Introduction to WebSphere Process Server and WebSphere Enterprise Service Bus
Unit objectives

- Describe the WebSphere Process Server and WebSphere ESB product offerings
- Identify the SOA core facilities provided by WebSphere Process Server
- Describe mediations and their relationship to WebSphere ESB
- Identify tools and programming models for WebSphere Process Server and WebSphere ESB
WebSphere Process Server and WebSphere ESB
Product Offerings

After completing this topic, you should be able to:
● Describe the fundamental services provided by WebSphere Process Server and WebSphere ESB
● List the architectural differences between WebSphere Process Server and WebSphere ESB
● Describe the relationship between WebSphere Process Server, WebSphere ESB, and WebSphere Application Server
● Describe the messaging resources used by WebSphere Process Server and WebSphere ESB
WebSphere Product Offerings for building an ESB

- **Business Process**
  - **WebSphere Process Server**
    A higher level solution to design, automate and manage composite applications and operational business processes.
    **Built on WebSphere ESB.**
  - **WebSphere ESB**
    A basic Enterprise Service Bus. Provides Web Services connectivity and data transformation.
    **Built on WebSphere Application Server.**
  - **WebSphere Application Server**
    A world-class J2EE foundation providing industry-leading levels of availability, scalability, and performance.
  - **WebSphere Message Broker**
    An advanced Enterprise Service Bus. Provides *universal* connectivity and data transformation.
    **Built on WebSphere MQ.**
  - **WebSphere MQ**
    Provides reliable integration messaging to connect applications and Web services across more than 80 supported platform configurations.

*Increased capabilities / automation*
WebSphere Process Server V6 highlights

- **WebSphere Application Server Foundation**
  - Clustering*, failover, high availability and robust platform
  - Single administration environment
  - Common Event Infrastructure – Process Management

- **Service Oriented Architecture platform**
  - A uniform invocation programming model (SCA)
  - A uniform data representation model (Business Objects)
  - Powerful tools to build and reuse standard components

- **Powerful Staff Components**
  - Participating and Originating and Ad-Hoc Tasks
  - Multi-level escalation
  - Client components that are ready to use (JSF)

- **Business Processes**
  - WS-BPEL standard

- **Business State Machines, Business Rules and Transformations**
  - Advanced services to build integration solutions

- **A single Process Integration platform**
IBM WebSphere Process Server V6

- IBM WebSphere Process Server V6 provides a single process-integration framework with the following business process features:
  - The **business flow manager** provides a run-time environment for WS-BPEL 2.0 business processes.
  - **Service component architecture** provides a universal service invocation programming model.
  - **Business objects** provide a universal data representation model.

- This solution is based on the IBM WebSphere Application Server V6 platform.
  - Deploy and manage enterprise applications in a J2EE 1.4 compliant application server.
  - Leverage programming model extensions for advanced features beyond the current J2EE standard.
  - Create and manage multiple application servers in a node, cell or cluster environment.
WebSphere Process Server Architecture Overview

**Service Components**
- Business Processes
- Human Tasks
- Business State Machines
- Business Rules

**Supporting Services**
- Mediation Flows
- Interface Maps
- Selectors
- Business Object Maps
- Relationships
- Adapters

**SOA Core**
- Service Component Architecture
- Business Objects
- Common Event Infrastructure

WebSphere Application Server (J2EE runtime)

© Copyright IBM Corporation 2006
WebSphere ESB

- Provides Web Services connectivity, JMS messaging, and service-oriented integration
  - Improve flexibility through the adoption of service oriented interfaces
  - Minimize disruption by using an ESB to handle integration logic
  - Allow the flow of business events and add needed intelligence to that flow
  - Simple to develop, build, test, deploy and manage (integrated, interactive and visual development)
  - Supports many ISV solutions
  - Pre-built mediation functions available
  - Dynamically re-configurable to meet changing business needs

- Seamless integration with WebSphere platform
  - Leverages WebSphere qualities of service: clustering, fail-over, systems management, security
  - Can be extended to leverage WebSphere Process Server as needs dictate
  - Integrates with IBM Tivoli security and systems management offerings
WebSphere ESB architecture overview

