



Official Publication of the Northern California Oracle Users Group

NoCOUG

J O U R N A L

Vol. 26, No. 2 · MAY 2012

\$15

Knowledge Happens

Show Me the Way

An interview with the genius behind Statspack and AWR.

See page 4.

Big Data Appliance

Learn about the game-changer from ACE Director Gwen Shapira.

See page 12.

RAC and Ruin

An excerpt from Jonathan Lewis' latest book.

See page 18.

Much more inside . . .

Oracle WebLogic Server 11g Administration Handbook

A Book Review by Brian Hitchcock

Details

Authors: Sam Alapati
ISBN: 978-0-07-177425-3
Pages: 528
Year of Publication: 2011
Edition: 1
List Price: \$60
Publisher: Oracle Press



Overall Review: Excellent resource for anyone new to WebLogic Server.

Target Audience: Anyone that will be managing WebLogic Server.

Would you recommend this book to others: Yes.

Who will get the most from this book? Administrators.

Is this book platform specific: No.

Why did I obtain this book? See overall review below.

Overall Review

I need to be prepared to support Fusion Applications. I found many resources, and among them were three books from Oracle Press. This is the second of them. The first was *Managing Oracle Fusion Applications* and the third is *Handbook and Oracle Fusion Middleware 11g Architecture and Management*. Since I have reviewed other Oracle Press books, they sent me copies of each to read and review.

Fusion Applications is built on top of Fusion Middleware, and WebLogic Server is a central piece of Fusion Middleware. WebLogic Server (WLS) provides the “container” where Java EE applications are deployed. The business functionality of Fusion Applications is provided by Java EE applications, which explains why WebLogic Server is critical to supporting Fusion Applications.

Even if you won't be supporting Fusion Applications, WebLogic Server is replacing Oracle Application Server in various Oracle products. As an example, the latest version of Oracle Enterprise Business Suite (EBS), version 12.2, will have WebLogic Server.

I've been working with WebLogic Server and Fusion Applications for about a year, and I wanted to read this book because I had several specific issues I wanted to learn about. I saw many WebLogic Server instances go into the state FAILED_NOT_RESTARTABLE, and I needed to understand what causes this. While this book never mentions this issue by name, I did learn

how the WebLogic Server instance goes through various states while starting up. When this process encounters issues, this is when the instance fails.

This book is very dense with specific technical detail. I learned many things about managing WebLogic Server. The focus is on how things work so that you can implement and manage WebLogic Server in your environment. Many issues that had been confusing to me about WebLogic Server were cleared up by reading this book.

Introduction

The introduction explains that WebLogic Server is mainly used to deploy web applications. It provides the environment specified by the Java Enterprise Edition (Java EE) standards to support enterprise Java applications. Further, WebLogic Server is itself a Java program that supports various services needed by Java EE applications. I agree with the author's view that WebLogic Server presents an overwhelming number of new concepts that a beginner must deal with. I've been there very recently myself. The target audience is explained to be WebLogic Server administrators, middleware administrators, and DBAs that need to support WebLogic Server. This is important in that many DBAs will get tasked with WebLogic Server support, if only because it is another Oracle product. The difference between web server and web application server (such as WebLogic Server) is explained clearly. I was reassured to read that no prior knowledge of WebLogic Server or Java programming is assumed or needed.

Chapter 1—Installing WebLogic Server and Using the Management Tools

The three major administrative tools for managing WebLogic Server are identified here: the Administration Console, the Node Manager utility, and the WebLogic Scripting Tool (WLST). By the way, it turns out that you must like acronyms to work with WebLogic Server. The available versions of WebLogic Server are explained (SE, EE, and Suite). The author points out that there is a lot of new terminology to learn: for example, the concept of a “machine” in the WebLogic Server world, which can be confusing. This caught my attention because I was confused about this term myself. The list of terms that is explained, each in detail, is lengthy and welcome. I learned a lot from this. Following this is a section on WebLogic Server concepts. Again, I learned a lot. Specific examples include Configuration and Runtime MBeans, Listen Ports and Listen Threads, and why there are two versions of the Java

Virtual Machine (JVM) included with WebLogic Server.

The chapter then covers the WebLogic Server installation process. The author covers this in detail, and I installed WebLogic Server and the sample applications on my laptop. I highly recommend this. Using the Administration Console, logging in, and navigating all make a lot more sense when you are using your own installation. WebLogic Scripting Tool is also covered and many examples are explained.

