

Oracle Support for the Spring Framework

An Oracle White Paper
May 2007

Oracle Support for the Spring Framework

INTRODUCTION

In the unique world of software development, freedom of choice is not an abstract concept. The freedom to choose a technology or framework that enables you to build applications in a manner that meets your requirements is an absolute must.

Oracle recognizes this requirement and continues to invest significant resources in developing, testing, optimizing and supporting open source technologies in the Oracle Fusion Middleware stack.

One such open source technology is the Spring Framework, a leading application framework that enables developers to quickly and easily create high quality applications for deployment into high-end application servers. Oracle has invested considerable resources into ensuring that Spring-based applications can be seamlessly deployed into, and run on, the Oracle Containers for J2EE (OC4J) application server.

Customers are using Oracle together with open source technologies – including the Spring Framework – to lower costs and increased developer productivity in mission-critical environments.

EVOLVING REQUIREMENTS

Open source technologies are growing in popularity. Many factors are contributing to this increase, ranging from real and perceived cost savings to a desire for greater flexibility in development and support.

The popularity of Spring as a framework upon which to build enterprise applications is evidenced by its more than 1 million downloads. By providing a clearly layered architecture that allows developers to pick and choose those pieces of the framework they need and abstractions that work seamlessly with standards-compliant service implementations, Spring provides a viable alternative to traditional monolithic Java application design frameworks.

Yet as with any technology, users of an open source framework want to be able to easily deploy, run and manage applications into a runtime environment without the need for extensive customization or configuration. In short, they want their applications to just work.

The Spring Framework is designed to make use of and run on complementary Java Enterprise Edition (Java EE) services provided by a standards-compliant application server such as OC4J.

STANDARDS

An application framework is just that – a set of cohesive components that can be leveraged and extended to facilitate easier, cleaner, more manageable application construction. While it provides the foundation upon which applications are built, a framework is dependent on a robust runtime environment to serve the applications and provide the essential lower-level services they require.

Spring is designed to make use of and run on complementary Java Enterprise Edition (Java EE) services provided by an application server. These services include database connectivity; transaction management (JTA); Web services; messaging (JMS); persistence (JPA); and application management.

Part of Spring’s appeal is that developers can utilize the standard Java EE services provided by the application server in a generic manner, without having to program to vendor-proprietary APIs. The dependence on application server services is thus a developer choice.

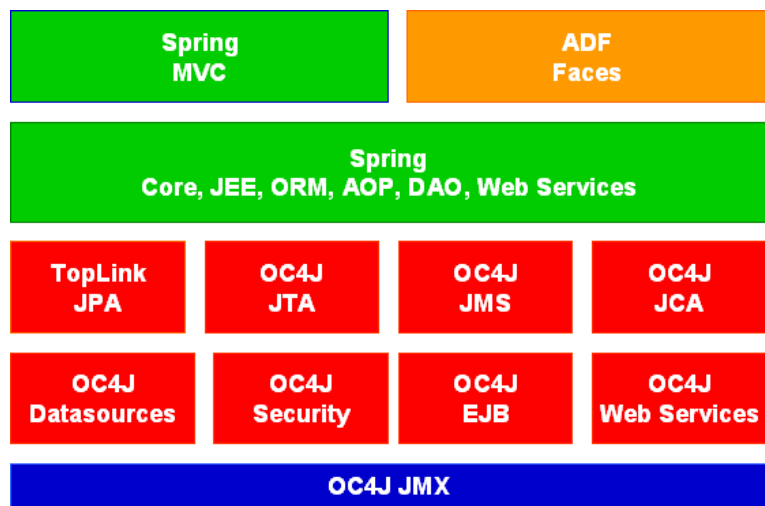


Figure 1: Spring running within the Oracle Containers for J2EE (OC4J) runtime

For maximum success and flexibility however, Spring applications are frequently deployed on fully Java EE compliant containers that provide the services and APIs typically required by an enterprise-class application.

To ensure plug-and-play of Spring applications, the runtime environment must also adhere to standard specifications for deployment, such as the J2EE Application Deployment standard (JSR-88); and for management, such as Java Management (JSR-77) and Java Management Extensions (JMX). Without adherence to such standards, developers would be forced

to design applications for deployment into a specific application server – a clear violation of the development community’s demand for flexibility.

