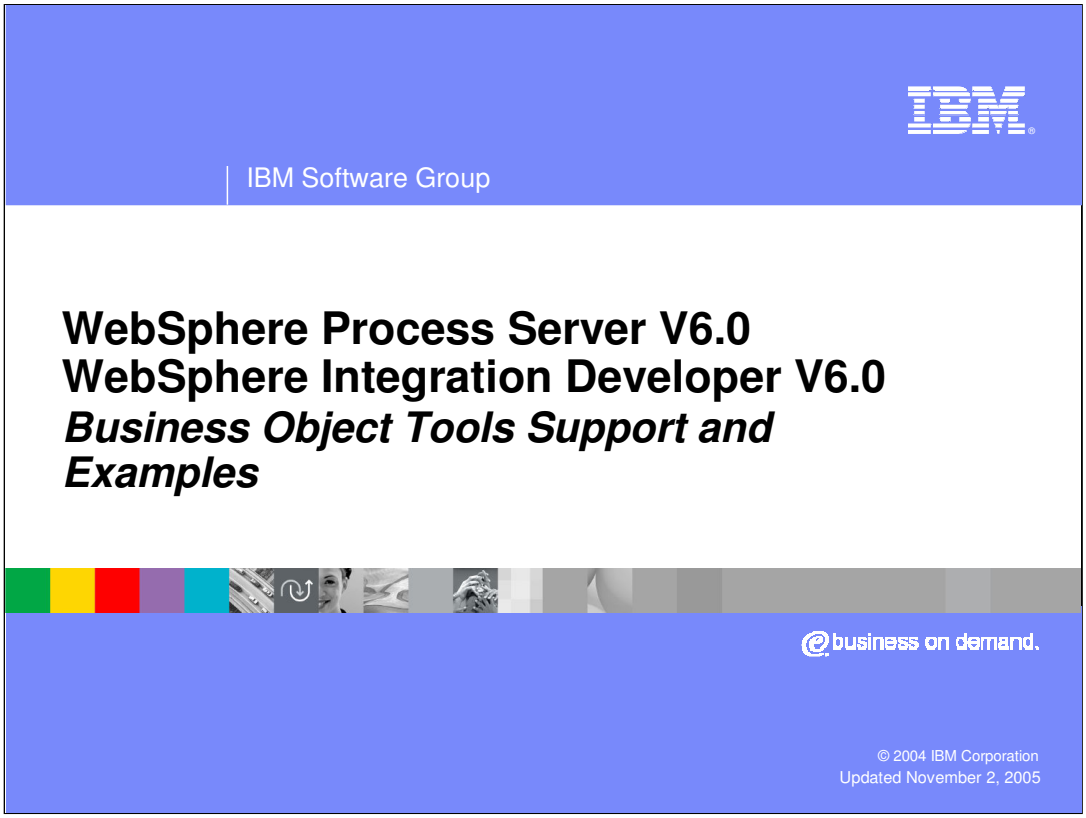
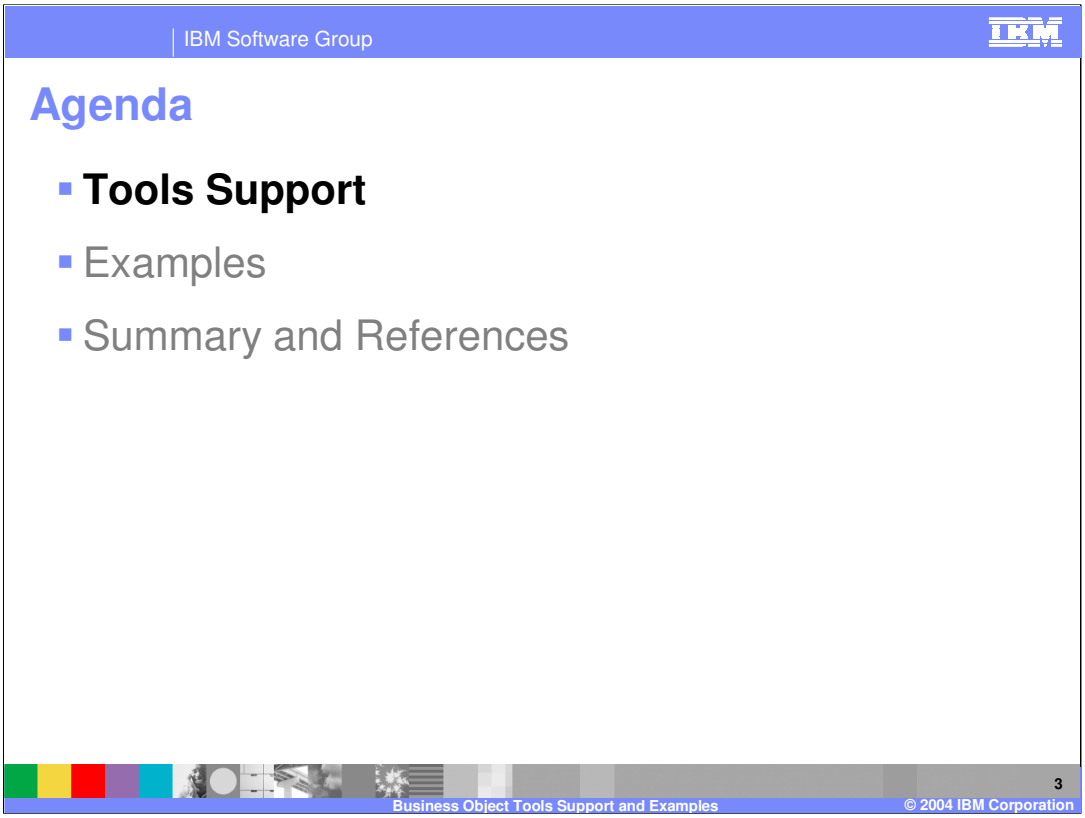


