



IBM Software Group

WebSphere® Process Server V6.0.2 WebSphere Integration Developer V6.0.2

WebSphere Process Server Overview



@business on demand.

© 2007 IBM Corporation
Updated February 17, 2007

This presentation provides an overview of the WebSphere Process Server version 6.0.2.

Goals

- Goal of this presentation
 - ▶ Introduce a high level architectural view of WebSphere Process Server
- You will learn about:
 - ▶ The new programming model introduced by the server
 - ▶ The components comprising the server architecture
 - ▶ The WebSphere Process Server product heritage
- Helpful prior knowledge
 - ▶ IBM SOA Foundation - Business Process Management

The goal of the presentation is to provide you with an understanding of the high level architecture of WebSphere Process Server version 6. This is done by first explaining the new programming model that WebSphere Process Server supports, followed by an overview of the various components that provide WebSphere Process Server's capabilities. Finally you will see the predecessor products and how they relate to version 6 of WebSphere Process Server. To understand how WebSphere Process Server fits into the larger picture, it is recommended that you have already reviewed the Business Process Management Introduction presentation available as part of this series of presentations.

Background

- IBM SOA Foundation defines:
 - ▶ SOA life cycle, reference architecture and scenarios
 - ▶ Business Process Management scenario
 - WebSphere Process Server identified as the runtime environment
- WebSphere Process Server
 - ▶ Introduces a uniform programming model
 - ▶ Provides a standards based process integration platform
- WebSphere Integration Developer provides the development tools

The IBM SOA Foundation provides a definition of Service Oriented Architecture as viewed by IBM. It defines a SOA life cycle, a SOA reference architecture and a set of SOA scenarios that describe typical SOA implementations along with a list of IBM products that support those scenarios. One of the scenarios is the Business Process Management scenario, which is closely aligned with SOA life cycle and identifies WebSphere Process Server as the product associated with the deploy stage of the life cycle, providing a process integration runtime environment. The WebSphere Process Server process integration platform provides a uniform programming model for representation of data, invocation of services and structure of composite applications, all based on existing or emerging standards. The development of composite applications for WebSphere Process Server is done using the tools provided by WebSphere Integration Developer. Therefore, as the programming model and server components are described, they reflect the capabilities for building applications enabled by WebSphere Integration Developer.

Background

- Terms encompassing WebSphere Process Server
 - ▶ WebSphere Business Integration
 - ▶ WebSphere Process Integration
 - ▶ Business Process Management
- Usage and scope overlapping
 - ▶ There are subtle differences



WebSphere Process Server is associated with the terms WebSphere Business Integration, WebSphere Process Integration and Business Process Management. All three have overlapping meanings but there are some subtle differences.

WebSphere Business Integration generally tends to be used to encompass integration technologies and products that existed before the introduction of WebSphere Process Server version 6. However, it is still sometimes used in a context that includes WebSphere Process Server. It has a broad scope, encompassing a wide variety of integration technologies that may or may not be part of a SOA environment.

WebSphere Process Integration tends to be used to encompass integration technologies and products that are associated with WebSphere Process Server and the life cycle within which WebSphere Process Server participates. The scope is really all products and technologies that participate in some way to enable integration in the support of business processes, particularly within SOA environments.

Business Process Management is a term used throughout the IT industry related to the definition, optimization and improvement of business processes. This applies not only to SOA environments but also to non-SOA environments. However, it is given a much more specific meaning when viewed in terms of the IBM SOA Foundation Business Process Management scenario. Used in this sense, it identifies specific products within a life cycle model and reference architecture. It is therefore a somewhat more focused term and narrower in scope than WebSphere Process Integration.

Section

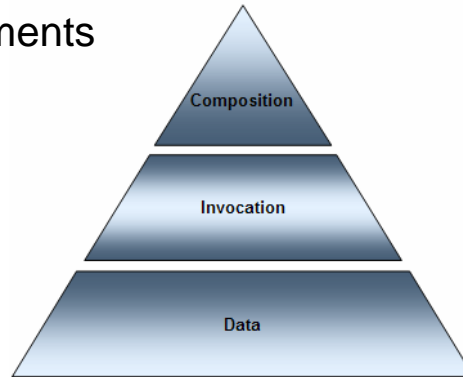
Programming Model

This section introduces the programming model supported by WebSphere Process Server.

