



IBM Software Group

# WebSphere® Process Integration

## *Introduction*



@business on demand.

© 2006 IBM Corporation  
Updated January 30, 2006

This presentation will provide an introduction to WebSphere Process Integration as of version 6.0.1, including WebSphere Process Server, WebSphere Enterprise Service Bus, WebSphere Integration Developer and WebSphere Adapters.

## Goals

- Lay Groundwork to Understand Process Integration
  - ▶ Vision
  - ▶ User Roles
  - ▶ A New Programming Model
  - ▶ What are the Components
  
- Understand Product Heritage



The goal of this presentation is to provide a basic understanding of WebSphere Process Integration, which will enable you to put into context the more detailed material provided as part of this education.

To accomplish this, the overall vision for WebSphere Process Integration will first be provided and the suite of products that fit into that vision identified. Because this education focuses on WebSphere Process Server, WebSphere Enterprise Service Bus, WebSphere Integration Developer and WebSphere Adapters, emphasis will be placed on where these products fit into the overall picture.

Understanding this requires identifying the various user roles that make up the end to end task flow involved in process integration, looking specifically at the integration developer role.

Next, the new programming model, which is specifically targeted to meet the requirements of the integration developer role will be examined.

Then the components provided as a part of process integration will be introduced. This will be a very brief overview as each of these components is addressed in detail in separate presentations.

Finally, product heritage of previous integration products which have contributed to the functionality that is provided by WebSphere Process Integration will be discussed.

## Service Oriented Architecture (SOA)

### The Key to Business Flexibility

***The flexibility to treat elements of business processes and the underlying IT infrastructure as standardized components (services) that can be reused and combined to address changing business priorities***

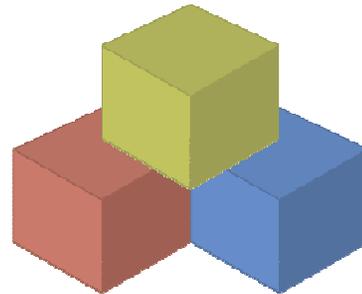
Services are the “building blocks” for business processes

Packaging business functions from applications in a simple and standardized way creates **services** that are available for use

Services are used directly to get the right information to the right people at the right time

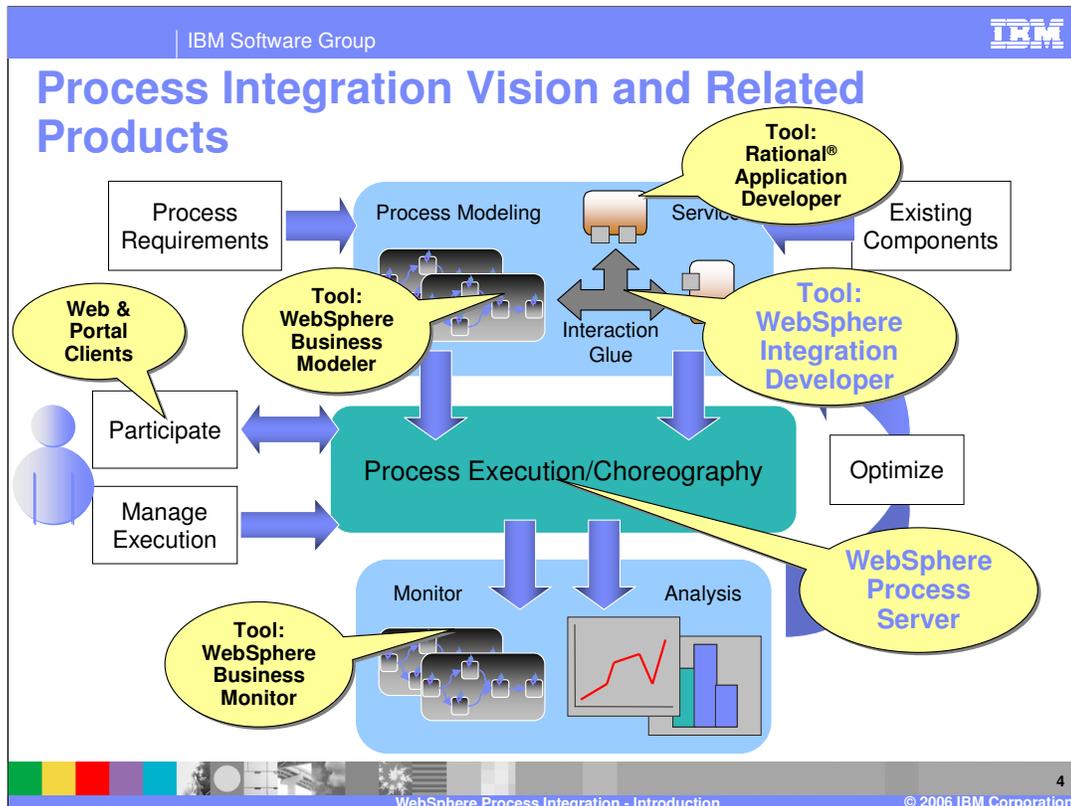
Services can be **reused and combined** to deploy composite applications to address new opportunities or changing business priorities.