This presentation will provide an overview of business objects.

Goals

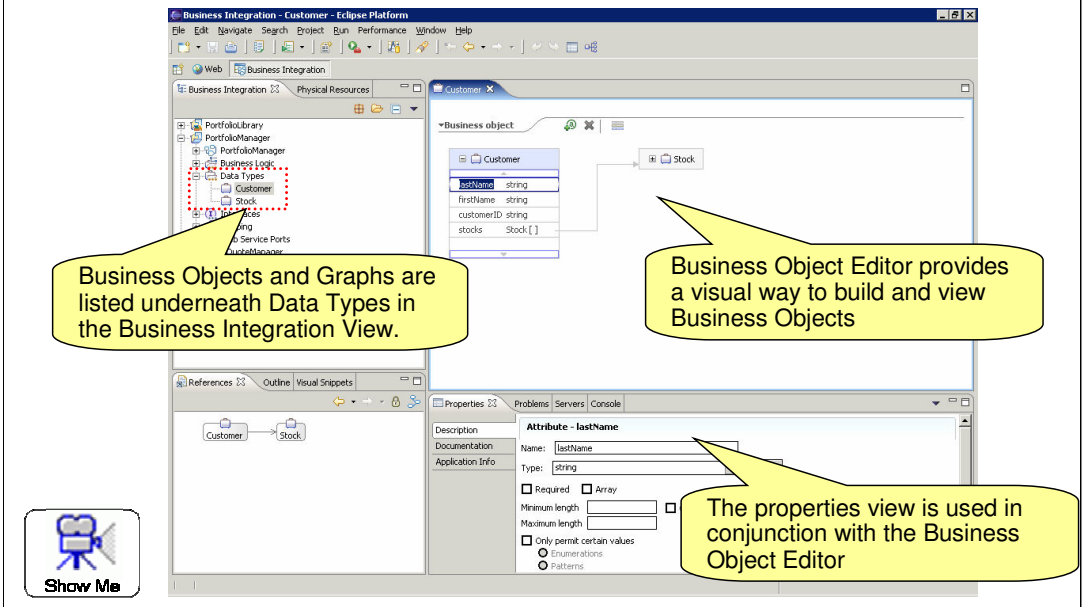
- Introduce tools support for defining Business Objects
- Provide a simple example of an SCA-based application that uses Business Objects

The goals of this presentation are to provide an introduction to the tools support for defining business objects using WebSphere Integration Developer V6.0. The presentation concludes with a simple example of an SCA-based application that uses business objects and provides an example of using the business object programming model.