Introduction to the programming model

- Programming model elements

- ▶ Data
- ▶ Invocation
- ▶ Composition



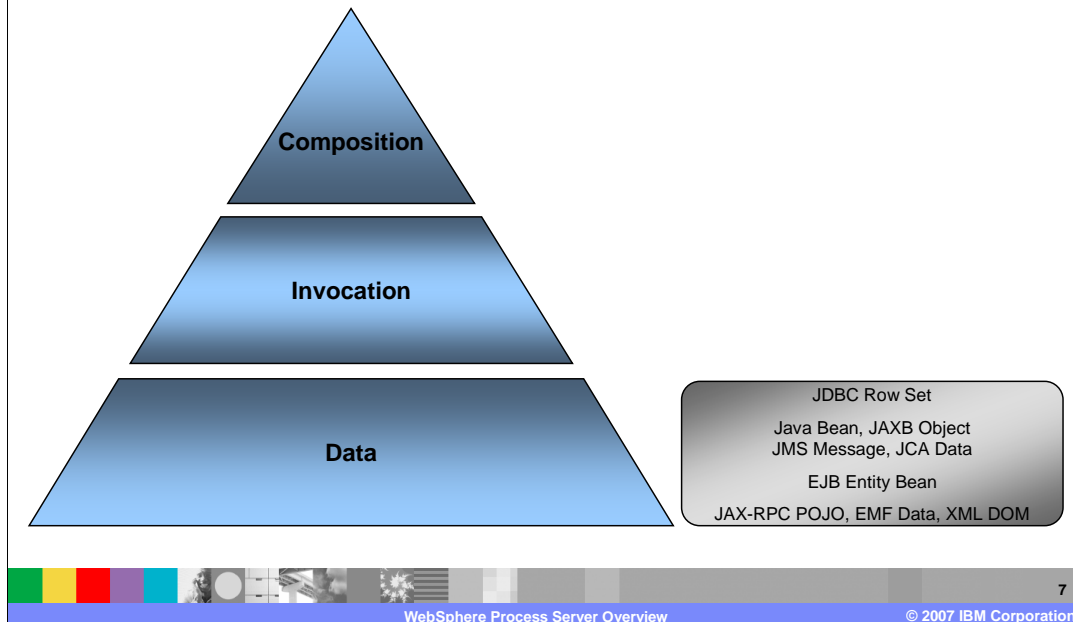
- Complex without WebSphere Process Server
- WebSphere Process Server introduces a new and simplified programming model

Business processes fundamentally consist of accessing, manipulating and moving data required by an enterprise to engage in business. The right data must be in the right place at the right time for people and automated processes to perform their assigned tasks.

Data can be thought of as something that is stored with some defined representation or encoding on some sort of physical device, whether an external storage device or internal memory. In order to access, move or manipulate the data, an invocation mechanism based on some programming construct within a runtime environment must be used. A composition mechanism must then be used to define the right set of invocations acting on the right set of data in order to perform a specific business process.

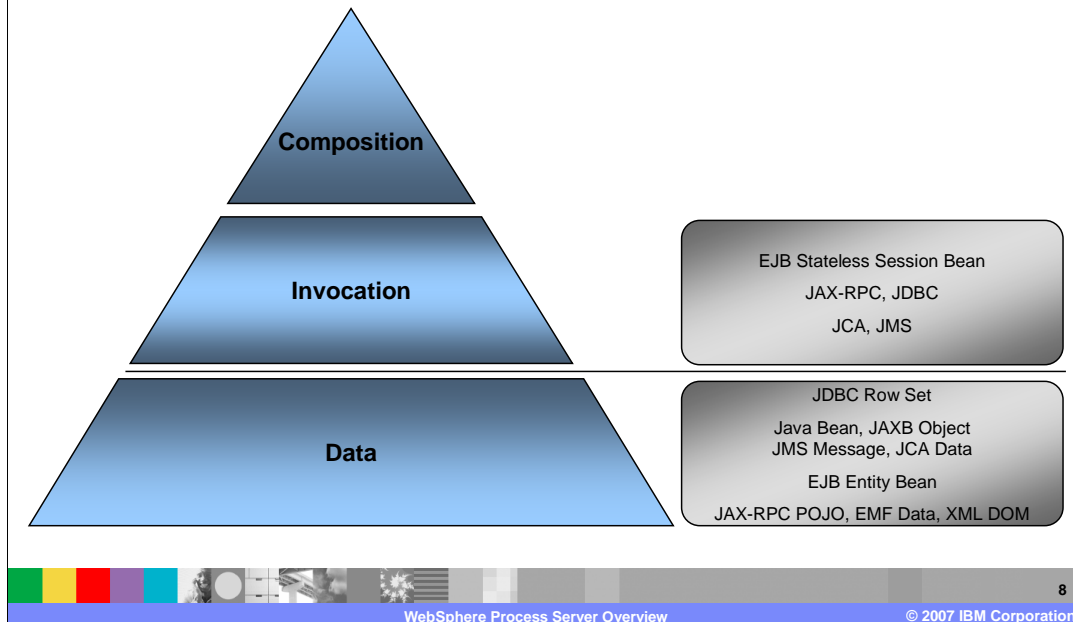
The elements of data, invocation and composition make up the programming model required to perform business processes. The following slides will describe how defining a business process with these programming model elements can be complex without WebSphere Process Server. The new programming model constructs introduced by WebSphere Process Server greatly simplify the programming model and therefore the overall development process.