Chapter 2—Administering WebLogic Server Instances

Managing WebLogic Server instances is the focus of this chapter. The Admin Server and Managed Servers are explained as well as the difference between development and production modes. The many options available for starting and stopping WebLogic Server instances are covered. Node Manager capabilities, as well as how to start, stop, monitor and configure Node Manager, are covered. When WebLogic Server starts up it goes through various “states,” ending up in a RUNNING state ready to accept and service client requests. This process, referred to as the “life cycle,” was something I had not seen before. It helped me understand what causes WebLogic Server instances to fail.

This chapter also provides an introduction to the Ant tool. Ant is a Java-based tool that uses XML build files to start and stop WebLogic Server instances as well as many other tasks. Again, this helped me because I had seen the message “BUILD SUCCESSFUL” many times but didn’t know this was generated from Ant.

Chapter 3—Creating and Configuring WebLogic Server Domains

The concept of a domain is central to understanding WebLogic Server. The structure of a domain is covered, including server instances, server clusters, and the directory structure that is set up when you create a domain. The XML configuration file for the domain is explained. MBeans come up again as these are the mechanism Java uses to monitor and change the configuration of servers within the domain.

When you make changes to the configuration of any of the servers in a domain, you do so through the Administration Console. This process makes changes to the editable MBeans, and the Admin Server uses a Lock and Edit Mechanism to make sure only one person is making configuration changes at a time. This process is covered as well as creating domain templates. These templates are useful when you need to create many WebLogic Server domains.

The steps needed to create a domain are presented, which extends the installation process begun in Chapter 2. Advanced domain configuration options and the default network channel as well as the administration channel are explained.

Chapter 4—Configuring Naming, Connections, Transactions, and Messaging

The goal of this chapter is to explain how WebLogic Server provides services needed by enterprise applications. The services covered are naming, database connectivity, transactions, and messaging. Each of these services is supported by a different part of Java, and each is covered. JNDI, which is Java Naming and Directory Interface, provides a way to connect to

naming and directory services such as DNS and LDAP. Java applications use JNDI to find resources by name. JDBC, of course, is used for database connectivity. JTA, Java Transaction API, is used to control transactions. The Java Messaging Services, JMS, allow Java applications to create, send, and receive messages.

How each of these services works, as well as how to configure each, is covered in detail. I found the section on JDBC to be very valuable. It explains data sources, which WebLogic Server uses to set up pools of database connections. Creating and configuring data sources is discussed. The JMS section is very thorough.

Chapter 5—Configuring the WebLogic Server Environment

Here we learn about thread pools, which WebLogic Server uses to service requests. The management of thread pools is how WebLogic Server manages its run-time performance. Each request is assigned to an execute thread. WebLogic Server automatically adjusts the number of threads available as the workload varies. Work managers are covered in detail. They allow you to set up different priorities for different requests.

When there are too many requests, WebLogic Server can become overloaded. The options for dealing with overload and failure conditions are covered. Especially interesting to me was the coverage of dealing with stuck threads. This is often the cause of managed servers going to FAILED_NOT_RESTARTABLE state.

The steps needed to set up WebLogic Server self-health monitoring are presented, followed by a more detailed look at how to optimize the network configuration. This involves creating custom network channels.

Chapter 6—Monitoring and Troubleshooting WebLogic Server

WebLogic Server provides many ways to monitor the server instances. Overall, these monitoring features are part of the WebLogic Diagnostic Framework (WLDF). The monitoring dashboard is part of the Administration Console. Examples are shown for generating a diagnostic image capture, which provides a snapshot of what is happening inside the server and other components like JDBC and JMS. These images can be saved in a diagnostic archive.

We have learned in earlier chapters that MBeans are used in Java to store server configuration. The information in MBeans can be gathered by configuring metric collection. The Harvester component of WLDF is then used to collect the specified metrics on the schedule you set up.

WLDF also provides Instrumentation, which means code that can be inserted into server instances and deployed applications to generate extra instrumentation, which is useful when diagnosing server and application issues. The Data Accessor component of WLDF is provided to retrieve the diagnostic information that has been captured. The navigation needed to access the monitoring screens within the Administration Console is shown, and examples of WLST script for monitoring are provided.

WebLogic Server provides logs for both server instance run-time information and application events. The different logs for the domain; each managed server; and the various subsystems, such as JDBC and JTA, are described. The structure of the logs,

how to view them, and how to maintain them are all explained.

This chapter concludes with a discussion of WebLogic Server troubleshooting. Java thread dumps are covered as well as the core dumps generated when a JVM crashes.