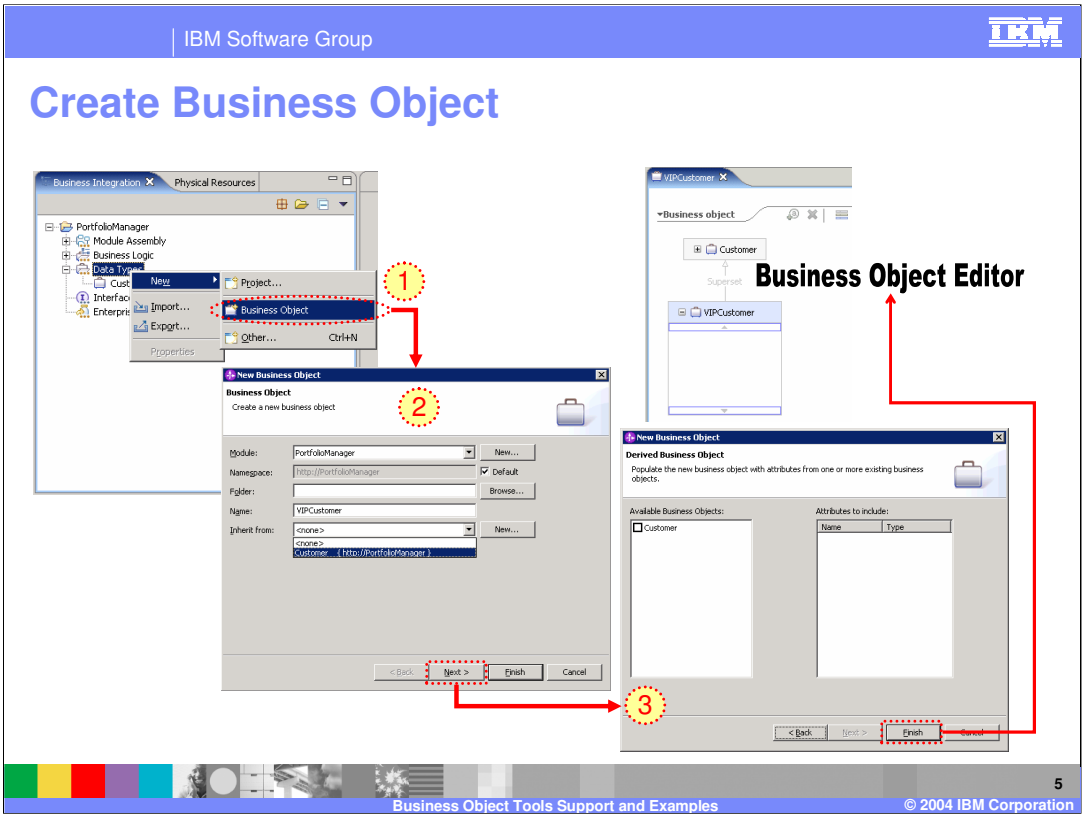


This section will provide an overview of the tools support for business objects.

