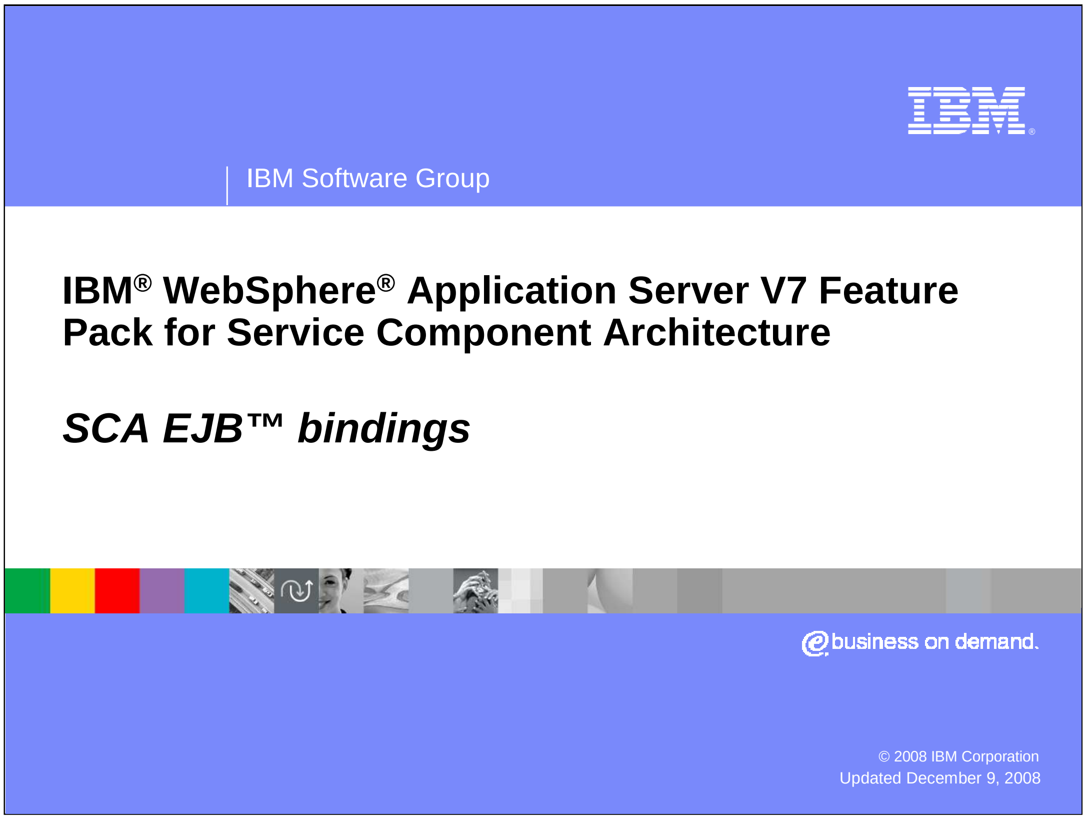


IBM Software Group

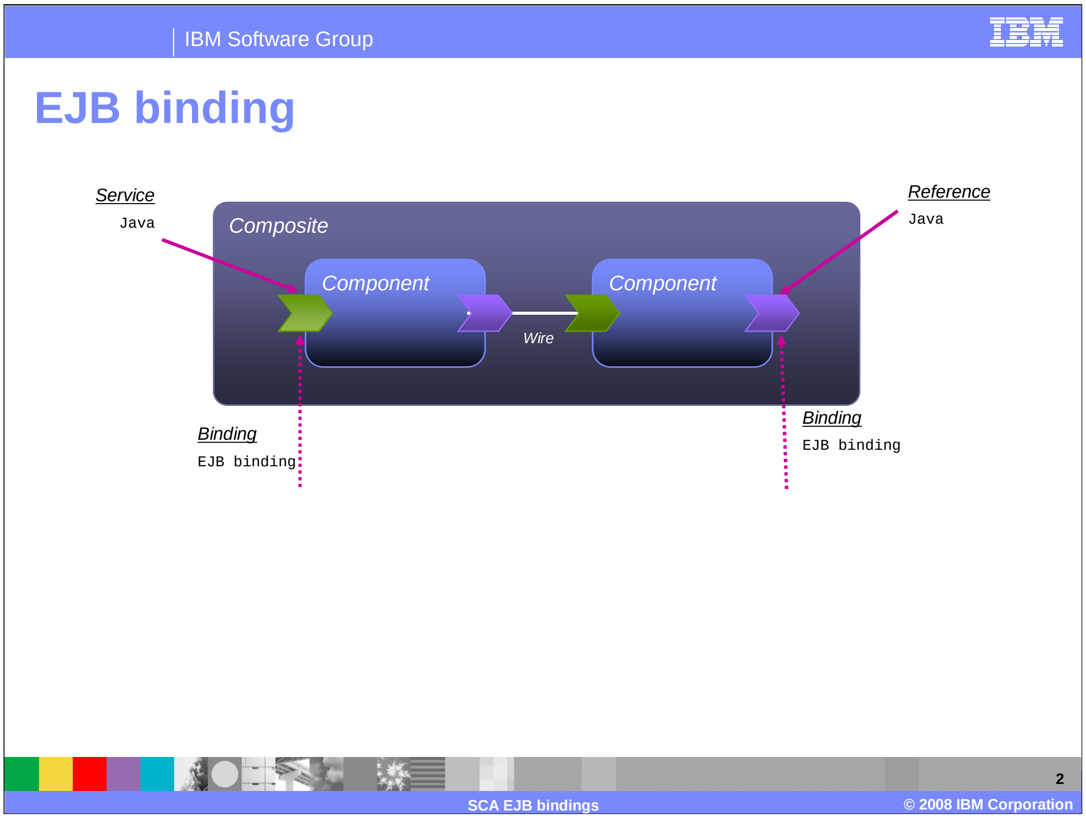
IBM® WebSphere® Application Server V7 Feature Pack for Service Component Architecture

SCA EJB™ bindings

business on demand.

Updated December 9, 2008

This presentation covers SCA EJB binding.