Data without WebSphere Process Server

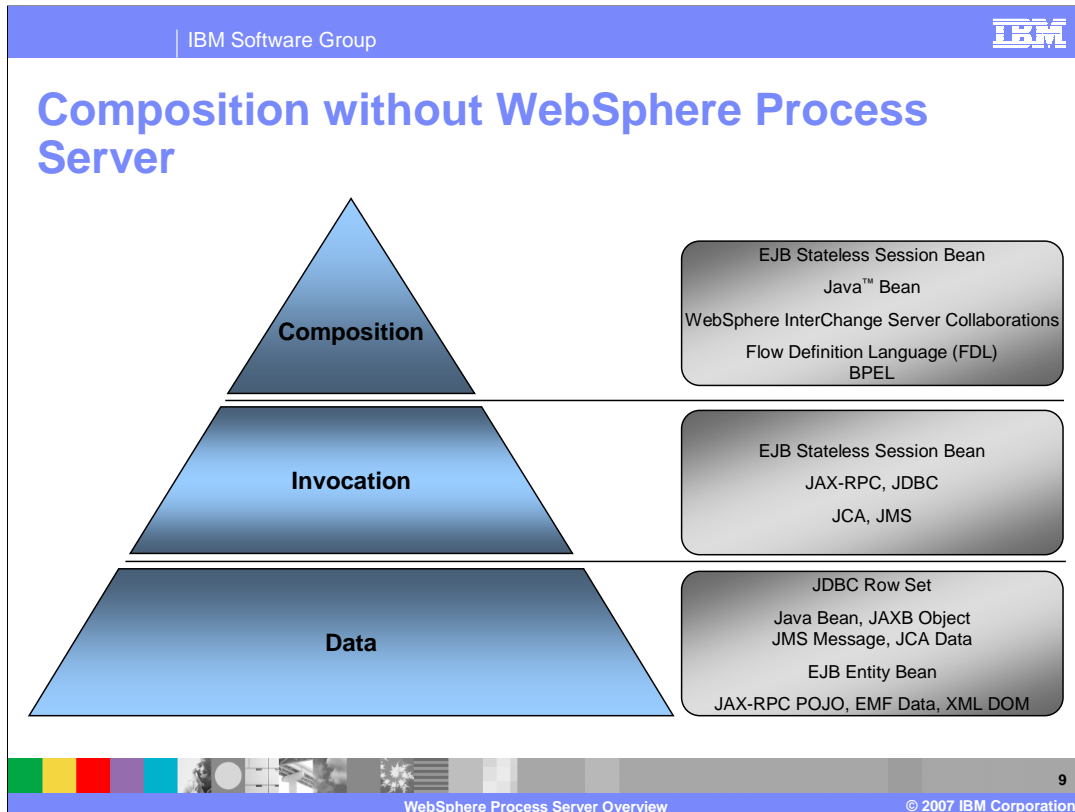


This slide shows that there are a wide variety of ways to represent and encode data. The possible representations include, but are not limited to, a JDBC™ row set from a relational database, a JMS message, JCA data coming in from an external application, a JAXB object, an EJB entity bean, Eclipse Modeling Framework data and many others.

Invocation without WebSphere Process Server



Invocation mechanisms are used to access, move and manipulate the data. The representation of the data determines what invocation mechanism must be used to do this. For example, JDBC must be used to obtain a JDBC row set from a relational database and JMS must be used to obtain data contained in a JMS message. Some of the invocation mechanisms that are used include EJB stateless session beans, JAX-RPC, JDBC, JCA for adapters, and JMS messaging.



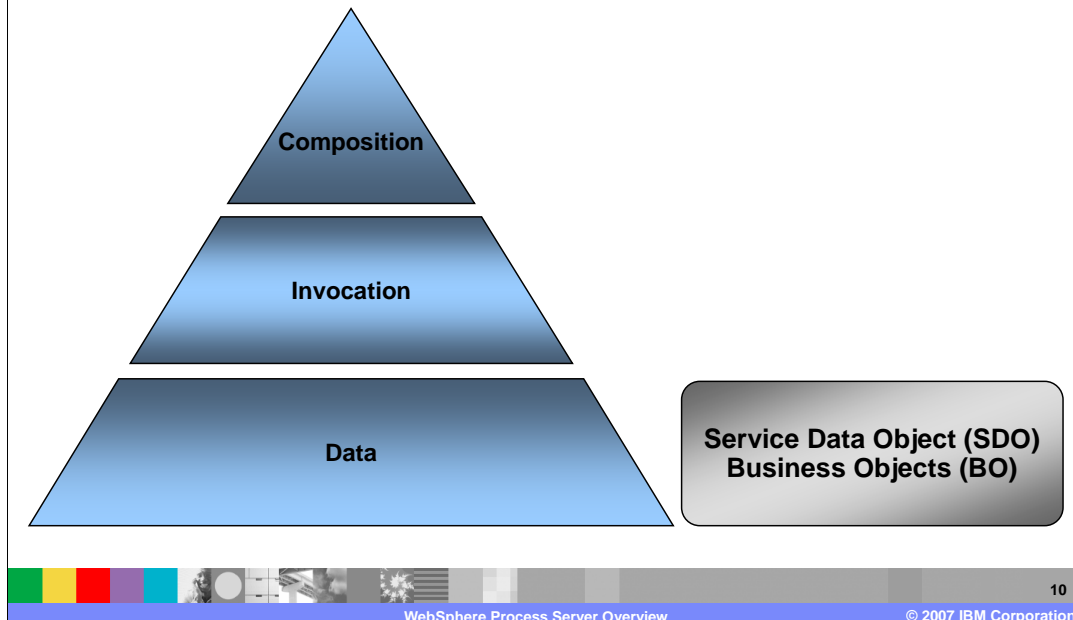
To build an application defining a business process, you must put together multiple invocations that access and process data. Composition defines how to assemble a series of invocations that work on data to construct a complete business process.

Composition can be done in a variety of ways, including, but not limited to, using a Java Bean or a stateless session bean, use of WebSphere InterChange Server collaborations, Flow Definition Language and Business Process Execution Language (BPEL).

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 be able to develop business processes. The representation that the data takes dictates the form of invocation and the type of invocations required can dictate the form of composition. Because of this, building the application for your business process is sometimes far from the optimal approach or the approach you would like to take.

As you can see, the overall result can at times end up as a complex solution.