Increasing use of **Web services** based on open standards complements existing services technology



Process integration is built on the concept of a Service Oriented Architecture. SOA has been a popular topic in recent years, but often the discussion of SOA has simply focused on the use of Web Services. The goal of the WebSphere Process Integration set of products is to make SOA a reality in a deeper way that goes well beyond just the utilization of Web Services.

The basis of Service Oriented Architecture starts with business processes. A service is a business task that provides some business function through a well defined interface. The use of the service is independent of the IT infrastructure needed to accomplish the task. Each service then becomes a building block that, when combined with other services, can provide an overall business process. The structuring of business processes in this way provides a flexible solution to real business problems, allowing the creation of new and modified processes easily. A key to this is the strong link between the use of services and the manner in which those services make use of the underlying IT infrastructure. The use of standards based technology, including Web Services, helps to make the SOA vision a reality.



This slide shows the vision for process integration and identifies the related products.

Starting at the top of the diagram, process modeling is used to take the requirements for a business process and develop a model that describes the business process, defining the steps and flow of work required to accomplish the business need. Once the model is completed, the services needed to accomplish the process will be known. In most cases there will be existing services that meet much of the need, but there could be new services that must be implemented using the development tools. The development tools are then used to assemble these existing and new services in such a way as to implement the process model.

The assembled business process can then be deployed into a runtime that supports both the business process execution environment and the packaging of business tasks as services. Human interaction will occur between those people who participate in various steps of the business process and the runtime environment in which it is executing. There will also be human interaction with the runtime environment by those who have responsibility to administer the business processes and the runtime environment.

While the business processes are running, data is captured that provides information about the processes. Both business related information and IT related information can be captured for monitoring. The data that is captured over some period of time can then be analyzed to determine if the business processes are performing as expected. This analysis can then be used to make improvements in the business model and business process implementations.

The products that fit into this vision for process integration are:

WebSphere Business Modeler provides process modeling capability. The output of the model can be used as a starting point to implement the business process and to establish how the business process will be monitored.

Rational Application Developer is used to build additional application function that has been identified through the process model.

The WebSphere Integration Developer is used to package application function as services and to compose new services through the assembly of other services. The output of the modeler is used by WebSphere Integration Developer for the implementation of the overall business process.

The WebSphere Process Server runtime environment is built on top of WebSphere Application Server. This provides a full J2EE application server on which the additional process integration and SOA capabilities are built.

The human interaction for both participants in the business process and administrators of the business process is done through browser based Web clients and Portal clients.

The WebSphere Business Monitor is used for monitoring and analysis of the business process execution.

WebSphere Integration Developer and WebSphere Process server are highlighted in this slide, as they are the focus of this education material.

## User Roles Defined

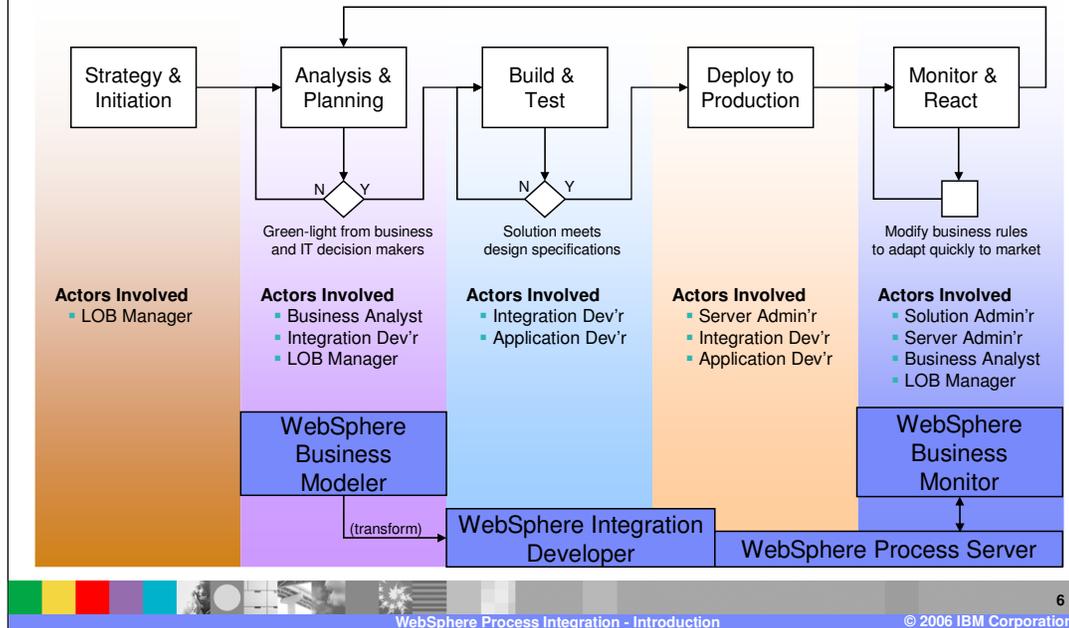
- A user role is *not* a real person
- A user role is “an abstract collection of needs, interests, expectations, behaviors and responsibilities characterizing a relationship between a class or kind of user and a system.”  
[Constantine & Lockwood, *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*, 1999]
- Individuals assume a user role
  - ▶ One person may assume many roles
  - ▶ Defined by activity and phase



To better understand process integration, it is necessary to have an understanding of user roles and how those roles fit into the overall end to end task flow. This slide introduces the definition of a user role.