Business Object Editor: Getting Started

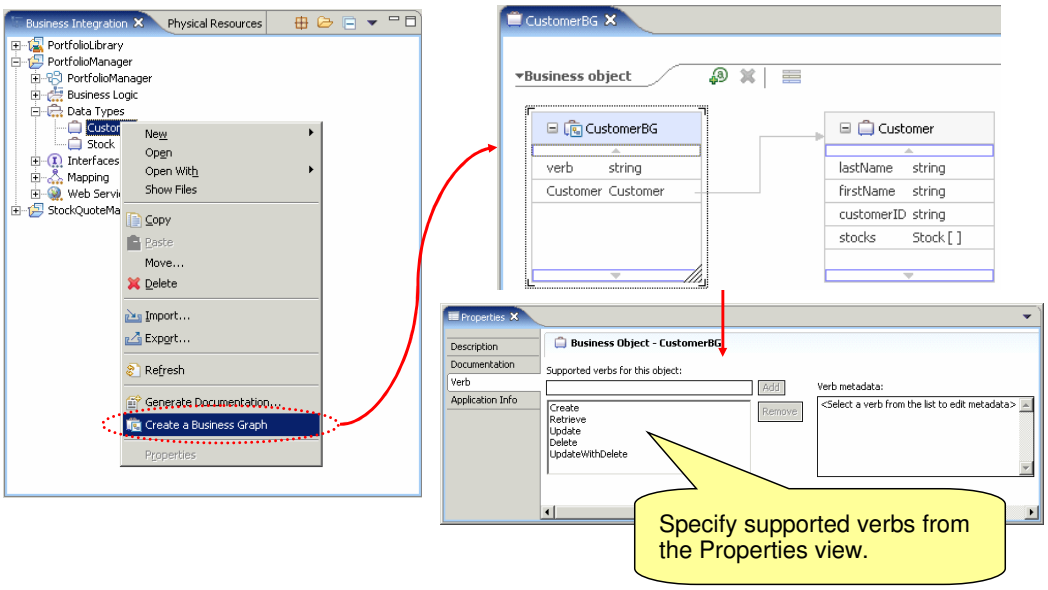


WebSphere Integration Developer V6.0 provides tools to help developers build and work with business object definitions. The workspace shown on this slides provides an overview of the tools available to work with business objects in WebSphere Integration Developer. Business objects and graphs that are included in a module or library project are listed in the business integration view underneath the Data Types category. The primary tool for editing and viewing a business object or business graph definition is the Business Object editor. In addition to this, the properties view is used in conjunction with the business object editor, and is also important in defining business objects and business graphs.



A new business object can be created using the New Business Object wizard. One way to launch this wizard is to right click on Data Types in the Business Integration view, and select New > Business object from the context menu. On the first panel of the wizard, you specify basic information about the new business object. This includes things such as the name and namespace (optional) for the business object. Also included on the first panel is the ability to select a parent business object to associate with the new business object. Selecting a business object in the inherit from field gives the new business object all of the attributes included with the parent business object plus any additional attributes that you add to the business object. On the second panel of the wizard you have the opportunity to select from the available business objects and select one or more attributes from that business object to include in the new one. This differs from the inheritance option because these attributes can be modified or changed while editing the new business object. In the case of inheritance, the parent attributes can not be removed from the new business object, and any changes to the parent object are automatically reflected in the child business object.

Create Business Graph



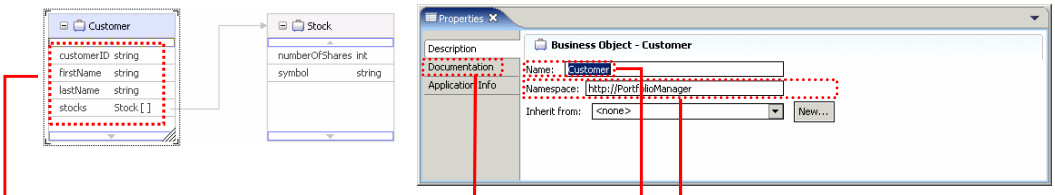
If you decide that you need the enhancements provided by a business graph, you can create a business graph definition by right clicking on the appropriate top level business object and selecting ‘Create a Business Graph’ from the context menu. The business object editor is used to build the business graph, and the properties view can be used to customize the verbs associated with the business graph.

Agenda

- Tools Support
- **Examples**
- Summary and References

This section will provide several examples of working with business objects.