EJB binding

Service
Java

Composite

Component

Component

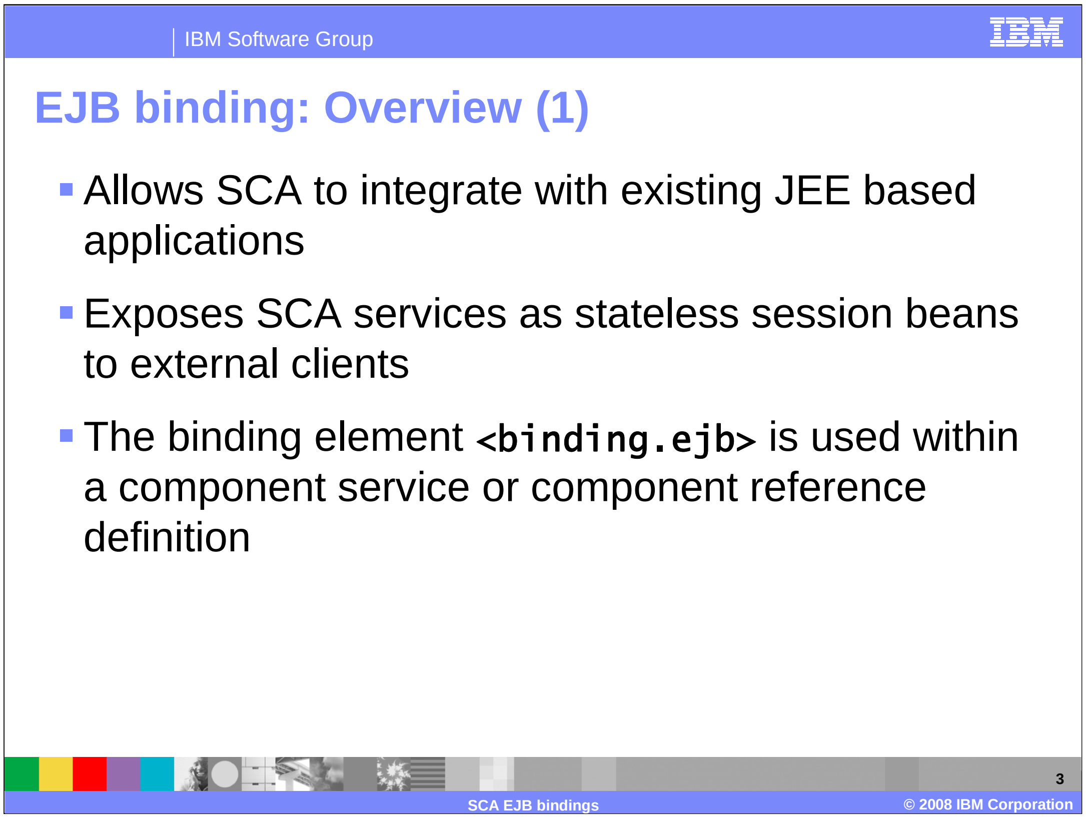
Wire

Reference
Java

Binding
EJB binding

Binding
EJB binding

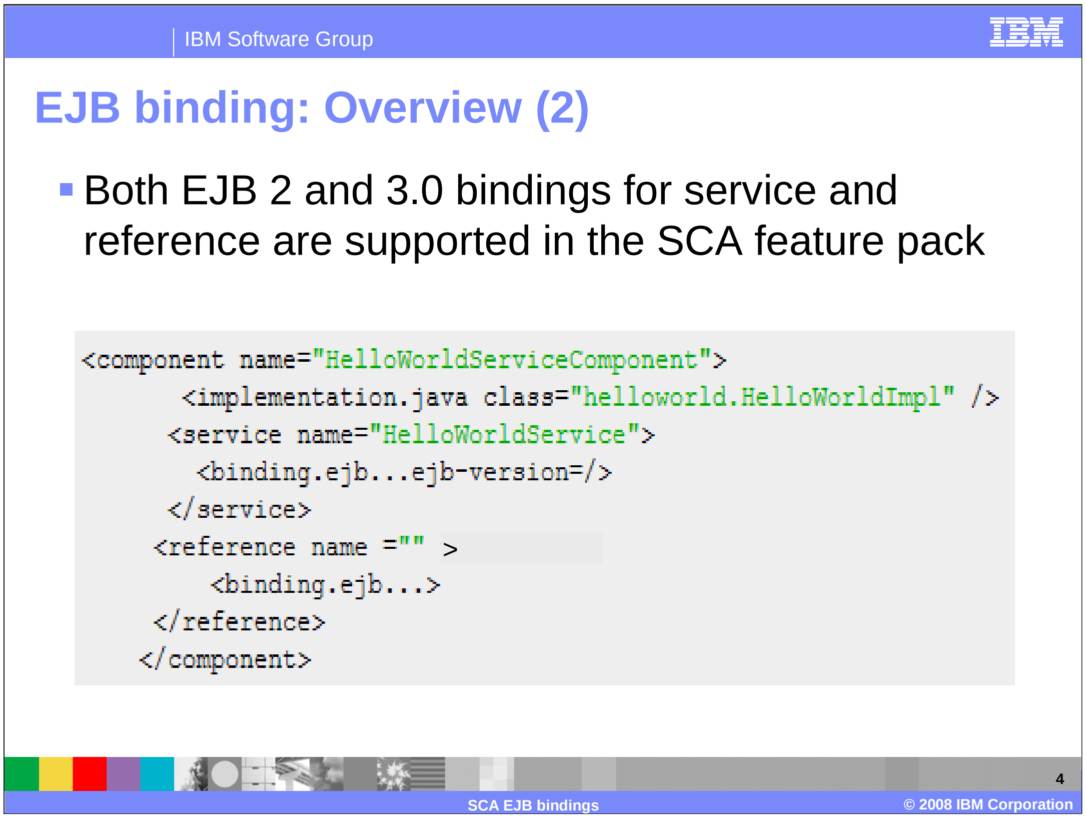
This diagram shows the use of the EJB SCA binding on both services and references.