A user role does not necessarily map to a specific person. The definition of a user role from Constantine and Lockwood, who were involved with usage-centered design, is that a user role is “an abstract collection of needs, interests, expectations, behaviors, and responsibilities characterizing a relationship between a class or kind of user and a system.” Individuals assume a user role when performing some particular task or activity. One particular person can assume many different roles at different times. The role they assume is determined by the activity they are involved in and the phase of the process they are involved with.

## Roles applied to an end-to-end taskflow:



This slide presents several of the user roles involved in the end to end task flow of business process integration.

The line of business manager is responsible for developing the business and the strategy that will be used to make the business successful. They define the products and services that the business provides and the strategy for how to market, sell and manage those products and services.

The line of business manager works with people in the Business Analyst role to define new products and services. The business analyst understands the business but also understands how to define and model the processes needed to implement the strategy defined by the line of business manager. The Integration Developer may also be involved in this stage. The Integration developer understands how to take a business model and implement it. The combination of the line of business manager, business analyst and integration developer working together will yield a business process model that addresses the business strategy and can be implemented. This analysis and planning activity is done using the WebSphere Business Modeler.

Once the model is complete it can be exported in a form that can be consumed by the WebSphere Integration Developer. In this phase the Integration Developer completes the implementation of the business processes according to the model. There could be a need for new application function to be developed in order to have all the services needed by the business process. A person assuming the Application Developer role will do this. The implementation will go through an iterative build and test cycle until it is ready to be deployed to production.

The Integration Developer and the Application Developer now work with the Server Administrator to deploy the business process to the production environment. The deployable business processes are exported from the WebSphere Integration Developer and installed into WebSphere Process Server. Once the Server Administrator is satisfied that all the requirements for production deployment have been met, the business processes will go in production.

Once the business process is running in production, the WebSphere Business Monitor is used to monitor the business process. The Server Administrator continues to be involved in ensuring the business processes are properly installed and configured. In addition, the Solution Administrator now becomes involved to monitor the business process and ensure that it is operating properly from a business perspective. Once some amount of historical data about the business process becomes available, the solution administrator will work with the business analyst and the line of business manager to analyze the data and determine if improvements can be made to the model and implementation of the business processes.

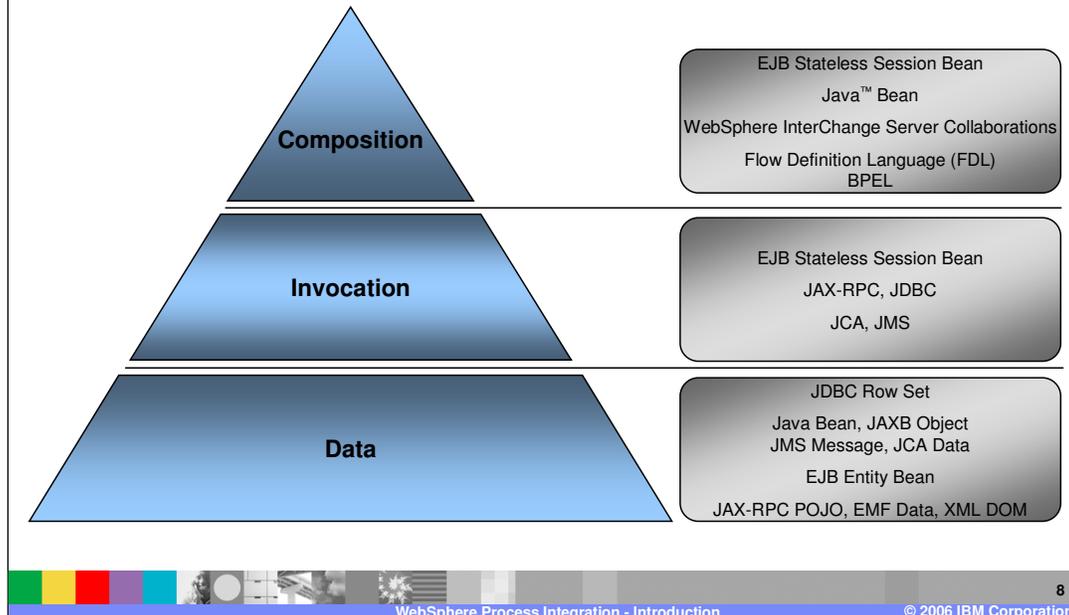
## Integration Developer Role

- The Integration Developer
  - ▶ The key target Role for WebSphere Integration Developer
- Integration Developer Responsibilities and Tasks
  - Creates interfaces and interface maps
  - Creates business objects, data maps and models data relationships
  - Creates business rules
  - Models & creates automated business processes (aka: workflows, process flows)
  - Creates points for human interaction
  - Models & creates service-oriented view of a solution
  - Builds solutions either top-down, bottom-up or meet-in-the-middle

The WebSphere Integration Developer tool is focused on meeting the needs of the person in the Integration Developer role. Since WebSphere Integration Developer is a primary focus of this education, it is important to understand more about the responsibilities of this particular role.