**Messaging:**
- MQ interoperability
- JMS 1.1

**Clients:**
- Lightweight Java Client coming
- C++ Client
- .Net Client
- Java and C/C++ Web Services Client

**WebSphere ESB**
- XSLT
- Message Logger
- Mediation Function
- Message Router
- DB Lookup

**WebSphere Application Server**
- Tivoli Access Manager
- DB2 Universal Database
- Edge Components
- UDDI
- Web Services Gateway

**Web Services:**
- SOAP/HTTP
- SOAP/JMS
- WS-*
- UDDI Registry 3.0

**SCA Programming Model:**
- WebSphere Adapter Support
- SCA
- SMO
- SDO
- SCA

© Copyright IBM Corporation 2006
WebSphere Application Server V6 offerings

- Provides the foundation for the WebSphere family of products.
- Provides the J2EE runtime.
- Delivers a high-performance and extremely scalable transaction engine for dynamic e-business applications.
- Available in multiple packages:
  - WebSphere Application Server V6 Express
  - WebSphere Application Server V6 (Base)
  - WebSphere Application Server V6 Network Deployment
- All product offerings provide the same core or base application server functionality.
WebSphere Application Server V6 messaging resources in WebSphere Process Server and WESB

- The Service Integration Bus (or SIBus) defines the messaging resources provided by WebSphere Application Server V6
  - Default messaging provider
  - Provides messaging infrastructure for messaging and service-oriented applications
  - A fully JMS 1.1 compliant JMS provider
    - Reliable message transport
    - Can also support intermediary logic to adapt message flow in the network
  - Supports the attachment of web services requestors and providers, as well as integration with WebSphere MQ
  - Capitalizes on WebSphere Application Server services
    - Security
    - Administration
    - Performance monitoring
    - Tracing
    - Problem determination
WebSphere Application Server, WebSphere Process Server, and WebSphere ESB
Having completed this topic, you should be able to:

- Describe the fundamental services provided by WebSphere Process Server and WebSphere ESB
- List the architectural differences between WebSphere Process Server and WebSphere ESB
- Describe the relationship between WebSphere Process Server, WebSphere ESB and WebSphere Application Server
- Describe the messaging resources used by WebSphere Process Server and WebSphere ESB
WebSphere Process Server – SOA core

After completing this topic, you should be able to:

● Describe the key architectural components of WebSphere Process Server
● Explain the SOA architecture of WebSphere Process Server
● Identify the supporting services provided by WebSphere Process Server
Service component architecture (SCA)
- SCA describes all integration artifacts as service components with well defined interfaces
- SCA defines the invocation model

Business Objects
- Universal means of describing data
- Based on Service Data Objects (SDO) standard

Common Event Infrastructure
- Used for capturing events that can be used to monitor applications.
WebSphere Process Server – supporting services

- Simplify common integration tasks
  - Simple mapping of fields from one representation to another.
  - Mapping of interface signatures (that is, operations and their parameters) to other interfaces representing the same underlying functionality and/or semantics.
  - Relationship management between data from different systems.
    - Allows a dynamic binding to be created between data (for example, Business Objects) that are semantically equivalent but were derived from different back-end Enterprise Information Systems in their specific formats.
WebSphere Process Server service components (1 of 2)

- **Business process**
  - WS-BPEL compliant process engine

- **Business State Machine**
  - For modeling a business process based on states and events
  - Example: an ordering process where the order can be changed or canceled at any time
WebSphere Process Server service components (2 of 2)

- **Human Tasks**
  - The machine can assign work to a person (and vice-versa)
  - A person can invoke a service of another person

- **Business Rules**
  - A means of implementing and enforcing business policy through externalization of business functions
  - Enables dynamic changes of a business process for a more responsive business environment
Having completed this topic, you should be able to:
- Describe the key architectural components of WebSphere Process Server
- Explain the SOA architecture of WebSphere Process Server
- Identify the supporting services provided by WebSphere Process Server
After completing this topic, you should be able to:

● Describe the key architectural components of WebSphere ESB supporting the development and deployment of mediation services
● Describe how the WebSphere ESB architecture supports integration with EIS services, WebSphere MQ services and WebSphere Message Broker
● Describe the WebSphere ESB client support
WebSphere ESB mediation

- Mediation: a service request interception by an ESB that typically centralizes logic
  - Routing
  - Transformation
  - Logging
  - Filter
  - Data handling

- Mediation flows are created to assemble mediations from pre-supplied components
  - No need to write integration logic code
  - Potential for libraries of third-party nodes
  - Packaged in mediation modules
Mediation Flows operate on messages between endpoints. Mediation primitives need to be able to operate on the messages.

