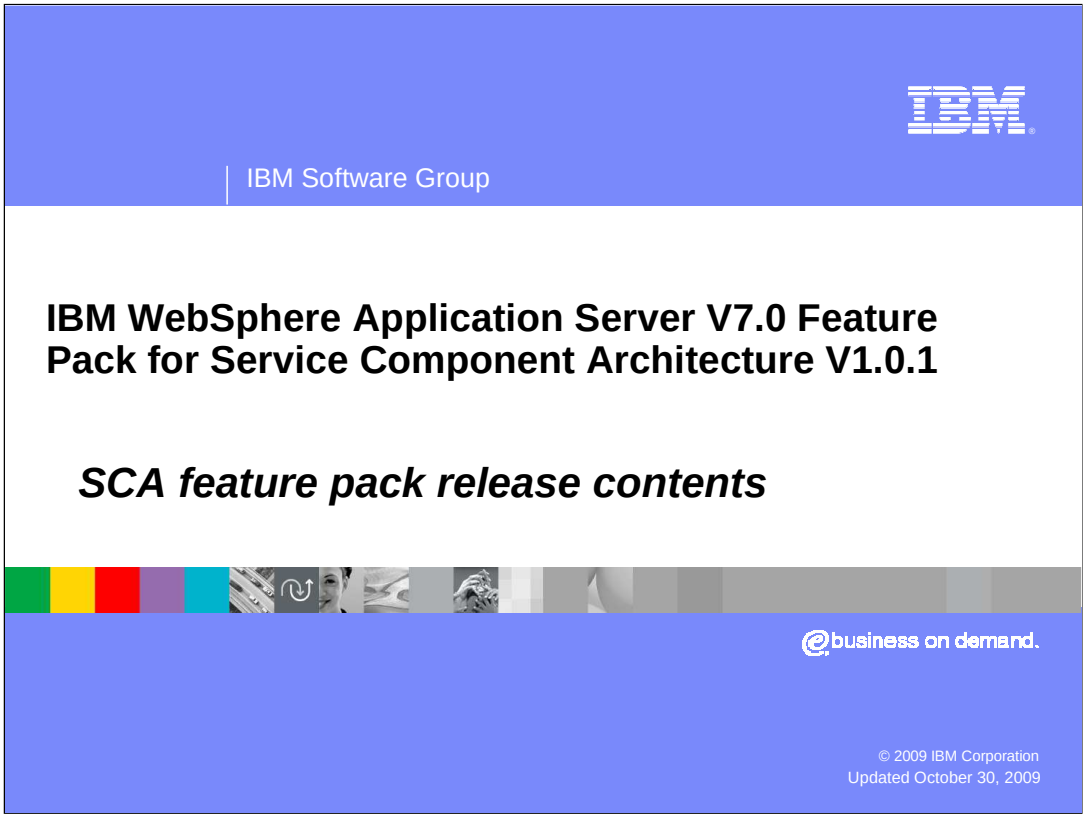
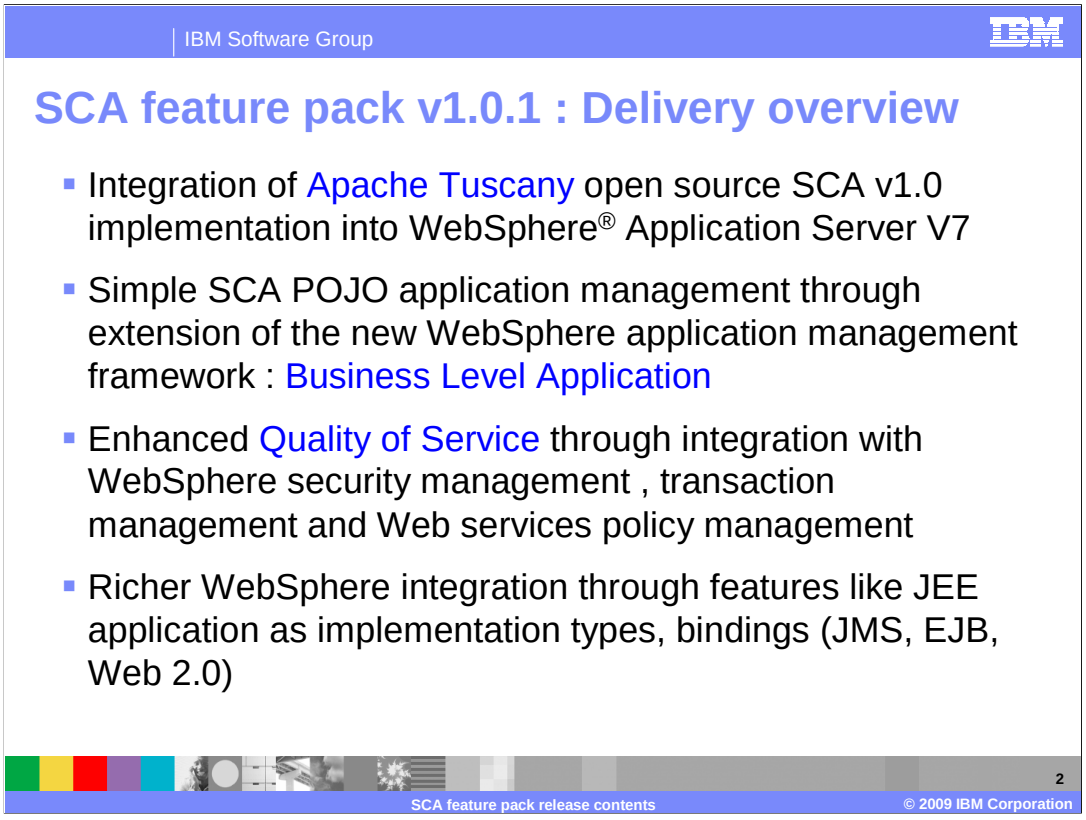


