oracle E-Business Suite implementation, it is important to understand your service level agreement with the customer, as well as the concurrent user requirements of the application. This will help you determine the level of scalability and availability that you will need to provide. Additional scalability and availability may be achieved by implementing multiple nodes that service the same function.

If you are considering implementing multiple nodes for load balancing, it is recommended that you consider implementing the additional nodes on commodity servers. Commodity servers are cheaper servers generally based on the Intel architecture running Linux. Implementing commodity servers will allow you to transition to a load balanced, multi-tier configuration with a lower total cost of ownership.

While details regarding implementing Oracle Web Cache were not discussed in this chapter, it is worth investigating this technology as part of your E-Business architecture solution. Overall performance may be significantly improved if Oracle Web Cache is implemented with your environment. Additional details regarding implementing Web Cache may be found in MetaLink Note 306653.1.

Infrastructure upgrade requirements, including client workstation, server, networking, and hardware firmware upgrades, to mention a few, should be implemented with caution. A “minor” upgrade to one of these components may cause outages for your Oracle Applications environment. Be certain to sufficiently test all such upgrades or modifications to the supporting Oracle E-Business Suite infrastructure, and have a plan to roll back changes if necessary.