Business Object: Example



Customer.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:bo="http://PortfolioManager"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://PortfolioManager">
  <xsd:include schemaLocation="Stock.xsd"/>
  <xsd:complexType name="Customer">
    <xsd:annotation>
      <xsd:documentation>This represents a Customer.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element minOccurs="0" name="lastName" type="xsd:string"/>
      <xsd:element minOccurs="0" name="firstName" type="xsd:string"/>
      <xsd:element minOccurs="0" name="customerID" type="xsd:string"/>
      <xsd:element minOccurs="0" name="stocks" type="bo:Stock" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```


Business objects in WebSphere Process Server V6.0 are modeled using XML schema. The example shown on this slide illustrates this by showing the schema definition for a customer business object built using the business object editor.

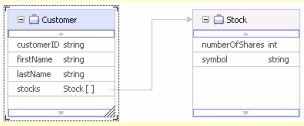
Business Object: Example (Continued)

```
ServiceManager serviceManager = new ServiceManager();
BOFactory bof =
            (BOFactory) serviceManager.locateService("com/ibm/websphere/bo/BOFactory");
DataObject customer = bof.create("http://PortfolioManager", "Customer");

customer.setString("firstName","John");
customer.setString("lastName","Doe");
customer.setString("customerID", "123-45-6789");

List l = new Vector();
DataObject stock = bof.create("http://PortfolioManager", "Stock");
stock.setString("symbol", "IBM");
stock.setInt("numberOfShares", 100);
l.add(stock);

stock = bof.create("http://PortfolioManager", "Stock");
stock.setString("symbol", "ABC");
stock.setInt("numberOfShares", 50);
l.add(stock);

customer.setList("stocks", l);
```

