



IBM Software Group

IBM WebSphere Application Server V7.0 Feature Pack for Service Component Architecture V1.0.1

JEE integration overview



@business on demand.

© 2009 IBM Corporation
Updated November 4, 2009

This presentation will cover the overview of JEE integration in the SCA feature pack.

Overview

- JEE integration for SCA allows existing WebSphere® customers to tie into their JEE applications
- Support for the Open Service Orientated Architecture (OSOA) organizations and JEE Integration for SCA V1.0.0 Specifications



The SCA programming model supports Java™ EE integration. As a result, you can take advantage of SCA annotations to enable the Java EE components, such as session beans, message driven beans, or Web components to consume SCA services. By using Java annotations that apply to SCA, you can enable an existing Java EE component so that it is recognized as an SCA component and so that the component can participate in an SCA composite

JEE Integration for SCA allows existing WebSphere customers to tie into their JEE applications more naturally and reuse JEE skills providing for a strong value-add for SCA applications deployed in WebSphere.

Support for the Open Service Orientated Architecture (OSOA) organizations JEE Integration for SCA V1.00 Specifications demonstrates IBM's commitment and leadership in Open Programming Initiatives.

Overview – Continued

- Java EE is the standard for Java-based enterprise applications today
- Does not define concepts required in SCA
 - ▶ Extensibility of component implementation technologies
 - ▶ Extensibility of transport and protocol abstractions
 - ▶ a notion of cross-application assembly and configuration



Java EE is the standard for Java-based enterprise applications today. While it offers a rich set of technologies, it does not define important concepts that are inherently required in service oriented architectures such as

- Extensibility of component implementation technologies
- Extensibility of transport and protocol abstractions

and a notion of cross-application assembly and configuration

SCA provides a standardized and extensible assembly language and methodology that can be layered on top of existing component models and runtimes.

Programming model

The SCA programming model behavior in JEE components is defined in the OSGi Specification for JEE Integration – Here is an example:

```
package services.accountdata;
import javax.ejb.Stateless;
import org.osoa.sca.annotations.*;
import services.backend.Brokerage;

@Stateless // EJB annotation
public class AccountServiceImpl implements AccountService {
    @Reference protected Brokerage backend; // SCA reference
    @Property protected String currency; // SCA property
    @Context protected SCAContext context; // SCA context

    public AccountReport getAccountReport(String customerId) {
        acctValue BigDecimal = Brokerage.getAccountValue(customerId,"IBM"); // use injected reference
        if (currency != "US DOLLARS") { // use injected property
            moneyChangerService = context.getService(moneyChanger.class,"MoneyExchange"); // use injected context
            acctValue = moneyChangerService(current,acctValue); // invoke SCA service
        }
        return backend(customerId, acctValue);
    }
}

(*) As documented, by specification, life cycle of this component is managed by JEE rules.
```



This example shows how a JEE component consumes an SCA service by using SCA annotations. Note the use of these annotations:

1. **@property** - The **@Property** annotation is used to define an SCA property
2. **@Reference** - Accessing a service using reference injection is done by defining a field, a setter method parameter, or a constructor parameter typed by the service interface and annotated with an **@Reference** annotation.
3. **@context** - The **@Context** annotation type is used to annotate a Java class field or a setter method that is used to inject

a composite context for the component. The type of context to be injected is defined by the type of the Java class field or type of the setter method input argument, the type is either **ComponentContext** or **RequestContext**.

Section

Summary

To summarize...

Summary

- SCA programming model supports Java EE integration
- JEE Integration for SCA allows existing WebSphere customers to tie into their JEE applications more naturally and reuse JEE skills



The SCA programming model supports Java EE integration. JEE Integration for SCA allows existing WebSphere customers to tie into their JEE applications more naturally and reuse JEE skills providing for a strong value-add for SCA applications deployed in WebSphere.

References

- **SCA Java EE integration specification:**
http://www.osoa.org/download/attachments/35/SCA_JAVAEE_Integration_V100.pdf?version=1
- **IBM Education Assistant**
http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wasfpsca/plugin_coverpage.html
- **SCA feature pack information center**
http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.soafep.multiplatform.doc/info/welcome_nd.html
- **SCA white papers**
http://www.ibm.com/developerworks/websphere/library/techarticles/0812_beck/0812_beck.html



References

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WASv7SCA101_JEE_Integration_Overview.ppt

This module is also available in PDF format at: ._WASv7SCA101_JEE_Integration_Overview.pdf



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

EJB, Java, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2009. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.