EJB binding: Overview (1)

- Allows SCA to integrate with existing JEE based applications
- Exposes SCA services as stateless session beans to external clients
- The binding element `<binding.ejb>` is used within a component service or component reference definition

EJB binding allows SCA to integrate with existing JEE based applications by exposing SCA services as stateless session beans to external clients. The EJB Session Bean binding enables SCA developers to treat previously deployed session beans as SCA services, by wiring them into an SCA assembly as an SCA reference. SCA service deployers expose an SCA service as a session bean for consumption by Java EE applications. The binding element used within a component service or component reference for EJB binding is `<binding.ejb>`.

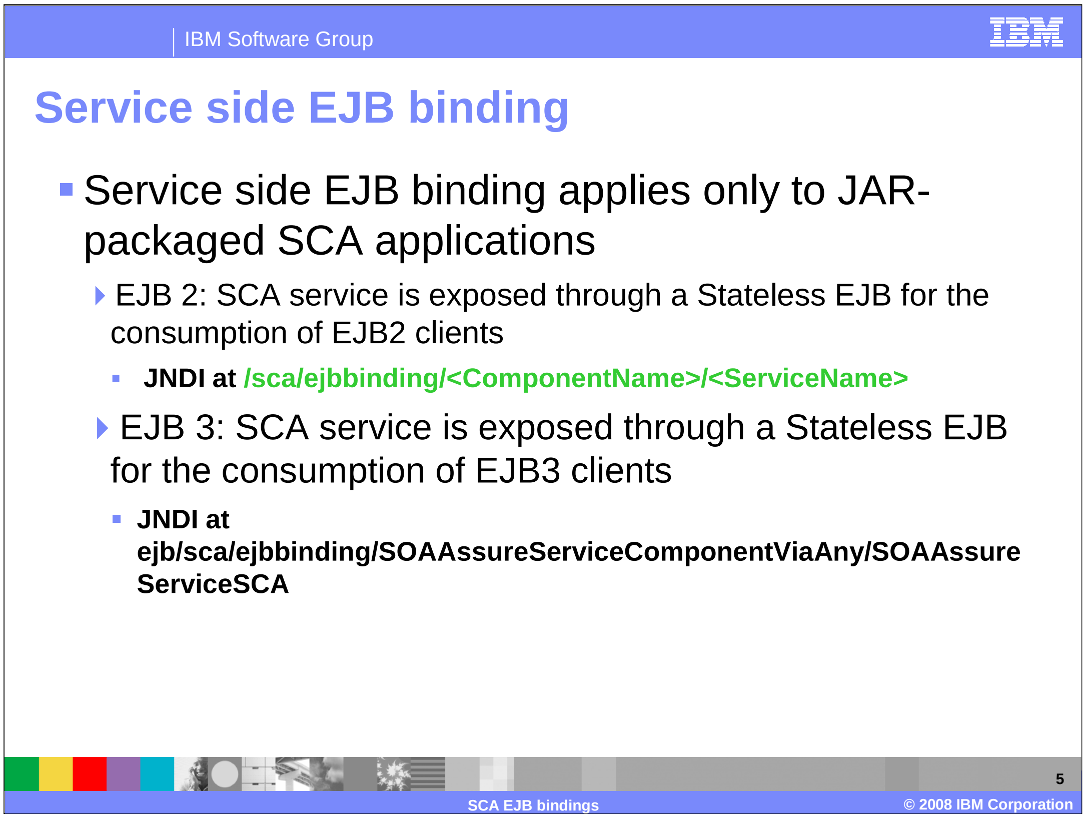


EJB binding: Overview (2)

- Both EJB 2 and 3.0 bindings for service and reference are supported in the SCA feature pack

```
<component name="HelloWorldServiceComponent">
       <implementation.java class="helloworld.HelloWorldImpl" />
      <service name="HelloWorldService">
        <binding.ejb...ejb-version=/>
      </service>
     <reference name ="" >
         <binding.ejb...>
     </reference>
    </component>
```

In the SCA feature pack, support is provided for EJB binding in both 2 and 3.0 style for both service and reference. Support is also provided for reference target. However, something to keep in mind is that, reference target is more of a corner case with EJB binding as compared to what it is in the Web service binding. There is really no reason to go from SCA-to-SCA (such as SCA client invoking SCA service) over EJB binding.