This presentation covers the SCA feature pack. First you will see a high-level overview, then, additional features and enhancements along with scenarios that apply to the feature pack.



IBM WebSphere Application Server V7 feature pack for SCA delivers critical technology that enables adoption of key Service-Oriented Architecture (SOA) principles. As part of the larger SOA foundation, which straddles all of IBM software brands, this feature pack delivers an integrated, open implementation of Service Component Architecture (SCA). WebSphere has taken the open source SCA V1.0 implementation from Tuscany, an Apache project, and integrated it with WebSphere Application Server. This integration ensures that all of WebSphere's capabilities work together with SCA applications to provide a natural environment for both the SCA and existing applications.

IBM WebSphere Application Server V7 feature pack for SCA and the underlying Tuscany framework is a "proof-point" delivery of SCA built using a plug-in concept. IBM plans to augment and enhance this initial support in upcoming releases by providing additional plug-in capabilities. In this release there have also been additions in the application management with simple SCA POJO applications being extended into business level applications in WebSphere Application Server. Quality service has also been enhanced through the integration with WebSphere security management, transaction management and Web services policy management. Integration through features like JEE application as implementation types, bindings (JMS, EJB, Web 2.0) were added.

New in SCA feature pack V1.0.1 delivery

- JMS Bindings (services, references)
- JEE Integration
  - SCA programming model injection into Java™ EE apps
  - JEE artifacts
- Web 2.0 support
  - ATOM
  - JSON-RPC
- Spring support
- SDO 2.1.1 support (JSR235 compliant)
  - Additional technology samples
    - JMS sample
    - JEE sample

SCA feature pack V1.0 was delivered last year, this year SCA feature pack V1.0.1 has added additional features. Included in this release is JMS Integration for the OSOA (JMS Binding support). What this means is that IBM adopted OASIS <wireFormat> technology which allows for simple message transformations. This includes an API and hook point to plug in a user routine to convert between custom JMS payload formats and SCA application data, and another API and hook point to plug in a custom operation selector. Integration with WebSphere Default Provider and WebSphere MQ has been added. Support to dynamically create resources when they do not already exist has also been added. As far as integration goes, JEE Integration support including annotation injection in JEE components, EARs as composites, has also been added. For Web 2.0 support, support for the ATOM, JSON-RPC for access protocol to SCA services was enabled. Implementation.Widget allows JavaScript UI components to be composed on the server and returned to browsers or other Java script enabled runtimes, which allow SCA annotations to be used. DOJO subroutines have been inserted to connect the UI to the back end services.

Support for components implemented with the Spring Framework version 2.5.5 was added. This provides developers the ability to wrapper J2SE applications written for the Spring Framework platform in the SCA composition model. Additionally Implementation.Spring allows SCA to compose spring applications with other types of applications and expose the spring application through the various bindings supported by the SCA feature pack.

SDO 2.1.1 Support was added as a data-binding on SCA service definitions which are JSR235 compliant, built on with XML feature pack as a pre-requisite. This support also provides a Business Object SPI which allows WebSphere Process Server to build business objects on top of SDO infrastructure, but allows for compatibility with previous versions of WebSphere Process Server. Additional technology samples such as JMS sample and Spring sample have also been added.

SCA feature pack highlights (scenarios)

- Service composition
  - "Use what you have got and run it where it lives," or "Use your existing services to create new ones."
- Service development
  - "Know only what you need to know to get your job done," or "Maintain proper separation of concerns.
- Service agility and flexibility
  - ability to rewire, compose, and assemble business logic without impacting the business logic

Here are some scenarios that apply to SCA feature pack highlights. First, service composition; businesses today are challenged not only by competitors, but by social and economic pressures that directly affect their information technology systems. As businesses adopt SCA and build a growing inventory of business services, there is a real need to be able to compose, reuse, and otherwise assemble new services from those existing business services. One of the primary objectives of this release is to highlight usage of SCA as a coarse-grained composition model that can be used to assemble and compose existing services in your enterprise. The key principle of SCA demonstrated by this support can be described as "Use what you have got and run it where it lives," or "Use your existing services to create new ones."