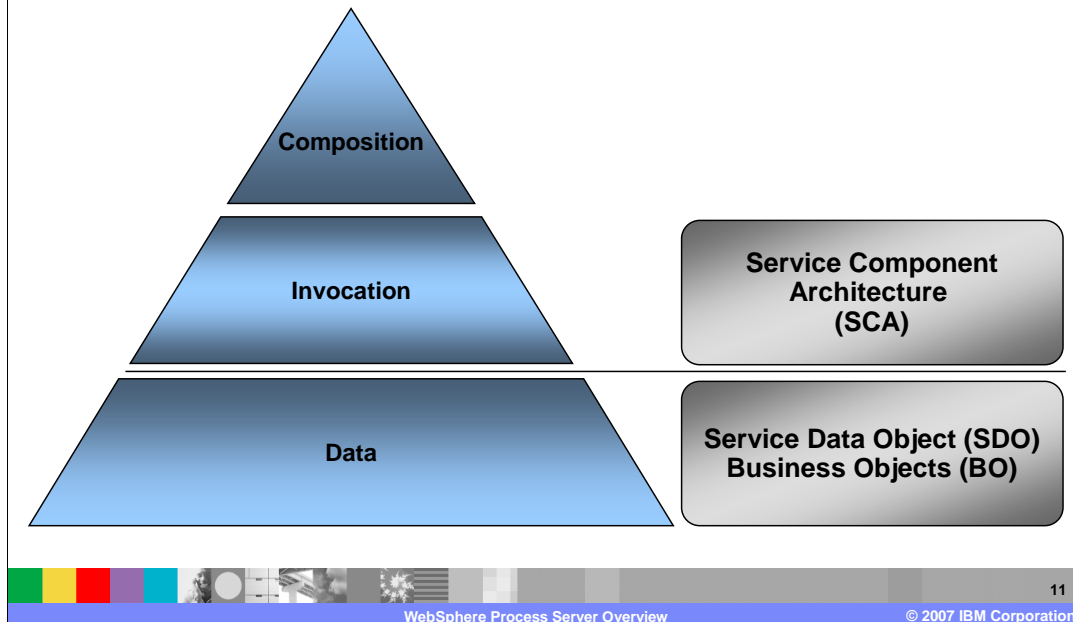
Data with WebSphere Process Server



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

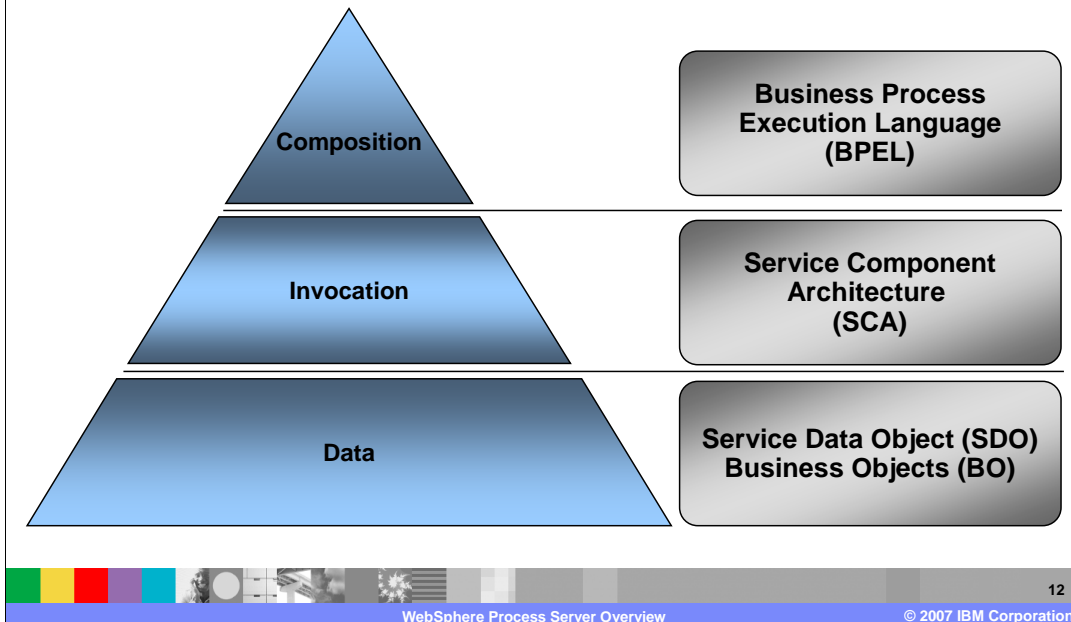
Data is defined using Service Data Object (SDO) technology. SDO is an evolving standard that provides an abstraction that can be used over various types of data representation, providing a common unified representation and mechanism for accessing data. WebSphere Process Server is based on the SDO 1.0 specification, with extensions. An SDO with the WebSphere Process Server extensions is called a Business Object (BO). The extensions for Business Objects contain additional information that is important for process integration solutions, further describing the data that is exchanged between services.

Invocation with WebSphere Process Server



Within WebSphere Process Server, invocation is done using Service Component Architecture (SCA), which provides a unified way to define and invoke services. With SCA, services are packaged as components with an interface. There are multiple different implementation types allowed for a component, but the invocation mechanism for a service component is the same across the varied implementation types. In addition, services can be associated with import and export bindings, which enable invocations to be made using various protocols, such as JMS messaging or Web Services, independent of the actual service implementation.

Composition with WebSphere Process Server



Finally, the composition of a business process is done using Business Process Execution Language (BPEL). BPEL allows you to build both short and long running business processes composed of business flow logic and invocations to services. The BPEL implementation in WebSphere Process Server version 6 is based on the WS-BPEL specification with extensions.