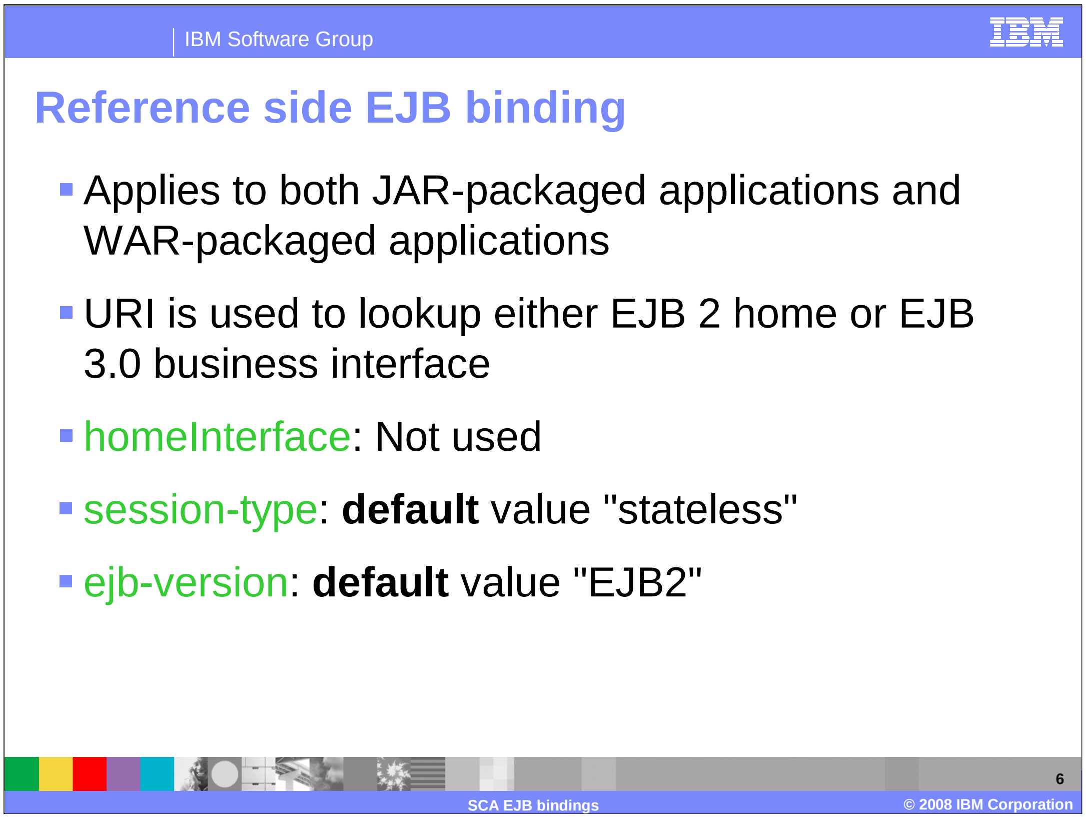
Service side EJB binding

- Service side EJB binding applies only to JAR-packaged SCA applications
  - EJB 2: SCA service is exposed through a Stateless EJB for the consumption of EJB2 clients
    - **JNDI at /sca/ejbbinding/<ComponentName>/<ServiceName>**
  - EJB 3: SCA service is exposed through a Stateless EJB for the consumption of EJB3 clients
    - **JNDI at ejb/sca/ejbbinding/SOAAssureServiceComponentViaAny/SOAAssureServiceSCA**

Service side EJB binding applies **only** to JAR-packaged SCA applications.

For EJB2--> SCA service is exposed through a Stateless EJB for the consumption of EJB2 clients. The stateless EJB for the service is bound to JNDI as shown.

Similarly, for EJB3 --> SCA service is exposed through a Stateless EJB for the consumption of EJB3 clients. The stateless EJB for the service is bound to JNDI.



Reference side EJB binding

- Applies to both JAR-packaged applications and WAR-packaged applications
- URI is used to lookup either EJB 2 home or EJB 3.0 business interface
- homeInterface: Not used
- session-type: **default** value "stateless"
- ejb-version: **default** value "EJB2"

Reference side EJB binding applies to **both** JAR-packaged applications and WAR-packaged applications if not otherwise stated. URI is used to lookup either EJB 2 home or EJB 3.0 business interface, following the naming convention of JEE if you are wiring to an existing JEE EJB module. Reference side can define a reference target (<reference target="compname/serviceName“) if the target is SCA service with EJB service binding" and deployed in the same server.

If you wire to an SCA service with binding.ejb, then use the values shown:

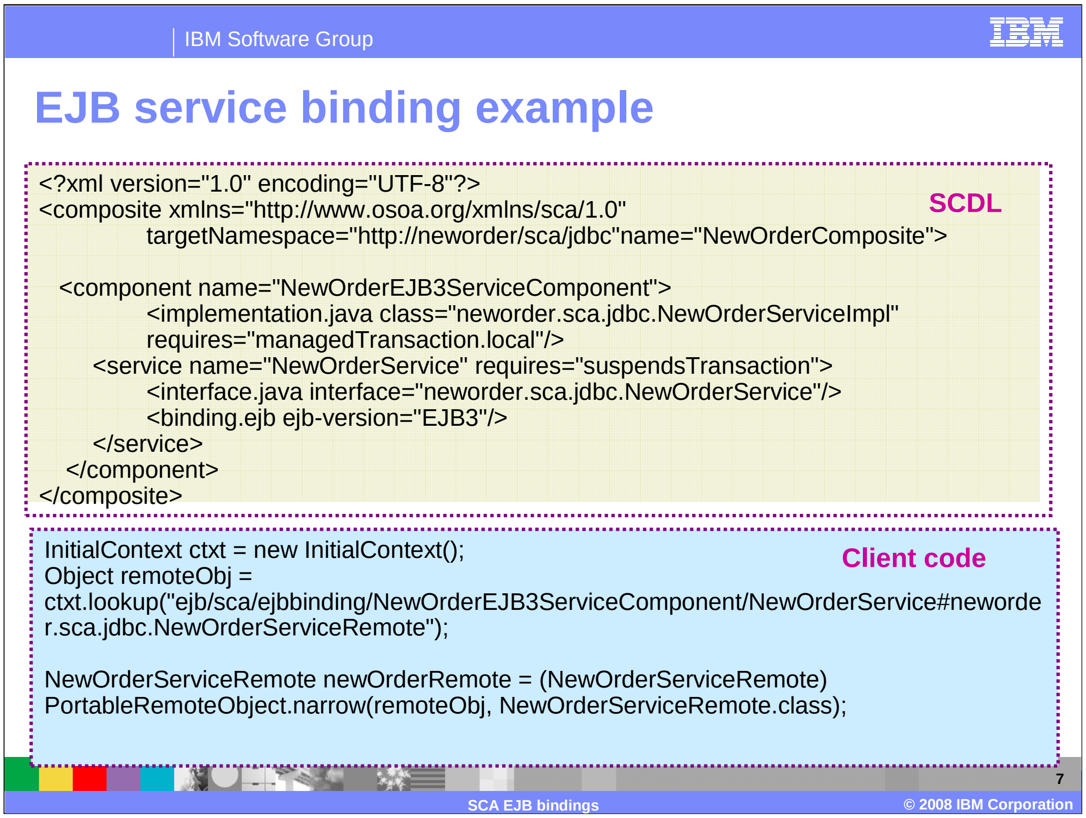
(1) homeInterface: Not used

(2) session-type: default value is "stateless“

(3) ejb-version: **default** value "EJB2"

Note that one can wire reference with EJB binding to a service with EJB binding even though not really a typical usecase. EJB service binding is mainly there to expose the SCA component to pure JEE clients or callers. And EJB reference binding can access Stateless EJB which is running under pure JEE environment.

When used on a reference, the EJB binding specifies the means for connecting an SCA component to a previously deployed or co-deployed session bean. The reference interface used with the EJB binding can be either a remote or local session bean interface. SCA deployment logic and the binding implementation will check the reference interface class to determine whether it is local or remote. If an SCA component needs to access both the local and remote interface of a session bean, then this should be modeled in SCA assembly through two references - one with the local interface and one with the remote interface.