Chapter 7—Working with WebLogic Server Clusters

WebLogic Server clusters provide scalability and reliability by providing load-balancing and failover capabilities. This chapter covers how to set up and manage WebLogic Server clusters. WebLogic Server clusters always belong to only one WebLogic Server domain, and the Administration Server for the domain does not run on any of the cluster nodes. This means that for a two-node cluster, for example, you need three nodes all in the same domain. Two of these nodes are the WebLogic Server cluster where applications are deployed, and the third node is where the Administration Server runs.

Three cluster architectures are discussed: basic, multitier, and proxy. The main difference among these is where the web tier, presentation tier, and object tier (the business logic) are running in the cluster.

You can configure a cluster using a WebLogic Configuration Wizard, the Administration Console, or WLST commands. This process involves creating the managed servers that will become the cluster members, creating the cluster itself, and then adding nodes to the cluster.

The options for starting and stopping are covered as well as how to monitor a cluster from within the Administration Console. WebLogic Server offers various algorithms for load-balancing servlets, JSPs, EJBs, and RMI objects on the cluster.

When a server failure occurs, the cluster can migrate an entire server or specific services to another server in the cluster, depending on the configuration chosen.

Chapter 8—Understanding WebLogic Server Application Deployment

Finally, the whole point of using WebLogic Server comes into focus. We set up WebLogic Server to deploy enterprise applications, and now we learn how deployments are done. The various types of applications that can be deployed in WebLogic Server are discussed. Applications are deployed to targets such as managed servers and clusters. The tools supplied with WebLogic Server for deploying applications are described.

WebLogic Server has been described earlier as a container for Java applications. When a Java application is deployed to a managed server, that sever needs to know the environment and product-specific configuration details. These are handled by specifying deployment descriptors and annotations. Deployment plans support moving an application from one server to another. A common example is moving an application from test to production. The deployment descriptors are exported from the test server to a deployment plan. This is then edited for the production environment.

Applications are packaged into an archived file or an exploded archive directory. The details of both and the reasons to use one or the other are covered. Using the Administration Console to deploy an application is presented, and many screenshots help explain the process. Examples of deployment using WLST scripts are also shown.

Chapter 9—Managing WebLogic Server Security

This chapter covers many topics relating to WebLogic security. We begin with an overview of how WebLogic Server security relates to Java EE security. WebLogic Server uses Oracle Platform Security Services, which was designed to secure Fusion Middleware. OPSS provides many security services, including authentication, SSO, authorization, security providers, and security stores.

WebLogic Server security focuses on securing server resources. These include the Administration Console, WLST, and the various server logs. The logical grouping of the various security resources is called a “security realm.” The configuration of all security providers and users, etc., is contained in the security realm, and all of this is stored in an LDAP server or an RDBMS.

Security providers handle specific security functions, such as authentication, identity assertion, and role mapping. The default security realm (myrealm) is created when you create a domain and contains all the required providers. Configuration options for providers are discussed, and screenshots from the Administration Console are included.

Users, groups, roles, and security policies are defined, and the process to create and configure each is covered. The configuration of the embedded LDAP server is explained next, as well how to migrate to a RDBMS security store. This chapter finishes with SSL configuration and overall Best Practices for WebLogic Server security.

Chapter 10—WebLogic Server Performance Tuning

The author tells us that performance tuning covers a broad area, including OS, JVM, server instances, databases, transactions, and more. This means that this chapter can only touch on some highlights of this subject. For each component, the critical issues for performance tuning are identified.

For the WebLogic Server, thread management and network I/O are the hot issues. For WebLogic Server, configuring Work Managers can help, but too much of this can actually hurt. For network I/O, tuning muxers, which read incoming requests on a server, using network channels, and tuning various network layer parameters can improve performance.

Tuning the JVM focuses on memory management. The section explaining how the JVM uses memory was great. I didn’t know how this worked. The issues around garbage collection are interesting, and I learned a lot reading about the configuration options available. Overall best practices for performance tuning are presented.

Conclusion

Oracle is moving to Java EE applications, and this means WebLogic Server is replacing Oracle Application Server. Anyone that supports Oracle systems will likely run into WebLogic Server in the not-too-distant future. This book is densely packed with technical detail for those that will be WebLogic Server administrators. I learned a great deal from reading this book, and it has helped me on the job. ▲ © 2012, Brian Hitchcock

The statements and opinions expressed here are the author’s and do not necessarily represent those of Oracle Corporation.