The integration developer is concerned with the definition of services, the data that flows between the services and how the services are put together to define the business processes. This involves defining interfaces and business objects. Assembling services into business processes requires both interface maps and data maps when interfaces or business objects do not exactly match. When data for the same real world entities are stored in disparate backend systems relationships must be defined that can be used to associate data for the same entity across the enterprise. Integration Developers also create business rules, define automated business processes and points of human interaction that are part of the business process. They also construct new services as required from application function provided by application developers. The Integration Developer maintains and works with a service oriented view of the business process solution.

## Programming Model – Today



WebSphere Process Integration introduces a new programming model. It is extremely important to understand this fundamental change in how applications are developed in order to understand the functionality provided in WebSphere Process Server and WebSphere Integration Developer. This slide depicts the programming model as it exists without WebSphere Process Integration.

Manipulation and handling of data is the core purpose and reason for all business processes.

There are currently various ways that data can be represented, including a JDBC row set from a relational database, a JMS message, JCA data coming in from an external application, a JAXB object, or an EJB entity bean.

Multiple ways of representing data require multiple ways of interacting with data. Invocation is the way in which a request is made from one piece of code to another. There are a variety of ways to invoke processes to obtain and work with data.

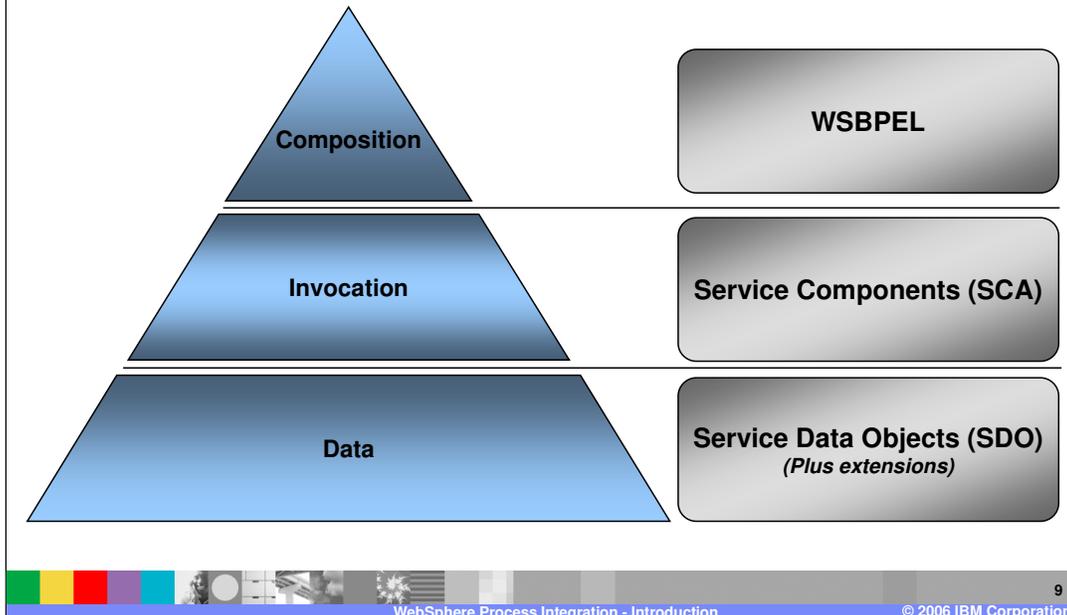
Current methods of invocation include EJB stateless session beans, JAX-RPC, JDBC for communicating with databases, JCA for adapters, and JMS messaging.

To build a solution you must be able to put together multiple invocations for data access and the processing that acts upon that data. Composition defines how to assemble a series of invocations that work on data to construct a complete process.

This can also be done in a variety of ways, including using Java with either a Java Bean or a stateless session bean, WebSphere InterChange Server collaborations, Flow Definition Language, and Business Process Execution Language.

All business process solutions are a composition of invocations that operate on data. With so many different mechanisms for doing this there is a need for many different kinds of skills to develop business processes. Also, the form that the data takes can dictate the form of invocation and composition which does not always yield the optimal approach.