Service Message Objects (SMOs) provide the common representation of a message for use by mediation primitives. SMO uses Service Data Object (SDO) to represent messages. All SMOs have the same basic structure as defined by the schema. Three major sections:
- body
- headers
- context

All information in the SMO is accessed as an SDO DataObject. Using XPath, using the generic DataObject APIs, using SMO specific APIs which are aware of the SMO schema.
WebSphere ESB – Integrating mediation services with EIS

- WebSphere Adapters allow components to communicate with a variety of EIS infrastructures
  - Uses a consistent SCA programming model
  - Adapters provide interfaces that define EIS functions and events
  - Data is passed to and from the EIS by business objects

- Mediation modules can use adapters to bind to EIS services
  - EIS imports define how to bind to an external EIS service
  - EIS exports subscribe a service to listen for external EIS events

- There are two types of adapters
  - J2EE Connector Architecture (JCA 1.5)
  - WebSphere Business Integration Adapters (based on JMS)
WebSphere ESB – Integrating mediation services with WebSphere MQ services

- SIBus can appear as a queue manager to WebSphere MQ using an MQ link

- The WebSphere MQ link converts:
  - WebSphere MQ message fields and properties into JMS fields and properties
  - MQRFH2 header fields to JMS
  - MQMD Report fields to JMS provider-specific properties

- Mediations can be attached to WebSphere MQ messages containing a forward routing path in the MQRFH2 header
  - For a reverse routing path, the ReplyToQ and ReplyToQMGR fields in the embedded message descriptor of the MQXQH transmission queue header represent a destination that is the first destination in the reverse routing path
WebSphere ESB – Integrating mediation services with WebSphere Message Broker

- WebSphere ESB and WebSphere Message Broker interoperate
  - WebSphere ESB handles standards-based web service interactions
  - WebSphere Message Broker provides advanced support for a wide range of messaging formats

Standards-based ESB: WebSphere ESB
- Web Services connectivity and data transformation:
  - HTTP, JMS, WebSphere MQ, Web Services, XML, WebSphere Adapters

Advanced ESB: WebSphere Message Broker
- Universal connectivity and data transformation:
  - HTTP, JMS, WebSphere MQ, Web Services, XML, WebSphere Adapters
  - Plus the following: Weblogic JMS®, Biztalk®, TIBCO EMS JMS®, TIBCO Rendezvous®, MQe, Multicast, Tuxedo®, FTP, COBOL Copybook, HIPAA, EDI-FACT, HL7, SonicMQ JMS®, ACORD, Real-time IP, AL3, Word/Excel/PDF, SWIFT, FIX, ebXML, EDI-X.12, MQTT, Custom Formats
WebSphere ESB client support

- WebSphere ESB provides message service clients that extend the connectivity of the enterprise service bus
  - **Message service clients for C/C++ and .NET**
    - Also known as XMS – brings the benefits of JMS to the non-Java world
    - XMS API has the same set of interfaces as the JMS API
  - **Web services client for C++**
    - A set of libraries and Java tools that enable the building of ANSI C++ web service client applications from existing WSDL files
    - The client code produced from the toolkit:
      - Is platform independent
      - Is development environment independent
      - Supports WS-I 1.0 base profile compliance
      - Supports SSL
      - Has a JAX-RPC style implementation
  - **J2EE client support from WebSphere Application Server ND**
    - Web services
    - EJB client
    - JMS client
Having completed this topic, you should be able to:

● Describe the key architectural components of WebSphere ESB supporting the development and deployment of mediation services

● Describe how the WebSphere ESB architecture supports integration with EIS services, WebSphere MQ services and WebSphere Message Broker