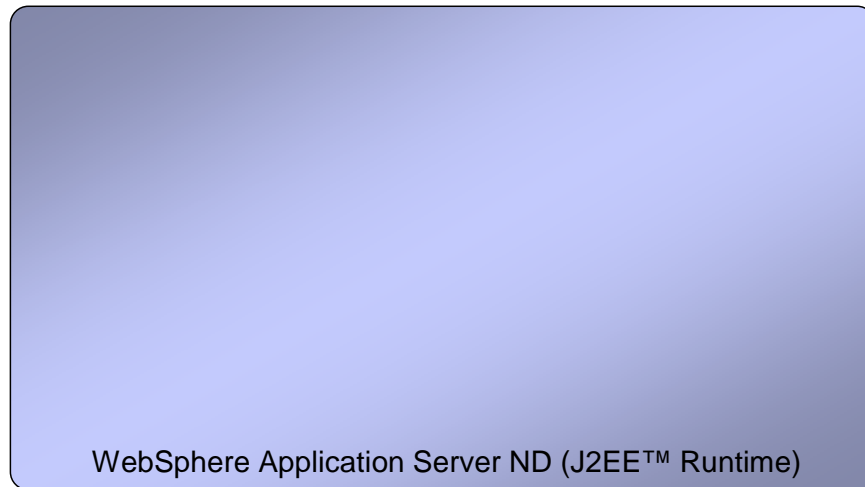
As you can see, WebSphere Process Server and the use of SDO, SCA and BPEL provides a much simpler programming model than was previously possible.

Section

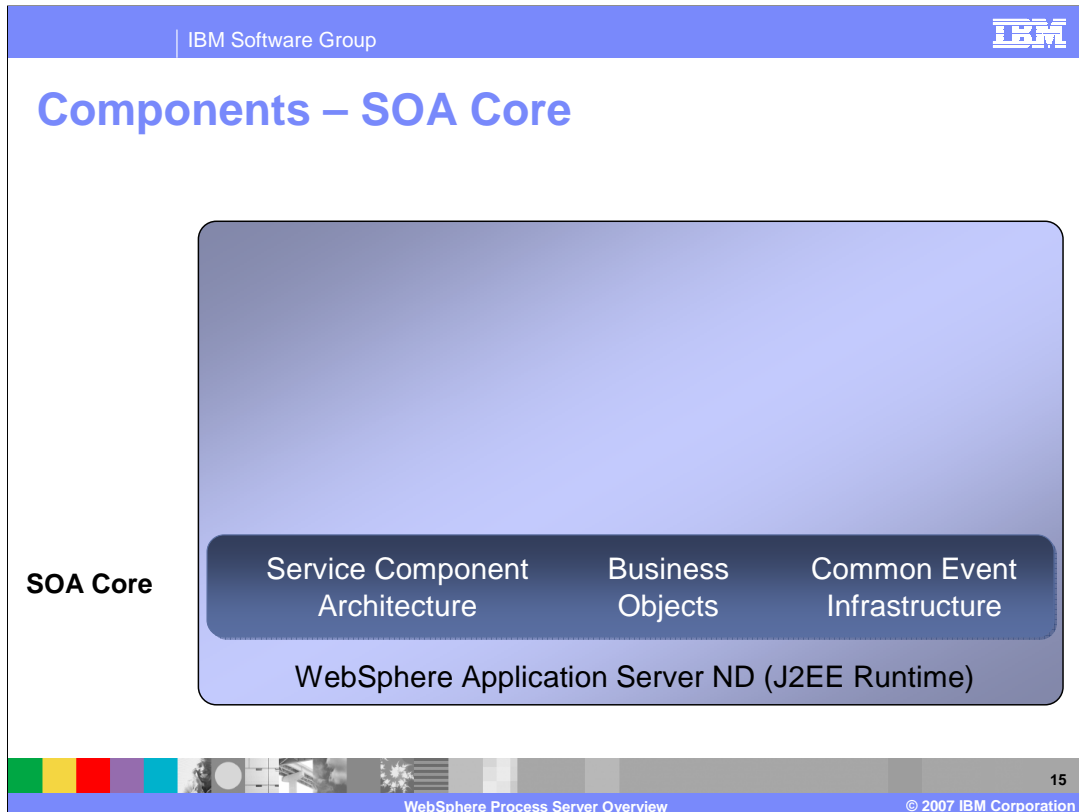
Component Architecture

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

Components – Base Application Server



WebSphere Process Server is built on WebSphere Application Server version 6.0.2, providing a robust J2EE application server runtime with capabilities that the process server implementation can exploit, such as JMS messaging, Web services support and Enterprise JavaBeans™. It can also make use of the application server qualities of service such as transactions, security and clustering. Overall, this provides a well proven and scaleable runtime environment for WebSphere Process Server.