## Programming Model – Simplification



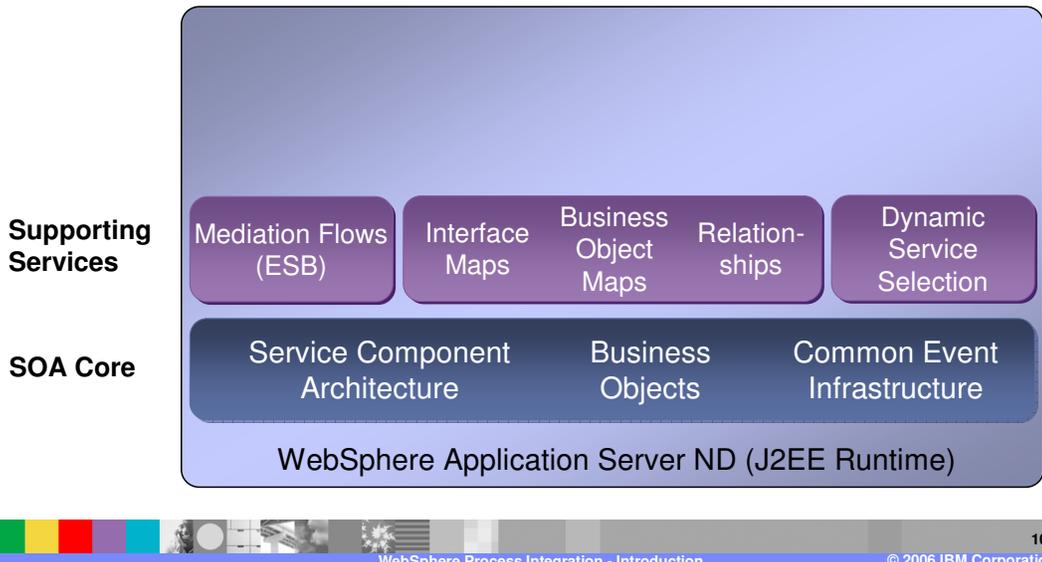
The programming model used with WebSphere Process Integration also involves building business processes by composing a series of invocations that act on data. However, with WebSphere Process Integration there is a simplified style in which this is done.

Data is defined using Service Data Objects (SDO). SDO provides an abstraction that can be used over various types of data, providing a common mechanism for accessing data. SDO are extended to provide additional functionality needed for process integration scenarios and are referred to as Business Objects.

Invocation is done using Service Component Architecture. SCA provides a standardized way to define and invoke services. The services are associated with different kinds of bindings so that the underlying invocations can be done with various invocation models, such as JMS or Web Services.

Composition is done using Business Process Execution Language (BPEL), which is used to define the overall business process. When the business process accesses data, it does so by making calls to SCA services passing Business Objects.

## Components – WebSphere Process Server V6.0.1 and WebSphere Integration Developer V6.0.1



This slide provides an overview of the components that make up the WebSphere Process Server version 6.0.1 and are enabled by the WebSphere Integration Developer version 6.0.1.

The WebSphere Process Server is built on WebSphere Application Server version 6.0.2. This provides a robust J2EE application server runtime with capabilities that the process server implementation can exploit, such as JMS messaging and enterprise beans. It can also make use of the application server qualities of service such as transactions, security and clustering. Overall, this provides a well proven runtime environment for WebSphere Process Server.

The Service Oriented Architecture (SOA) Core is the foundation in WebSphere Process Server. The main components of the SOA Core are the Service Component Architecture (SCA), Business Objects (BOs) and the Common Event Infrastructure (CEI).

SCA is the uniform programming and invocation model for business services that publish or operate on business data. This is one of the key components of the new programming model discussed on the previous slide.

Business Objects (BOs) represent the data that is passed within that framework. Business objects are extensions to Service Data Objects (SDOs), which carry additional information needed for some integration scenarios. SDOs in the form of Business Objects are another of the key components of the new programming model.

The Common Event Infrastructure (CEI) provides the foundation architecture for the management and handling of events produced by business processes. This is essential for enabling the monitoring of business processes with products such as the WebSphere Business Monitor.

On top of the SOA Core are a set of Supporting Services which provide for transformation and routing capabilities which are required by integration scenarios built using Service Component Architecture and Business Objects. The first of these supporting services are Mediation Flows which enable Enterprise Service Bus (ESB) capabilities, essentially enabling a loose coupling between service requestors and service providers. Mediation Flows enable the converting of protocols, the transformation of message content and formats and dynamic routing decisions to occur between a service requestor and service provider.

The next set of supporting services are Interface Maps, Business Object Maps and Relationships. This are shown together because of the close relationship between these supporting services.

Interface Maps are used to enable an SCA component making use of a particular interface to make calls to another SCA component that provides a semantically similar but syntactically different interface.

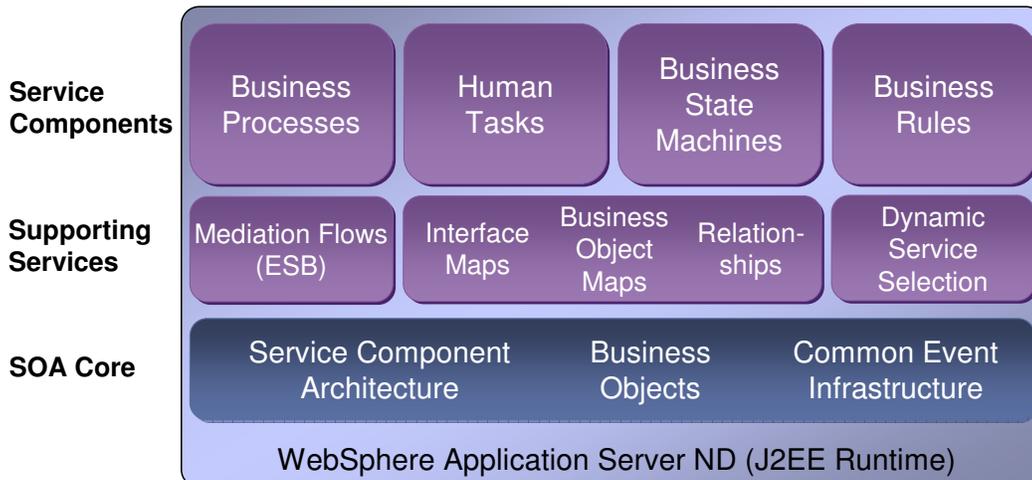
Business Object Maps enable the transformation of business data between fields of Business Objects representing the same business entity but of differing types. Business Object Maps are called from within Interface Maps to convert the Business Objects which are the inputs and outputs defined for the operations of the interfaces being mapped.