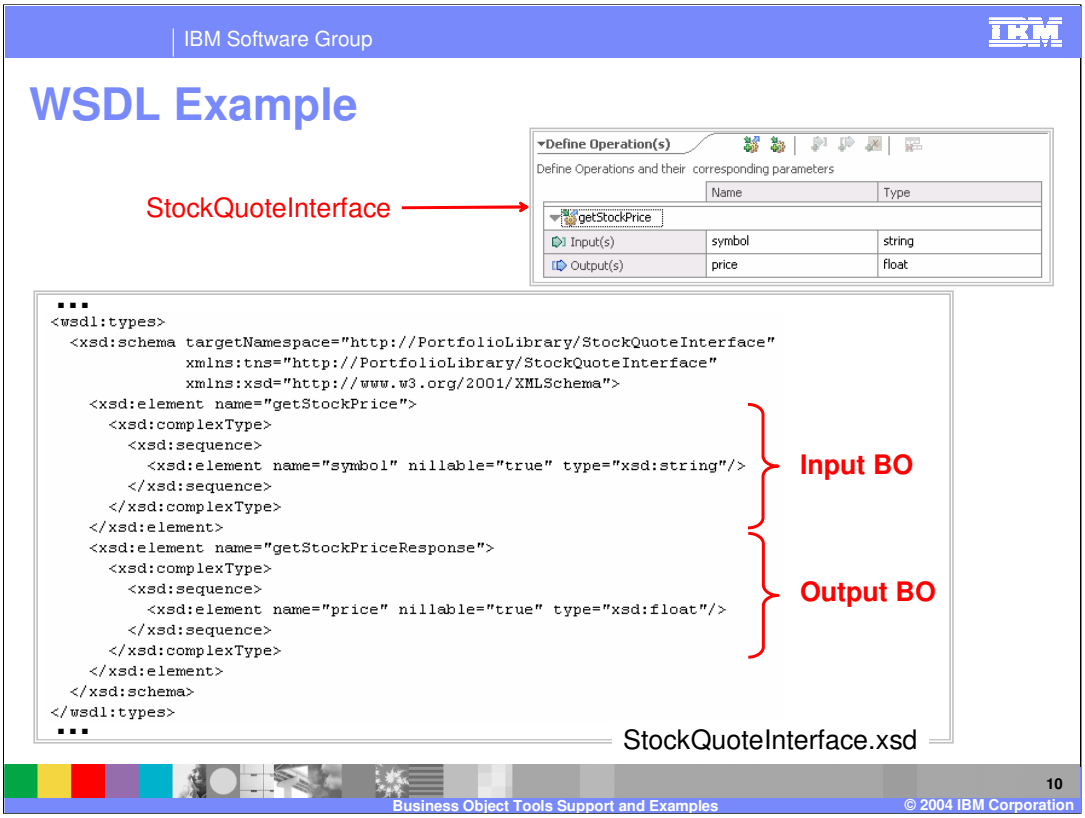
Create Customer BO

Set properties on Customer BO

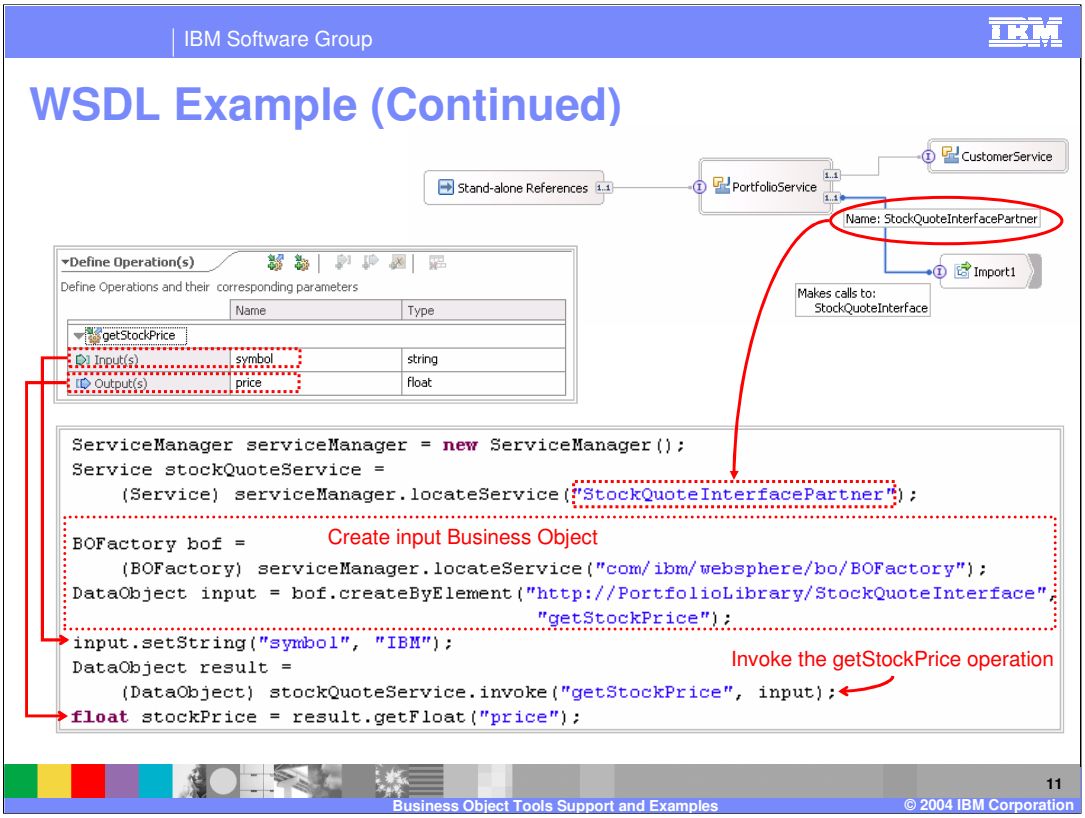
Create and set contained collection of Stock DataObjects



Once the schema definition for a business object is created, this definition will ultimately be used at runtime to exchange data between services within the WebSphere Process Server V6.0 runtime environment. The code snippet on this slide illustrates some of the fundamental client code that is needed to work with business objects. The first thing that is needed is to use the BOFactory API to instantiate the appropriate business object. At runtime, business objects are represented as commonj.sdo.DataObjects, and the create method of the BOFactory returns a DataObject. Once the business object has been created by the BOFactory, you are free to use the DataObject methods to get and set properties on the business object. The remaining part of the code snippet illustrates how to use the DataObject methods to set properties on the DataObject.