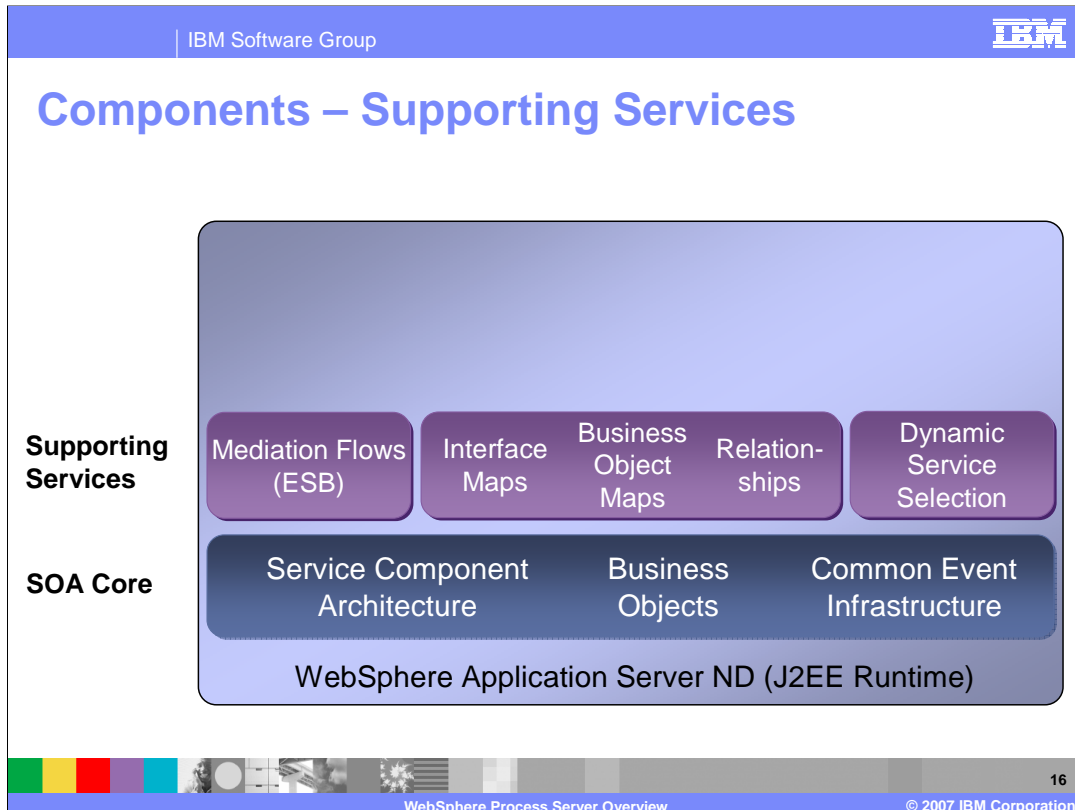


The Service Oriented Architecture Core is the foundation for the WebSphere Process Server. The main components of the SOA Core are the Service Component Architecture, Business Objects and the Common Event Infrastructure.

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 in the previous slides.

Business Objects represent the data that is passed within that framework and are extensions to Service Data Objects, carrying additional information needed for some integration scenarios. SDO in the form of Business Objects are another of the key components of the new programming model.

The Common Event Infrastructure (CEI) provides the basis for managing and handling events produced by business processes. It provides facilities for the generation, propagation, persistence and consumption of events, which 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 that 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 capabilities such as loose coupling between service requestors and service providers. Using Mediation Flows enables conversion of protocols and transformation of message content and formats. It also enables dynamic routing decisions to occur between a service requestor and service provider. If the SOA Core and Mediation Flows are all that you need, the WebSphere Enterprise Service Bus product delivers that subset of WebSphere Process Server capabilities.

The next set of supporting services are Interface Maps, Business Object Maps and Relationships, which are shown together because there is a close relationship between these supporting services.

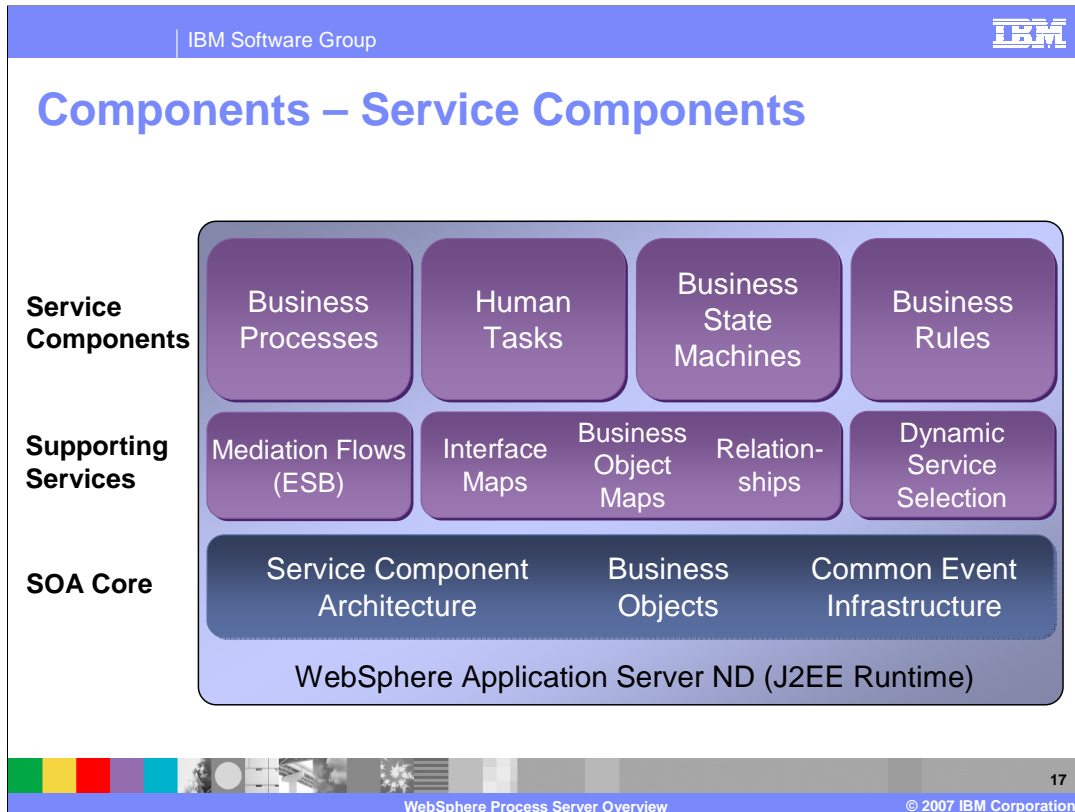
Interface Maps are used to enable an SCA component to make use of a particular interface to make calls to another SCA component that provides a semantically similar but syntactically different interface.