IBM Software Group

EJB service binding example

SCDL

```
<?xml version="1.0" encoding="UTF-8"?>
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0"
        targetNamespace="http://neworder/sca/jdbc"name="NewOrderComposite">

  <component name="NewOrderEJB3ServiceComponent">
        <implementation.java class="neworder.sca.jdbc.NewOrderServiceImpl"
        requires="managedTransaction.local"/>
     <service name="NewOrderService" requires="suspendsTransaction">
        <interface.java interface="neworder.sca.jdbc.NewOrderService"/>
        <binding.ejb ejb-version="EJB3"/>
     </service>
   </component>
</composite>
```

Client code

```
InitialContext ctxt = new InitialContext();
Object remoteObj =
ctxt.lookup("ejb/sca/ejbbinding/NewOrderEJB3ServiceComponent/NewOrderService#neworde
r.sca.jdbc.NewOrderServiceRemote");

NewOrderServiceRemote newOrderRemote = (NewOrderServiceRemote)
PortableRemoteObject.narrow(remoteObj, NewOrderServiceRemote.class);
```

7

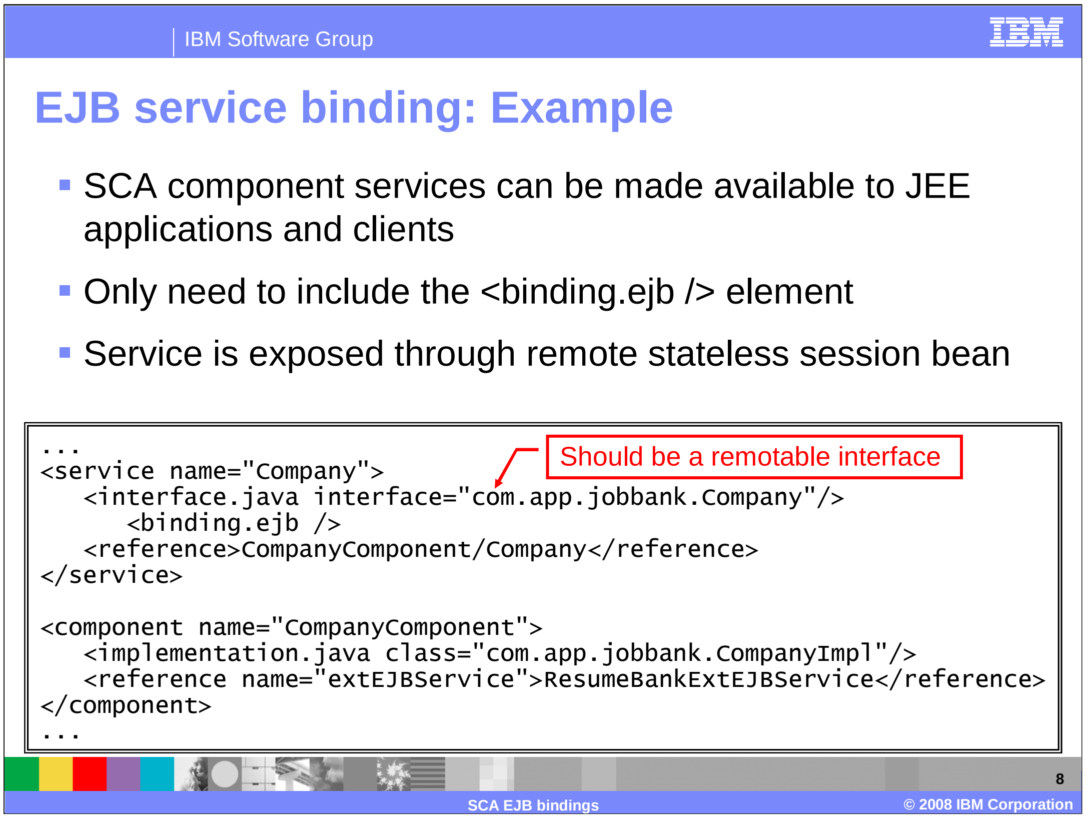
SCA EJB bindings

© 2008 IBM Corporation

This is an example of SCDL (Service Component Definition Language) that has a service exposed over an EJB 3.0 binding. Also shown is the client code for it.

A client that wants to invoke the resultant enterprise bean would treat it like any other enterprise bean and not like a regular SCA service.

Not that for Web service binding, CompositeContext.getService is not supported for a non-SCA binding, therefore, a getService() on the CompositeContext would not work here.



IBM Software Group

EJB service binding: Example

- SCA component services can be made available to JEE applications and clients
- Only need to include the <binding.ejb /> element
- Service is exposed through remote stateless session bean

```
...
<service name="Company">
   <interface.java interface="com.app.jobbank.Company"/>
      <binding.ejb />
   <reference>CompanyComponent/Company</reference>
</service>

<component name="CompanyComponent">
   <implementation.java class="com.app.jobbank.CompanyImpl"/>
   <reference name="extEJBService">ResumeBankExtEJBService</reference>
</component>
...
```