A common way business objects are used in WebSphere Process Server V6.0 is to represent the document literal message definitions for WSDL port type interfaces. This slide introduces a simple interface that is used on the following slide to demonstrate how to work with this type of business object from the perspective of the SCA client programming model.



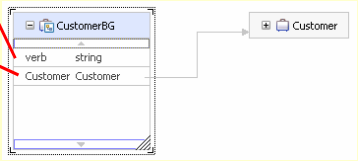
In this example, there is an interface (StockQuoteInterface) with an operation called getStockPrice. This operation takes as input a string that represents the stock symbol, and returns a float that represents the current price of the stock. In a simple application this interface is associated with an import component, and another service (called PortfolioService) references the imported service. The snippet of code shown at the bottom of the slide represents the code that would be needed in the PortfolioService implementation in order to invoke the getStockPrice functionality.

In this code example, after the stock quote service has been located using the service manager, the next step is to create the input data object that will be passed in when invoking the service. For this, you use the createByElement method on the BOFactory interface. As input into the createByElement, you use the namespace and name of the input message found in the WSDL definition file for the StockQuoteInterface. Note that the return from the invoke call is a data object.

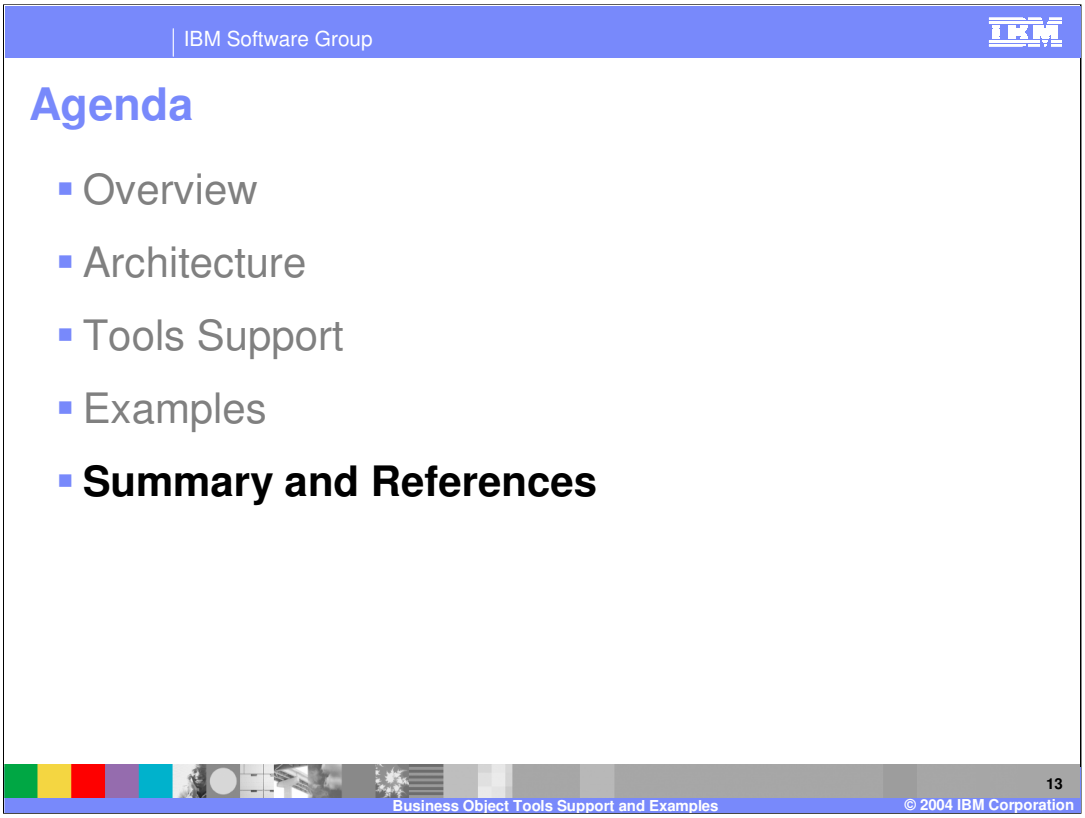
Business Graph: Example

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:bg="http://PortfolioManager" xmlns:refbo="http://PortfolioManager"
          xmlns:wbibg="http://www.ibm.com/xmlns/prod/websphere/bo/6.0.0"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://PortfolioManager">
<xsd:include schemaLocation="Customer.xsd"/>
<xsd:import namespace="http://www.ibm.com/xmlns/prod/websphere/bo/6.0.0"/>
<xsd:complexType name="CustomerBG">
   <xsd:complexContent>
    <xsd:extension base="wbibg:BusinessGraph">
     <xsd:sequence>
      <xsd:element maxOccurs="1" minOccurs="0" name="verb">
       <xsd:simpleType>
        <xsd:restriction base="xsd:string">
         <xsd:enumeration value="Create"/>