Business Object Maps enable the transformation of business data between Business Objects that are of differing types but represent the same business entity. 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 back end systems. Relationships are defined within the context of Business Object Maps.

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

The last of the supporting services is Dynamic Service Selection, which provides for dynamic invocation of a target component based on a date and time criteria.



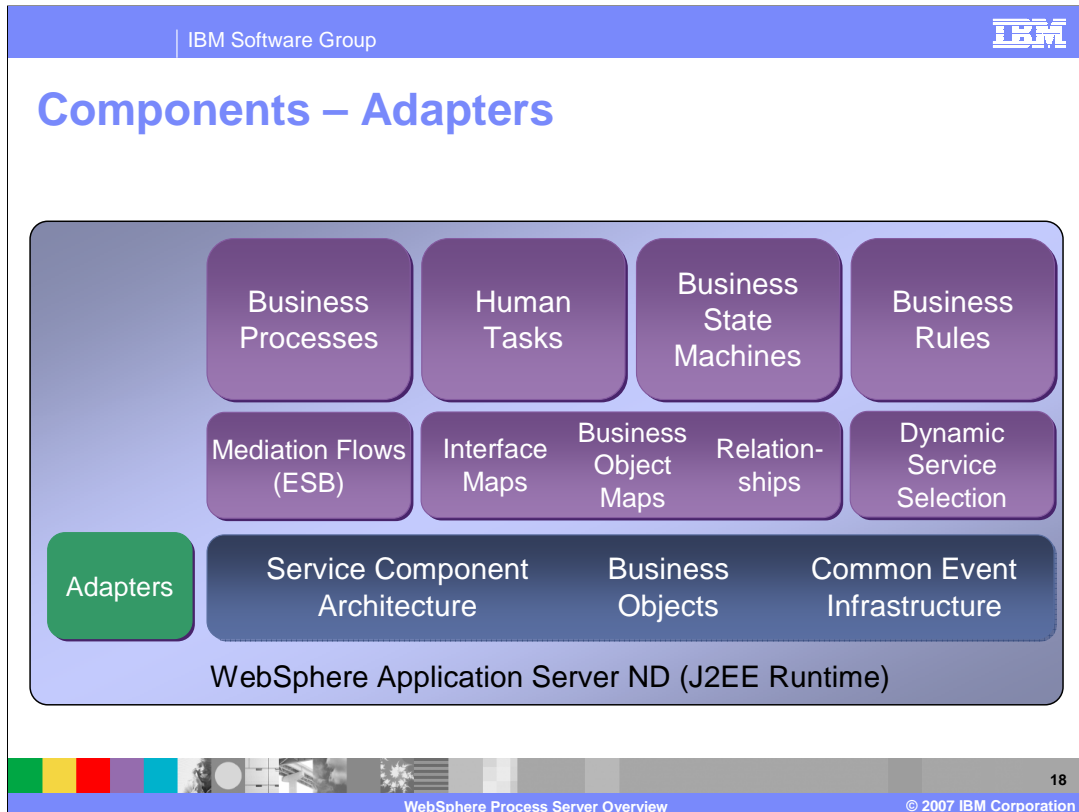
On top of the supporting services are the service components which enable the actual business functionality.

Business Processes are the first of these service components and are a fundamental part of the programming model, addressing the ability to do composition. 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 activities that comprise the process take place, 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.

The Human Tasks service component enables the Human Task Manager and provides 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 performed 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 a BPEL business process or packaged as an SCA component, enabling it for use by any client that can make SCA invocations.

The Business State Machine service component provides 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 handle the fact that an order can be canceled at any time up until it is 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.

The last service component, Business Rules, provides a means for 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 and therefore enables a more agile business. There are two styles of business rules, if-then rulesets and decision tables. To allow a business user to modify business rules, a Web client is provided. This client enables business rules to be changed using a natural language specification. The result is that the business user can make the changes, reducing the need for application developer or integration developer involvement.



Although not packaged as a part of WebSphere Process Server, adapters play an important role in enabling WebSphere Process Server to serve as a process integration platform.