Oracle continues to enhance its support of Spring as part of its overall investment in high-quality open source technologies.

ORACLE’S PRODUCT STRATEGY

Oracle continues to be committed to offering customers choice, flexibility and a lower cost of computing. By investing in open source technologies, Oracle strives to offer open source solutions as a viable choice for application development.

Oracle’s embrace of Spring is evidence of this commitment. The framework has been fully certified on OC4J, ensuring customers that Spring-based applications will perform as intended on the OC4J server. The current production Oracle Containers for J2EE 10g R3 (10.1.3.2) release provides an ideal runtime for Spring applications, due to Oracle’s focus on key Java EE 5.0 APIs like JPA and EJB 3.0 that are complementary to Spring’s development model.

Oracle has an extensive history of close collaboration with Interface21, the developers of Spring. Beginning with the Spring 1.2 release, the framework has supported Oracle TopLink as a data access and persistence solution, offering Spring developers the most robust and scalable persistence implementation on the market.

Oracle continued to work closely with the Spring development team to assist with the Java Persistence API (JPA) integration in Spring 2.0. The result is that Spring 2.0 distribution includes and uses TopLink Essentials – the reference implementation of the Java Persistence API (JPA).

Spring is often used for building transactional applications, as it provides a natural transaction abstraction that lets developers choose their backend transaction engine provider. To provide enhanced transaction management support, OC4J 10.1.3.2 provides a tight integration with the Spring transaction subsystem. This integration provides quick and direct access to the OC4J transaction manager and exposes additional OC4J features to Spring transaction users.

Spring enables application development based on “plain old Java objects”, or POJOs. In Java EE 5, the EJB 3.0 subscribes to a similar POJO component model and offers more natural integration points between Spring and the container than existed in the past. The OC4J container has been enhanced to use and support Spring beans as integrated components within the container. By bringing Java EE and Spring together in unparalleled ways, OC4J can offer developers a superset of features from both.

Integrating Spring into the OC4J runtime is made simple through the flexible OC4J class loading infrastructure. Developers can easily configure shared libraries to make Spring available to all applications across an

Oracle Application Server cluster. The OC4J shared library framework even allows concurrent use of multiple Spring versions by applications.

In scenarios where load time byte code weaving is used, the OC4J class loader framework gives Spring special access to a native class loader API. This implementation enables Spring to participate as a first-class citizen in the container runtime.

Spring applications are deployed like any Java EE application into an Oracle Application Server cluster, automatically inheriting the capabilities inherent in its management framework. Oracle's JSR-88-based deployment framework provides the ability to edit descriptors at deployment times and persist changes to a re-usable deployment plan. No special configuration or code is required; applications are picked up by the OC4J runtime and made available immediately.

Once deployed, the JMX-based Application Server Control management console provides support for administering and managing Spring applications deployed into an Oracle Application Server cluster. The Spring framework allows Spring beans to be exposed as managed resources (MBeans) through the console, enabling them to be treated as standard resources within the OC4J container.

To further support the development process, a Spring plug-in for Oracle's free JDeveloper IDE has been released and can be easily installed using JDeveloper's "Check for Updates" feature. The plug-in provides such enhancements as creation of the Spring library within JDeveloper and transparent registration of Spring schema. Additional features include direct editing of Spring configuration files and Spring property validation capabilities, all within a Spring-friendly, feature-rich development environment.

The JDeveloper plug-in complements the Spring support currently provided in Eclipse, which developers may also use to build and deploy Spring applications into the OC4J runtime.

CONCLUSION

The business and development communities have clearly embraced Spring as a framework for building mission-critical applications. As demand for flexibility and freedom of choice in application frameworks continues to grow, so undoubtedly will the popularity of the Spring Framework.

Oracle remains committed to supporting this technology as a viable option for development of enterprise-quality solutions. Through OC4J, Oracle Fusion Middleware will continue to provide customers with the industry-leading runtime environment for deploying, hosting and managing Spring-based applications.



Oracle Support for the Spring Framework

May 2007

Author: Dan Hynes

Contributing Authors: Mike Kieth, Mike Lehmann, Shaun Smith

Oracle Corporation

World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

oracle.com

Copyright © 2007, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.