         <xsd:enumeration value="Update"/>
        </xsd:restriction>
       </xsd:simpleType>
      </xsd:element>
      <xsd:element name="Customer" type="bg:Customer"/>
     </xsd:sequence>
    </xsd:extension>
   </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

CustomerBG.xsd



This slide provides a simple example of an XML schema definition for a business graph that wraps a top level business object called customer.



This section will provide a summary of business objects.

Business Object Framework: Summary

- Important part of the SCA solution in WebSphere Process Server V6
  - Abstraction layer for data access
- Built upon SDO technology
  - Enhanced to provide additional capabilities
- Provides patterns and capabilities similar to the ICS business object

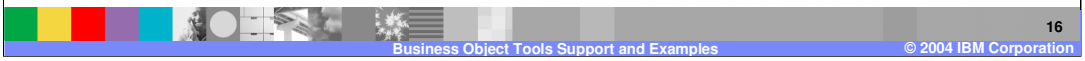
The business object framework is an important part of the SCA solution in WebSphere Process Server V6.0 because it provides an abstraction layer for data access. This framework is built upon SDO V1.0 technology, but several additional APIs are included to provide some additional capabilities not found in this version of SDO. Finally, the business object framework provides support for patterns and capabilities that are similar to the ICS business object.

References

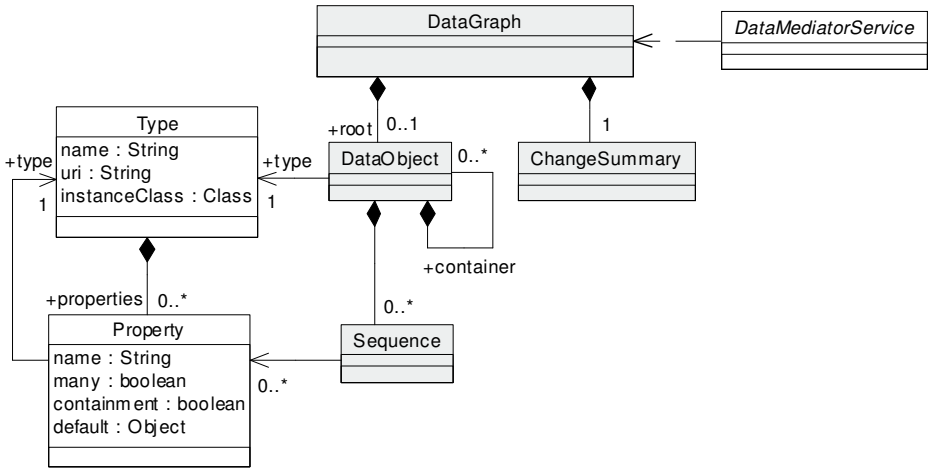
- XML Schema
  - http://www.w3.org/XML/Schema
- Service Data Objects (SDO)
  - http://www-128.ibm.com/developerworks/library/j-commonj-sdowmt/
  - http://www-106.ibm.com/developerworks/java/library/j-sdo/
- Eclipse Model Framework (EMF)
  - http://www.eclipse.org/emf/

Section

Appendix



Service Data Objects: Object Model



Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM(logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

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

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and 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 and/or changes in the product(s) and/or program(s) 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.

Information is provided "AS IS" without warranty of any kind. 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 (e.g., 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.