● Describe the WebSphere ESB client support
Developing for WebSphere Process Server and WebSphere ESB

After completing this topic, you should be able to:

- Describe design considerations and alternatives for developing mediation services using capabilities offered by WebSphere Process Server and WebSphere ESB
- Explain the role of WebSphere Integration Developer for component development
- Describe the programming model for WebSphere Process Server and WebSphere ESB
Design considerations for mediation services

- When implementing and deploying mediations, there are several choices

1. Use supporting services only, and deploy to WPS
   - Selectors
   - Interface Maps
   - Business Object Maps
   - Relationships

2. Develop mediation services as mediation modules (logical separation of concerns)
   - Deploy them to WebSphere Process Server
   - Deploy them to WebSphere ESB

3. Use a hybrid approach dictated by your requirements
WebSphere Integration Developer overview

- **IBM WebSphere Integration Developer V6** is a development environment for building integrated applications based on a service-oriented architecture (SOA.)
- Enables integration developers to create, manage and test services for IBM WebSphere Process Server V6 and WebSphere ESB V6.
- The features in Integration Developer V6 aimed at separating business logic from implementation details.
  - The business integration perspective and features provide graphical tools for integrating business data, interfaces and logic.
Development platform architecture

- IBM WebSphere Integration Developer V6 is based upon IBM Rational Application Developer V6
  - Includes the core J2EE 1.4 development features for the IBM WebSphere Application Server V6 platform

- Both development tools are built on top of the IBM Rational Software Development platform
  - Based on Eclipse 3.0
  - Installed once per system with the first product
WebSphere Integration Developer

- WebSphere Integration Developer provides an integrated, interactive and visual development environment for rapid development of integration logic
  - Requires minimal knowledge of Java or J2EE
- Simple to develop, build, test, deploy and manage services components
  - Get up and running quickly with comprehensive documentation, easy to understand samples
  - Provides a simplified and visual development experience for standards-based artifacts like XML schema, WSDL, XSLT, etc.
  - Supports the declaration of services and connectivity through a visual composition model
  - Allows easy orchestration of mediation functions with first-class support for intelligent message routing, enrichment, and transformation
  - Offers a seamless integrated tooling approach to connect between service-oriented and message-oriented services
  - True role-based support provides a simplified administration experience
WebSphere Process Server and WebSphere ESB programming model overview

- Modules are the basic deployment unit for WebSphere Process Server and WebSphere ESB
  - Modules for business services (implements the logic of a process)
  - Mediation modules (transforming a service invocation)

- Steps to develop a module
  1. Define interfaces for the components in the module
  2. Define, modify or manipulate business objects used by service components
  3. Define or modify service components through its interfaces
     - For mediation components, you can use the mediation flow editor to map between operations on the end points of a mediation, and use mediation primitives to compose the mediation logic visually.
  4. Optionally, export or import service components
  5. Create an EAR file for installing a module that uses components
Having completed this topic, you should be able to:

- Describe design considerations and alternatives for developing mediation services using capabilities offered by WebSphere Process Server and WebSphere ESB
- Explain the role of WebSphere Integration Developer for component development
- Describe the programming model for WebSphere Process Server and WebSphere ESB
WebSphere Process Server and WebSphere ESB relationship

![Diagram showing the relationship between WebSphere Process Server and WebSphere ESB]

- **WebSphere Process Server**
  - Business Processes
  - Human Tasks
  - Business State Machines
  - Business Rules
  - Interface Maps
  - Selectors
  - Business Object Maps
  - Relationships
  - Adapters

- **WebSphere ESB**
  - Mediations
  - Service Component Architecture (SCA)
  - Business Objects
  - Common Event Infrastructure

- **SOA Core**
  - Clustering (+Virtualization)
  - WebSphere Application Server
  - Network Deployment
  - AppServer Foundation

- **Choreography**
  - Supporting Services

© Copyright IBM Corporation 2006
Having completed this unit, you should be able to:

- Describe the WebSphere Process Server and WebSphere ESB product offerings
- Identify the SOA core facilities provided by WebSphere Process Server
- Describe mediations and their relationship to WebSphere ESB
- Identify tools and programming models for WebSphere Process Server and WebSphere ESB