Relationships enable the correlation and synchronization of data representing the same business entity stored across multiple backend systems. Relationships are defined within the context of Business Object Maps.

The use of this set of supporting services is most commonly used in conjunction with the use of adapters, which represent backend systems that are being integrated through the capabilities of the WebSphere Process Server.

Dynamic Service Selection provides for a dynamic invocation of a target component based on a date and time criteria.

## Components – WebSphere Process Server V6.0.1 and WebSphere Integration Developer V6.0.1



Business Processes are a fundamental part of the programming model, providing the composition aspect of the programming model. In WebSphere Process Server, the business processes are defined using BPEL. Business processes provide an implementation of a process model that describes the logical order in which the different activities of the process are being executed, making calls out to the individual SCA services that implement the specific activities. As a result, a business process is the set of business-related activities, rules and conditions that are invoked in a defined sequence to achieve a business goal.

Human Tasks are enabled by the Human Task Manager and provide the human task capabilities for WebSphere Process Server.

Human Tasks allow people to participate in a business process in a machine-to-human scenario, a human-to-machine scenario and in a human-to-human scenario.

In the machine-to-human scenario an automated process creates tasks for people who participate in the execution of a business process, whereas the human-to-machine scenario allows a person to create a task that is executed by an automated service. The human-to-human scenario allows a task to be created by a person for another person.

Human tasks can be integrated directly into the BPEL for a business process or can be packaged as an SCA component for use by any client that can invoke an SCA component.

Business State Machines are another way of modeling a business process. There are some processes that are highly event driven and are well suited to being thought of in terms of a state transition diagram. For example, a business process for order processing has to deal with the fact that an order can be canceled at any time during the order process up until the order is actually shipped. It can be difficult to model these kinds of processes in a business process. The Business State Machine component allows you to model the business process using similar constructs as UML 2.0 state machine support, and then generates BPEL for the implementation.

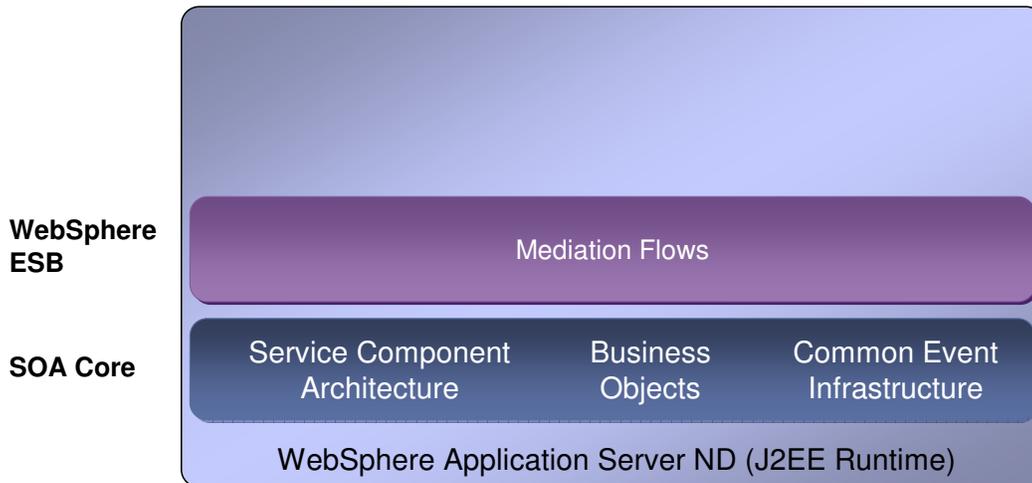
Business rules are a means of implementing and enforcing business policy through externalization of business function.

Externalization enables the business rules to be managed independently from other aspects of an application. This independence allows for dynamic updating capabilities of the business rules to provide for a more agile business. There are two styles of business rules, if then rulesets and decision tables.

A Web client is provided where the parameters of business rules can be changed by a business user using a natural language specification of these rules rather than requiring an application developer or integration developer to change the application.

So you can see that the WebSphere Process Server version 6.0.1 provides a complete and robust set of capabilities that can meet the needs of any integration scenario. The WebSphere Integration Developer version 6.0.1 provides the high level abstractions in the tools that enable the Integration Developer to construct and define business process integration applications.

