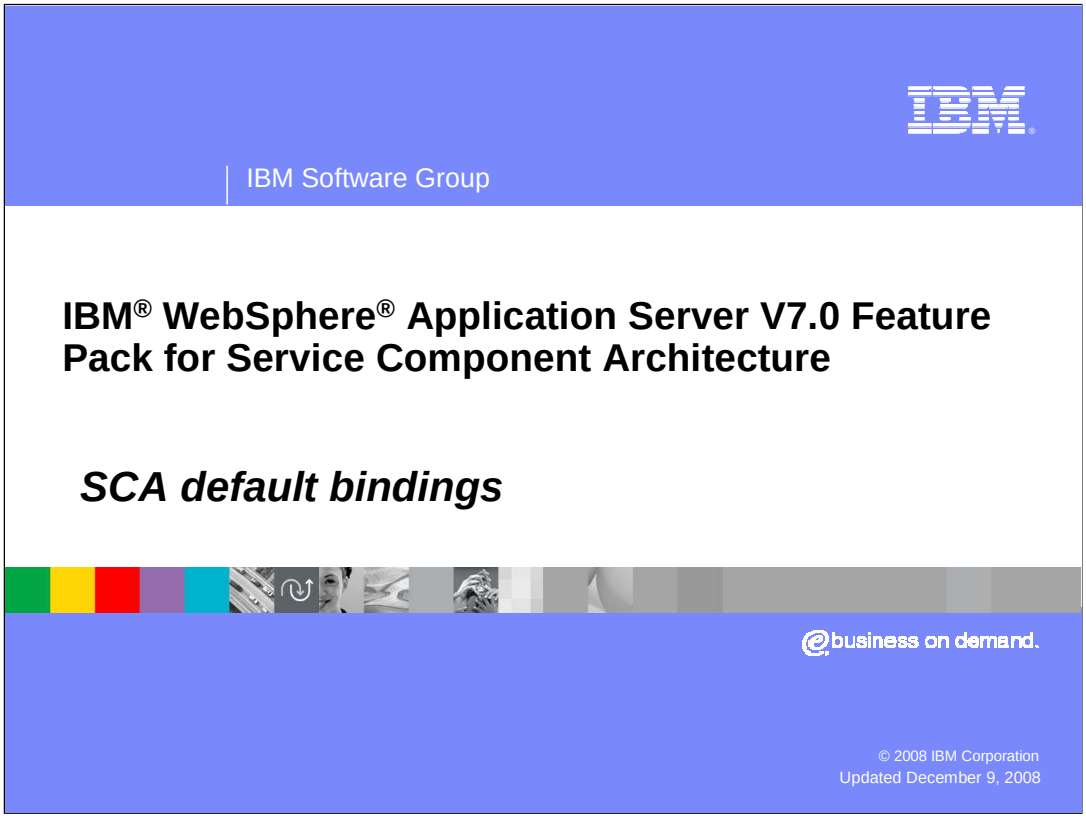


This presentation will cover the SCA default bindings

Default binding

- Also known as SCA binding
- Only used when a service and its client are both running in the same domain
- Used when no other binding is specified for a configuration of a component reference or service
- Default binding is the preferred binding to use for an SCA client invoking an SCA service
- Not intended to be interoperable in any way with other SCA runtimes' implementations

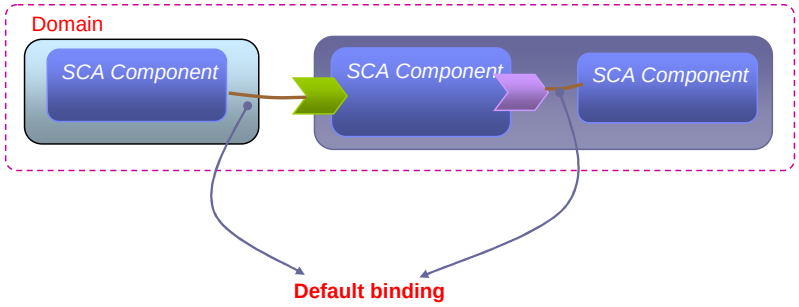
Every SCA runtime also provides an *SCA binding*. This binding is also known as the default binding meaning that it is the binding used when no other binding is specified for a configuration of a component reference or service. It is the natural binding to use for SCA-to-SCA invocations such as SCA client invoking SCA service.

The protocol this binding uses is not specified, therefore the SCA binding is only used when a service and its client are both running in the same domain. Since every vendor wants applications built on its products to perform well, it is safe to assume that this binding will most often use a binary protocol. This is not required, however; an SCA runtime is free to choose different protocols in different situations, all of which fall under the umbrella of the SCA binding. Additionally, this binding is not intended to be interoperable in any way with other SCA runtimes' implementations. For example, you cannot call over the default binding to invoke a WebSphere Process Server SCA service from an SCA feature pack client.

*** Note that SCA feature pack default binding implementation serializes xml although not specified in the specification.

The SCA binding can be used for service interactions between references and services contained within the SCA domain. It provides a very simple, easy-to-use way for an assembler to wire a reference to a service within an SCA composite, or at the Domain level. It tells the SCA runtime to make the connection between the reference and the service, without being prescriptive about how it should be done. The SCA runtime is then free to use any communication method it has available that is suitable for implementing the wire. The form of its implementation is not defined by the SCA specification. It can be implemented in different ways by different SCA runtimes, and is therefore not an interoperable binding type. For interoperability, an interoperable binding type such as the Web service binding should be used.

SCA domains and default binding



- Components communicating within the same domain do not need explicit bindings specified

As mentioned earlier, a component that communicates with another component in the same domain, even one in another process or on another machine does not need any explicit bindings specified. Instead, the SCA default *binding* is used. Domains are an important concept in SCA. To see why, realize that even though SCA allows creating distributed applications, it does not fully define how components on different machines should interact. As a result, the communication among these components can be implemented differently by different products. However, an SCA runtime can allow a vendor to create a *container* that plugs into that runtime to support a particular technology as an example. A domain can contain one or more composites, each of which has components implemented in one or more processes running on one or more machines.

For the SCA feature pack the domain is the Network Deployment cell.

As an example, suppose a division of a large firm chooses a particular company as its SCA vendor. This division is likely to install their chosen vendor's SCA runtime on several machines. This is not an unreasonable expectation, because it mirrors how organizations have typically purchased and installed JEE products. These SCA runtimes will likely be managed by the same group of people, and this set of systems—with a common vendor's runtime technology and common management. This example provides the primary example of a domain.

Summary

- Default binding, which is the default binding, is used when no other binding is specified for a configuration of a component reference or service.

Default binding, which is the default binding, is used when no other binding is specified for a configuration of a component reference or service.

References

- SCA specifications:

http://www.osoa.org/display/Main/Service+Component+Architecture+Specifications

The set of SCA specifications can be found at the address shown here.

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_WASv7SCA_SCABindings_Default_Binding.ppt

This module is also available in PDF format at:
../WASv7SCA_SCABindings_Default_Binding.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:

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

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.