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

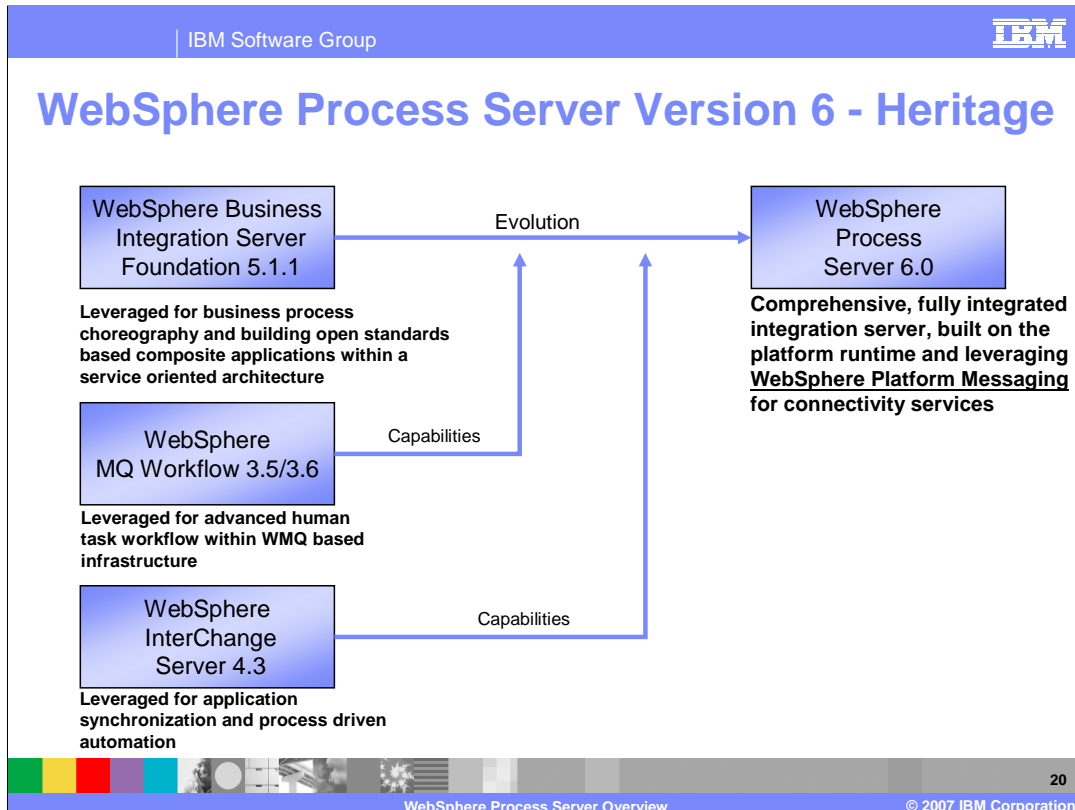
The functionality of an enterprise system is encapsulated by an adapter and business objects are used to pass data back and forth between the process server and the enterprise system. The use of business objects and service component architecture allow adapters to fit perfectly into the new programming model. The use of adapters provides a consistent framework for accessing backend systems.

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

Section

Heritage

This section covers the heritage of WebSphere Process Server version 6.



WebSphere Process Server version 6 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.

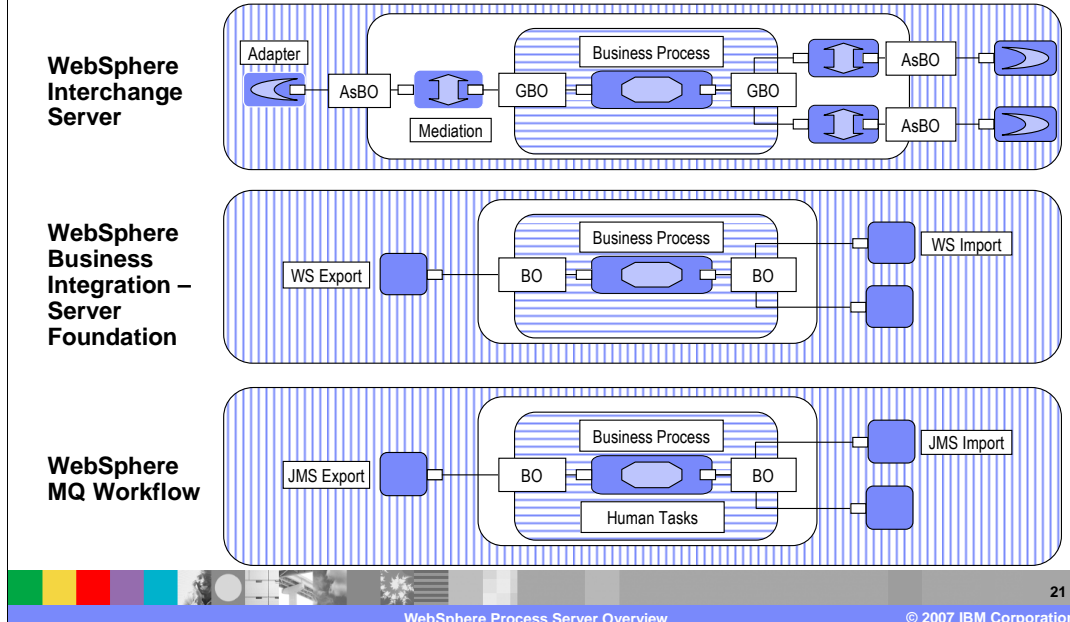
WebSphere Process Server is a direct evolution from WebSphere Business Integration Server Foundation v5.1.1. Just as WebSphere Business Integration Server Foundation was built on the J2EE application server environment provided by WebSphere Application Server V5, WebSphere Process Server is also built 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 has been designed into WebSphere Process Server. These include the integration of human task workflows and a strong messaging infrastructure.

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

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

Heritage Product Use Cases



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 of the slide 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. Once the process is complete, the output could be sent using JMS through a JMS SCA import.

Summary

- This presentation provided you with:
 - ▶ An introduction to WebSphere Process Server
 - ▶ The new programming model introduced by the server
 - ▶ The components comprising the server architecture
 - ▶ The WebSphere Process Server product heritage

In summary, this presentation provided an introduction to WebSphere Process Server. The new programming model, consisting of service component architecture, business objects and business process execution language was discussed. The components making up the WebSphere Process Server stack were also examined followed by a description of the heritage products that influenced the capabilities built into WebSphere Process Server.

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:

WebSphere

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.

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 (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.

© Copyright International Business Machines Corporation 2007. 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.