## WebSphere Enterprise Service Bus V6.0.1



The WebSphere Enterprise Service Bus version 6.0.1 is a separate product from the WebSphere Process Server version 6.0.1. Its primary function is to provide mediation flow capabilities for those environments where enterprise service bus functionality is needed but the other integration services provided by WebSphere Process Server are not needed. The base product stack is identical to WebSphere Process Server, being built upon WebSphere Application Server with all of its J2EE and quality of service capacities and the SOA core providing the Service Component Architecture, Business Object and Common Event Infrastructure functionality. Typically you would expect to see the WebSphere Enterprise Service Bus used in a Service Oriented Architecture based infrastructure to enable loose coupling between service requestors and providers, and thus enhancing the ability to rapidly change and introduce new business services into the environment.

## Adapters in Process Integration

- Adapters provide
  - ▶ EIS integration
  - ▶ Service oriented approach
  - ▶ Encapsulation of functions and events as Business Objects.
  - ▶ A consistent framework for access to back-end systems
  - ▶ Consistent configuration, deployment, and administration
- Adapters can be categorized as:
  - ▶ Application (PeopleSoft, SAP, Siebel)
  - ▶ Technology (JDBC, Flat File)
- Two major types
  - ▶ WBI Adapters (Existing adapters as used with InterChange Server)
  - ▶ WebSphere Adapters (New adapter architecture based on JCA 1.5)



Adapters are used within process integration for doing enterprise integration of various applications and back end systems that are external to the WebSphere Process Server or WebSphere Enterprise Service Bus.

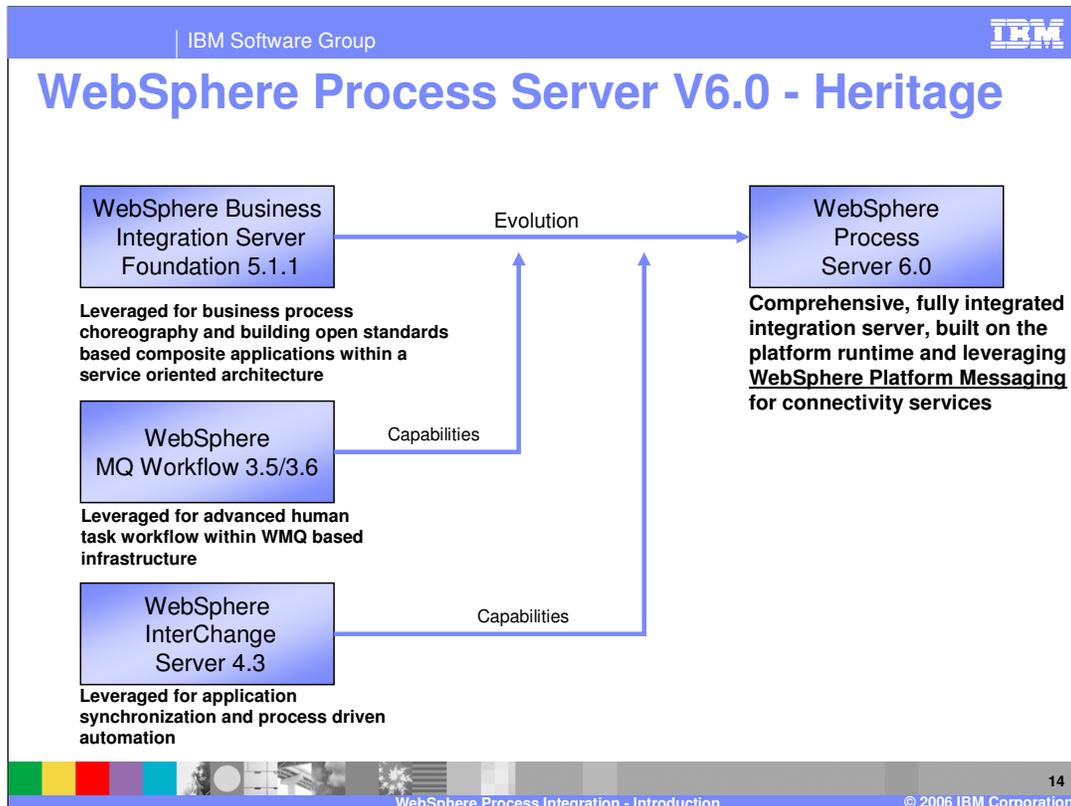
Adapters are integrated into the system using a service oriented approach and are fully compatible with the service component architecture.

The functionality of an enterprise system is encapsulated by an adapter and business objects are used to pass data back and forth between the process server or service bus and the enterprise system. The use of business objects and service component architecture allow adapters to fit perfectly into the new programming model.

The use of adapters provides a consistent framework for accessing backend systems and a consistent way to configure, deploy, and administer the adapters.

Adapters can be categorized as application adapters or technology adapters. Application adapters are used for connecting to specific applications such as PeopleSoft, SAP or Siebel whereas technology adapters are used for using a specific technology for interfacing with a backend system. For example, a JDBC adapter could be used for interacting with DB2® or with Oracle databases.

In addition to how adapters can be categorized, there are two major types or styles of adapters that can be used with the WebSphere Process Server or WebSphere Enterprise Service Bus. The WBI Adapters are the existing adapters that are used today with the WebSphere Interchange Server and other brokers. The WebSphere Adapters are new as part of WebSphere Process Integration and are compliant with the Java Connector Architecture (JCA) 1.5 specification.