Second, service development; SCA has a language-neutral programming model for which there are multiple language-specific specifications defined at OSOA. The concepts of SCA apply broadly across both Java and non-Java application environments. The SCA component model has at its heart a strong focus on a proper separation of concerns. The service consumer business logic author should not need to know the details of the service implementation. For instance, a Java service consumer should not be burdened with having to know that a target service is implemented using C++ or COBOL. Therefore another key objective of this delivery of SCA is to highlight the ease-of-use characteristics of SCA service development in Java. This is accomplished by demonstrating annotated plain-old java-object (POJO) components deployed using simple JAR packaging schemes, an easy to use assembly model, and powerful wiring abstractions that enable service definition over different transports and protocols. The key principle of SCA demonstrated by this support might be described as "Know only what you need to know to get your job done," or "Maintain proper separation of concerns." Note that for this feature pack, the programming language is Java.

Third, service agility and flexibility; SCA feature pack highlights the flexibility and agility of metadata bindings, and the appropriate separation of concerns. The ability to rewire, compose, and assemble business logic without impacting the business logic itself is key.

Functional description – highlights (1)

- Based on Tuscany SCA Java implementation
- POJO service component implementations
  - Tuscany component container
- Bindings for
  - Web services (WS feature pack), and
  - EJB2 and EJB3 (currently IIOP)
  - JMS
- Administrative extensions – Panels, commands, application install (based on BLA), MBeans
- Native SCA packaging model (jars)

The SCA feature pack is consuming a subset of Tuscany that implements SCA V1.0. In particular it includes assembly and Java support such as POJO service component implementations. It also includes bindings such as Web services, EJB2 and EJB3, and JMS. Administrative extensions include application management piece on business level applications. It also encompasses native SCA2 packaging model which include jars.

Functional description – highlights (2)

- Policy framework – interaction, implementation policy
  - Qos Extensions (security, transactions, reliability)
  - Web service profile integration
- JAXB \SDO data bindings
- RAS (log/trace), FFDC, ARM
- Technology samples
- OSGI Bundles – runtime and WebSphere Application Server extensions are separate
- Dynamic Resource Creation for development ease of use
  - JMS resources needed for bindings can be automatically created for messaging scenarios

In the area of policy framework, security policies, transaction policies, and reliability are also included in the SCA feature pack. In databindings, JAXB and SDO, data binding are supported. Technology samples such as multiservice, candystore, jobbank and so on, showing different bindings have been implemented are also included. Also new is the dynamic resource creation for development use. Dynamic resource creation enables JMS resources needed for bindings to be automatically created for messaging scenarios.

Section

Summary and references

The next section provides a summary of this presentation.

Summary

- IBM WebSphere Application Server V7 feature pack for SCA
  - makes service oriented architecture simpler for developers
  - preserves and enhances the value of existing assets
  - provides a consistent system-wide abstraction
    - service abstraction
    - can apply QoS policies (security, reliability, …) at the service abstraction
  - Simplicity, Consistency, Abstraction

SCA feature pack makes SOA simpler for developers in that application is structured as services and components. Business logic is not coupled to deployment infrastructure and there is a wide choice of component kinds (use the right tool for the job). As a result, this makes developers stay focused on solving business problems, rather than getting bogged down in the individual complexities of the technologies that connect service consumers and service providers. SCA feature pack preserves and enhances the value of existing assets which can be exposed as SCA services. SCA feature pack can model and integrate assets from a heterogeneous mix of new and existing runtime environments and provide a consistent system-wide abstraction. Service abstraction captures the SOA design and its mapping to underlying implementations. SCA feature pack can apply QoS policies (security, reliability …) at the service abstraction level across different runtime implementations. Business pressures from competition, marketplace, mergers and acquisitions, and so on requires applications to adapt to rapid change.

SCA applications are resilient to change and insulated from technology and infrastructural changes. They are architected to accommodate new technologies as they emerge.

Resources

- Open Service Oriented Architecture Web site for SCA v1.0 Specifications
  - http://www.osoa.org/
- OASIS Open CSA Web site for SCA v1.0
  - http://www.oasis-opencsa.org/sca
- Apache Tuscany Web site
  - http://incubator.apache.org/tuscany/
- SCA feature pack support Web site

http://www-01.ibm.com/support/docview.wss?rs=180&context=SSEQTP&dc=DB600&uid=swg21329175

Here are some resources that can be useful..

More resources

- DeveloperWorks

http://www.ibm.com/developerworks/websphere

- SCA feature pack V1.0 IBM Education Assistant content

http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.wasfpsca/plug-in_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

- Exploring the WebSphere application Server feature pack for SCA

http://www.ibm.com/developerworks/websphere/library/techarticles/0812_beck/0812_beck.html

- WebSphere Application Server V7 service component architecture FAQs

http://www-01.ibm.com/software/webservers/appserv/was/featurepacks/sca/faq.html

More resources..

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_Overview_SCAfepReleaseContents.ppt

This module is also available in PDF format at: ../WASv7SCA101_Overview_SCAfepReleaseContents.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, J2SE, Java, JavaScript, Open Service, 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.

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.