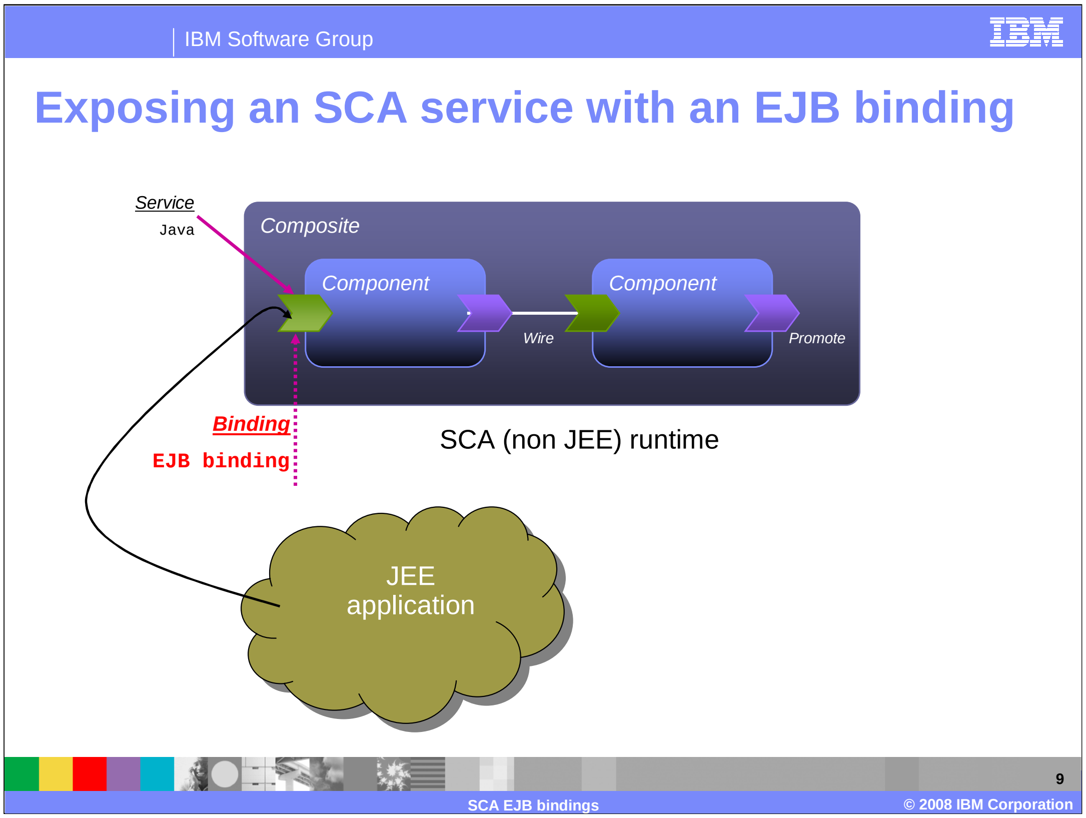
Should be a remotable interface

8

SCA EJB bindings

© 2008 IBM Corporation

This is an example of an EJB service binding. SCA component services can be made available to JEE applications and clients by exposing them through remote stateless session bean. In this scenario, <binding.ejb /> element needs to be included.



IBM Software Group

Exposing an SCA service with an EJB binding

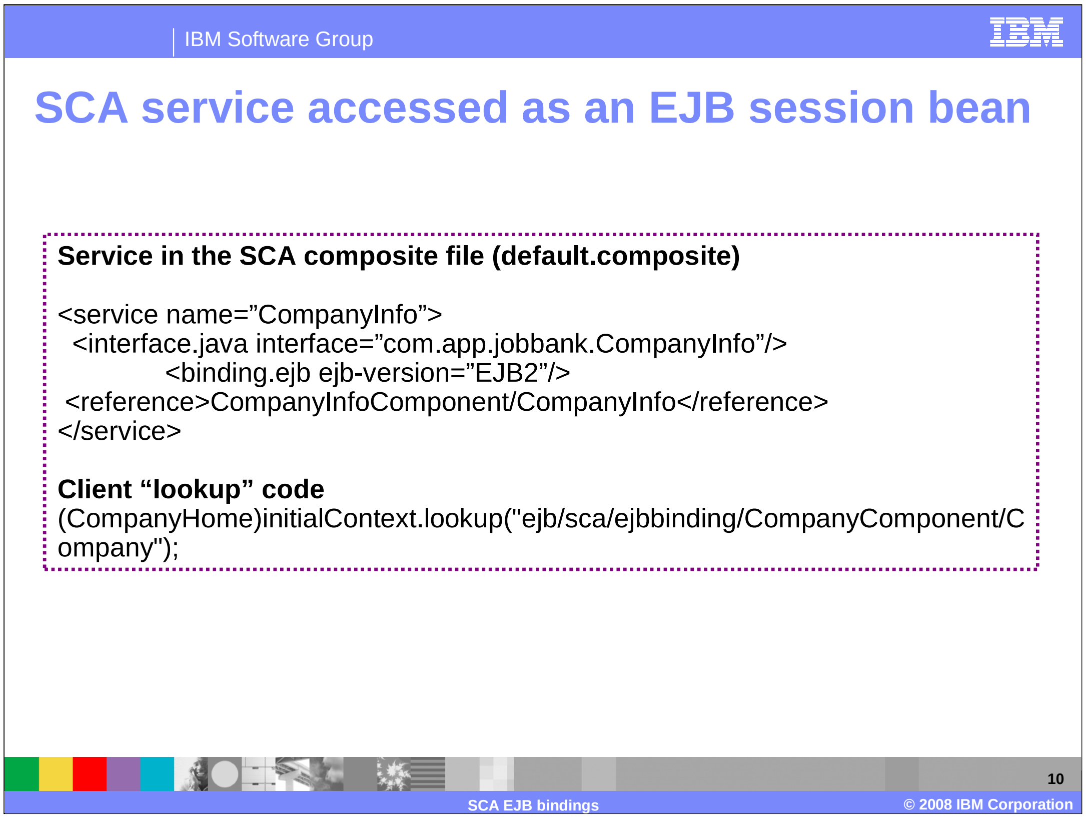
9

SCA EJB bindings

© 2008 IBM Corporation

This pictures shows an SCA Service accessed as an EJB Session Bean.

An SCA service is developed that will be called from a Java EE environment. Suppose the Java EE programmer doesn't know the SCA programming model. Suppose the programmer wants to use the Java EE programming model that he knows to invoke the SCA service (Example: new initialContext(), nc.lookup(), and so on). In this case, the SCA service has to be deployed into an SCA runtime that is capable of supporting the EJB binding. Note that deployment of this service can result in the generation and deployment of a session bean, along with its home interface.