WebSphere Process Server V6.0 contains technology that is a direct result of IBM's rich heritage in process integration and messaging technology. WebSphere Process Server is based on the collective customer and development experience derived from WebSphere Business Integration Server Foundation, WebSphere MQ Workflow and WebSphere InterChange Server.

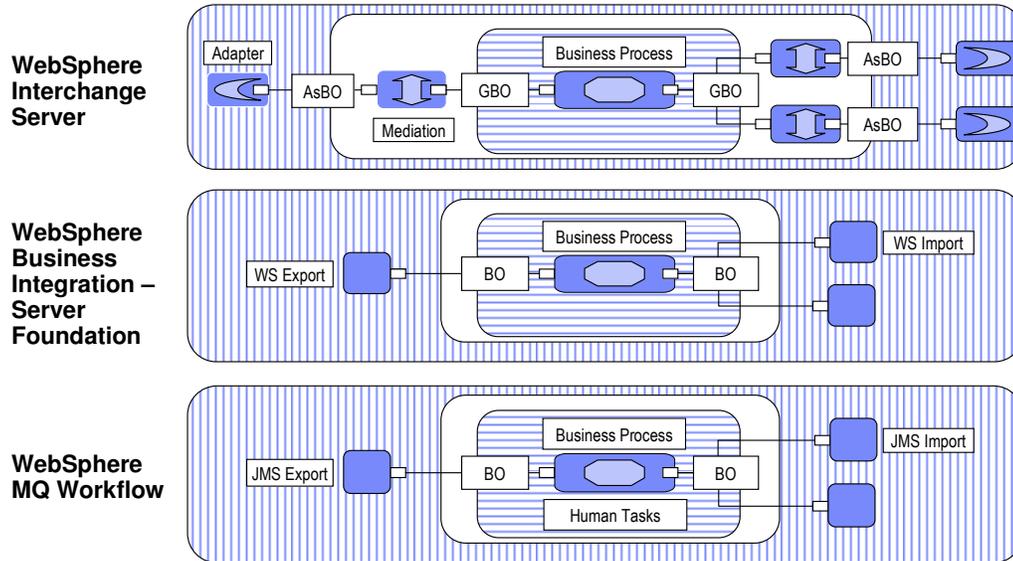
WebSphere Process Server is a direct evolution from WebSphere Business Integration Server Foundation 5.1.1. Just as WebSphere Business Integration Server Foundation was built on the J2EE application server environment provided by WebSphere Application Server V5, WebSphere Process Server is also built on J2EE application server technology provided by WebSphere Application Server V6. The capabilities of defining business processes with BPEL and making use of open standards such as Web Services has been carried forward.

In addition, the capabilities from MQ Workflow V3.5 and V3.6 have been designed into WebSphere Process Server. The integration of human task workflows and a strong messaging infrastructure have been built into the WebSphere Process Server.

Finally, the capabilities of WebSphere Interchange Server V4.3 have been designed into WebSphere Process Server as well. The ability to integrate disparate back end systems using adapter technology and providing a powerful infrastructure for application data synchronization along with process automation have also been built into the WebSphere Process Server.

The result is that WebSphere Process Server will be the platform for moving into the future in all areas of business process integration.

## Heritage Product Use Cases In WebSphere Process Server



15

WebSphere Process Integration - Introduction

© 2006 IBM Corporation

This slide depicts how the typical functionality from the three heritage products is provided using the WebSphere Process Server.

On the top is a typical WebSphere Interchange Server solution implemented in WebSphere Process Server. On the left, the adapter provides the source Application Specific Business Object (ASBO) from an external application. Using the sophisticated transformation features of WebSphere Process Server the ASBO is converted to a Generic Business Object (GBO). The GBO is then passed into a BPEL business process for further enhancement or action. As a final step, the GBO can be converted back to an ASBO for output to target systems using other adapters.

In the middle is a typical WebSphere Business Integration Server Foundation solution implemented in WebSphere Process Server. On the left is a Web Services SCA export which exposes the SCA service as a standard Web Service. The Business Process is a BPEL solution that integrates partner links exposed as SCA references and implemented as Web Services available using JMS SCA imports.

On the bottom is a typical WebSphere MQ Workflow solution implemented in WebSphere Process Server. On the left is a JMS SCA export receiving a message and interacting with a BPEL process. This process can support multiple types of component interactions including the Human Task Manager to support traditional workflow interactions. The Human Task Manager provides all the services to interact with the user and manage that interaction. Once the process is complete, the output could be sent using JMS through a JMS SCA import.

## Summary

- Built an understanding of WebSphere Process Integration
- Looked at the new programming model
- Surveyed the components
- Looked at the product heritage

The intention of this presentation is to provide a basic understanding of WebSphere Process Integration. The new programming model using system component architecture (SCA) and business objects was discussed as a basis for understanding this set of products. Components provided by the tools and the runtime were also examined, as well as heritage products and how they fit into the new product set.

## 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
AIK	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 NON-INFRINGEMENT. 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.

© Copyright International Business Machines Corporation 2005,2006. 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.