SCA service accessed as an EJB session bean

Service in the SCA composite file (default.composite)

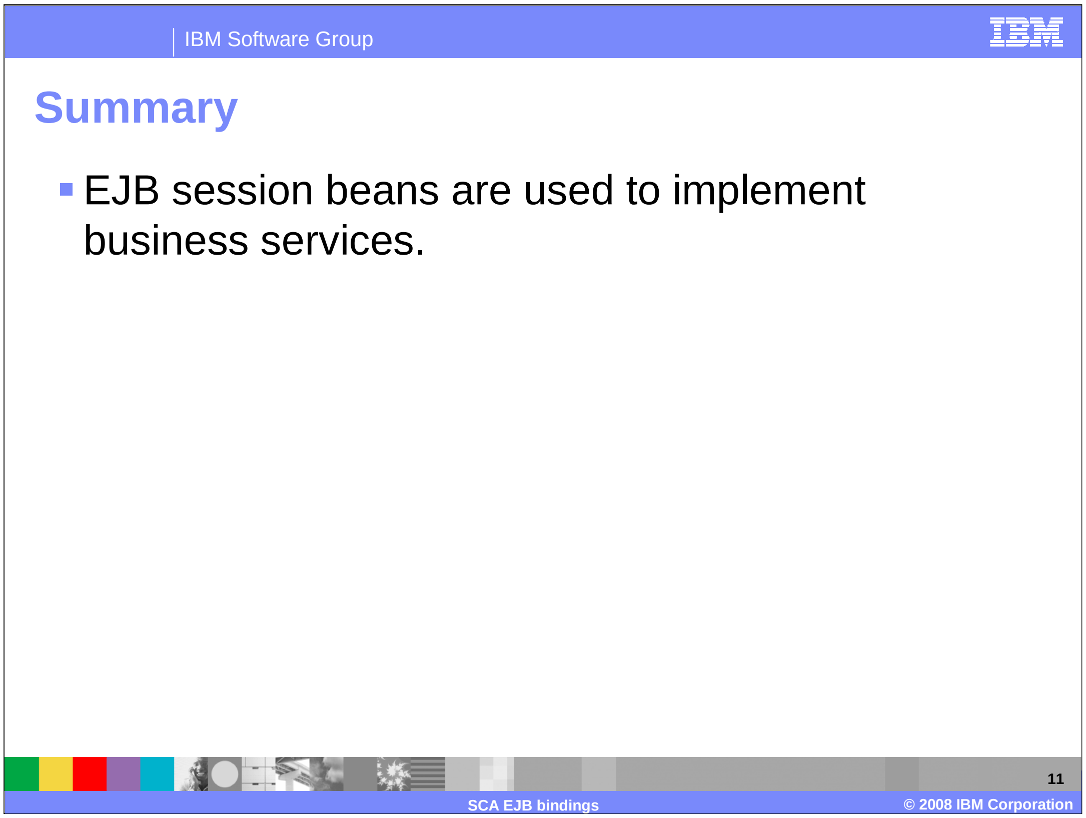
```
<service name=”CompanyInfo”>
  <interface.java interface=”com.app.jobbank.CompanyInfo”/>
              <binding.ejb ejb-version=”EJB2”/>
 <reference>CompanyInfoComponent/CompanyInfo</reference>
</service>
```

Client “lookup” code

```
(CompanyHome)initialContext.lookup("ejb/sca/ejbbinding/CompanyComponent/C
ompany");
```

Here is an example of how the service in the previous slide may be accessed as EJB session bean.

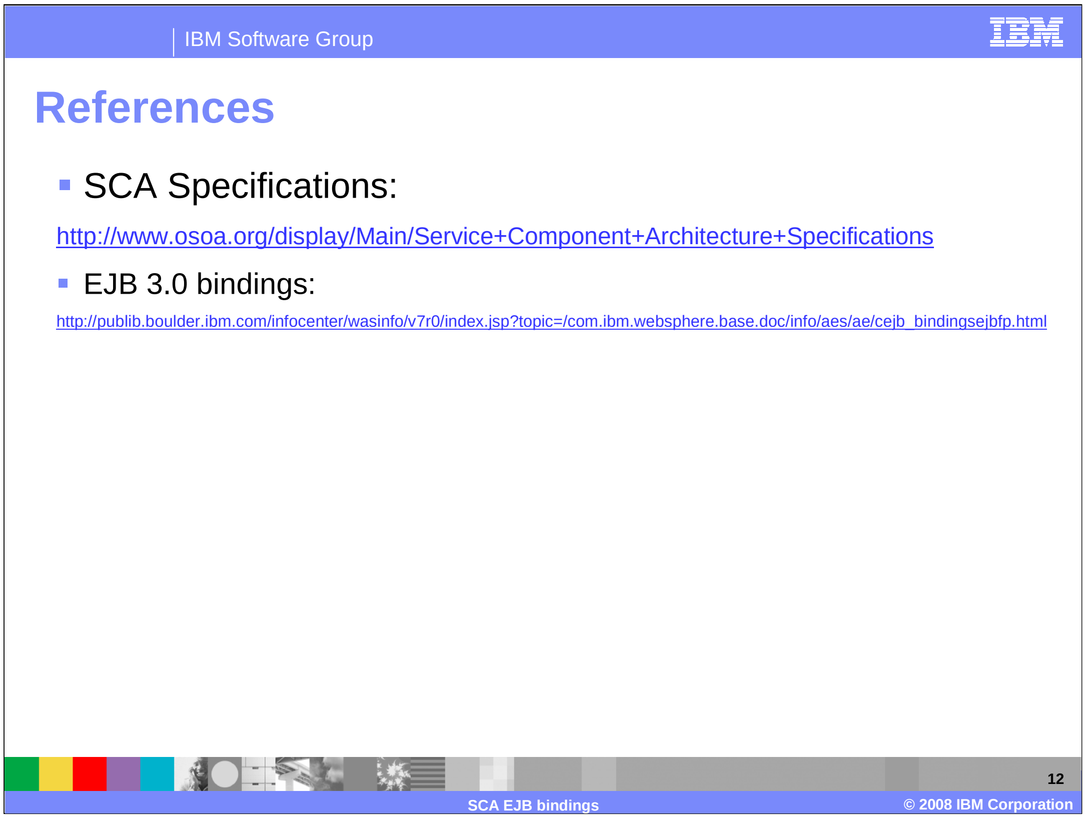
Since the client will use the standard Java EE programming model, the client needs to know the home interface of the SCA service. The service in the sca.composite file will do a “lookup” as shown.



Summary

- EJB session beans are used to implement business services.

EJB session beans are used to implement business services.



References

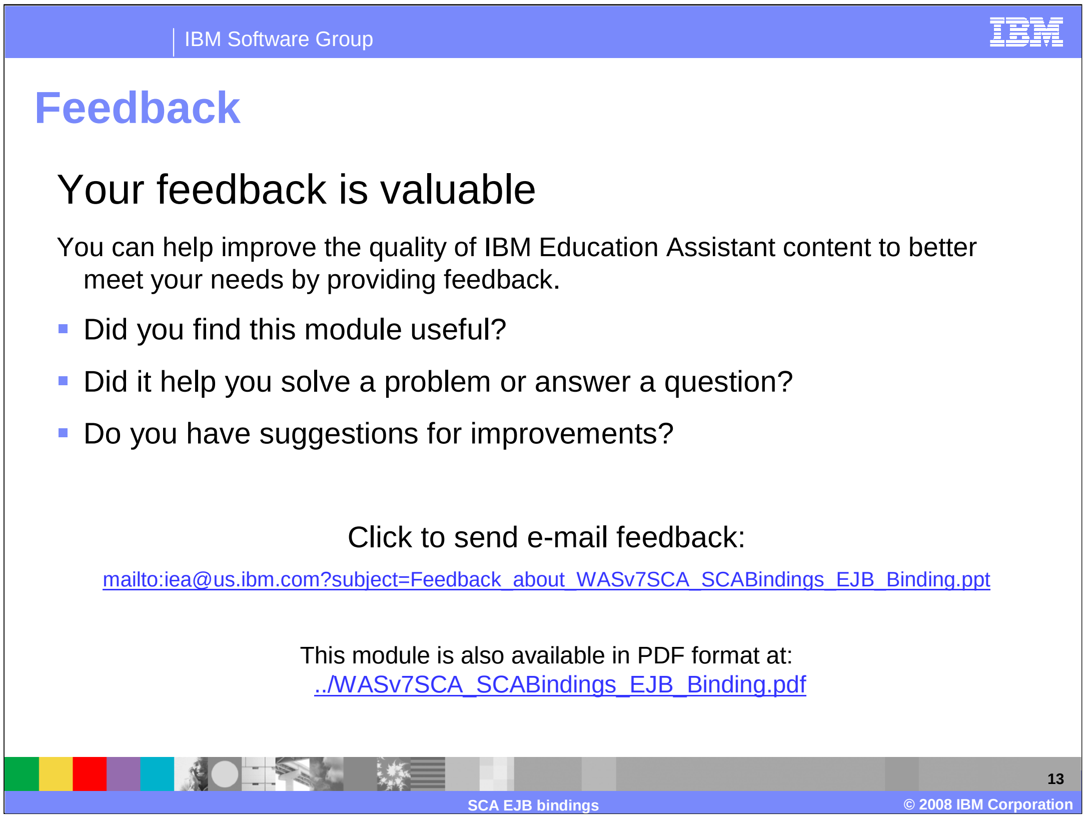
- SCA Specifications:

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

- EJB 3.0 bindings:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cejb_bindingsejbfp.html

The Service Component Architecture specifications and the information center article on EJB 3.0 bindings are available at the addresses shown here.



IBM Software Group

IBM

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_EJB_Binding.ppt

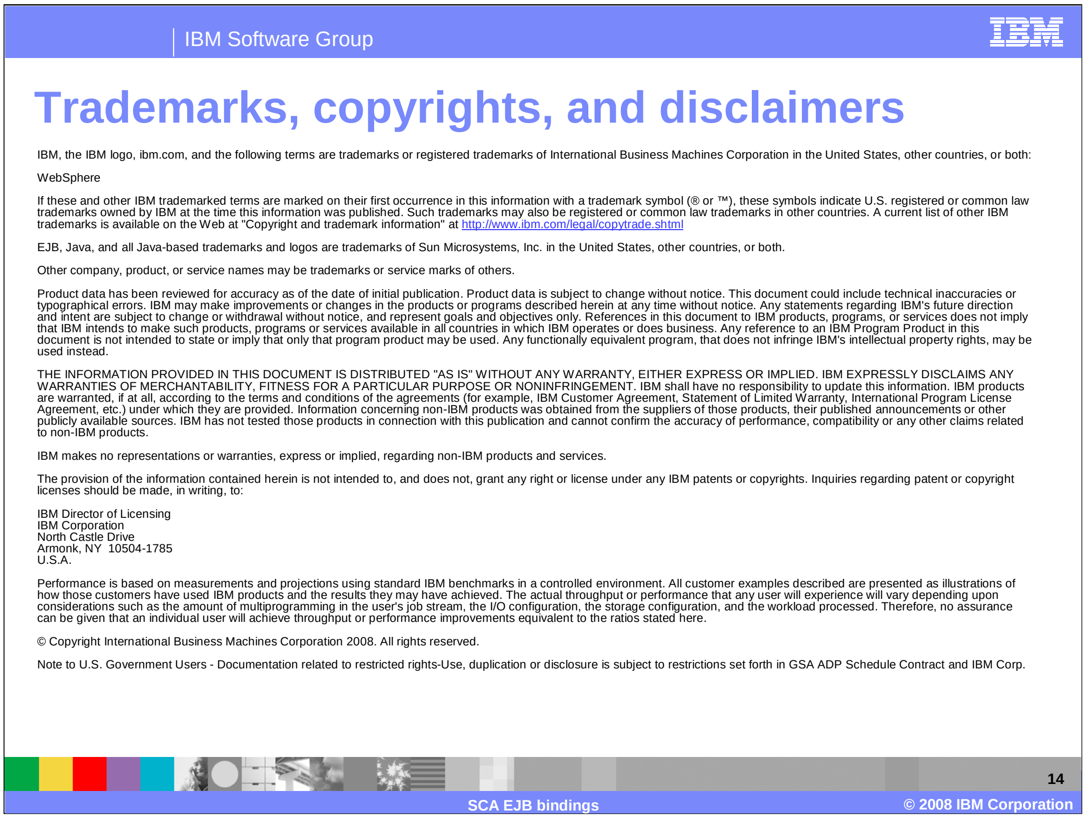
This module is also available in PDF format at:
../WASv7SCA_SCABindings_EJB_Binding.pdf

13

SCA EJB bindings

© 2008 IBM Corporation

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

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.

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.