WebSphere Process Server for z/OS

Product Overview

Version 7.0.0

30 April 2010

This edition applies to version 7, release 0, modification 0 of WebSphere Process Server for z/OS (product number 5655-N53) and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, send an e-mail message to doc-comments@us.ibm.com. We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Product overview for WebSphere Process Server

WebSphere® Process Server is a high-performance business engine to help form processes to meet business goals. It allows the deployment of standards-based business integration applications in a *service-oriented architecture* (SOA), which takes everyday business applications and breaks them down into individual business functions and processes, rendering them as services.

Related information

PDF documentation
WebSphere Process Server documentation (in PDF format)

Information roadmaps
Business Process Management information roadmaps on IBM®developerWorks organize information about WebSphere Process Server, WebSphere ESB, and the other products in the portfolio.

IBM Education Assistant
Multimedia educational modules about WebSphere Process Server, provided by IBM Education Assistant.

Overview
Overview tab, on product library Web page. Use this page to access announcements, data sheets, and other general library documents related to WebSphere ESB.

Introduction to WebSphere Process Server

IBM WebSphere Process Server is a business process integration server that has evolved from proven business integration concepts, application server technologies, and the latest open standards. WebSphere Process Server is a high-performance business engine to help form processes to meet business goals.

WebSphere Process Server allows the deployment of standards-based business integration applications in a service-oriented architecture (SOA), which takes everyday business applications and breaks them down into individual business functions and processes, rendering them as services. Based on the robust Java EE infrastructure and associated platform services provided by WebSphere Application Server, WebSphere Process Server can help you meet current business integration challenges. This includes, but is not limited to, business process automation.

WebSphere Process Server enables the deployment of processes that span people, systems, applications, tasks, rules, and the interactions among them. It supports both long-running and short-running business processes, providing transaction rollback-like functionality for loosely coupled business processes.

Hardware and software requirements

To view the official statement of supported hardware and software for WebSphere Process Server, see the WebSphere Process Server system requirements Web site.

Information roadmaps

To help you to navigate through the available information sources, both within and beyond the product information centers, business process management information roadmaps are available online on IBM developerWorks® at www.ibm.com/developerworks/websphere/zones/bpm/roadmaps/bpm_info_resources.html.

What is new in this release

WebSphere Process Server for z/OS®, version 7.0, includes enhanced capabilities for working with and administering human workflows, accelerated productivity across all process roles, time to value for implementing and deploying solutions, easier migration, and enhanced support for open standards. This version includes enhancements for migration, database scalability, and platform alignment.

Note: The information center has been updated for IBM WebSphere Process Server for z/OS, version 7.0.0.2. For more information about installing fix packs, see the APAR and PTF tables for WebSphere Process Server and for WebSphere Enterprise Service Bus.

New in WebSphere Process Server for z/OS, version 7.0.0.2

Welcome to WebSphere Process Server for z/OS, version 7.0.0.2, which includes the following new features:

- Migration enhancement that makes it possible for WebSphere Process Server for z/OS environments running V6.0.2 to migrate to V7.0 in a single step.
- Consumability improvements, including the ability to specify cluster names and cluster member names during the creation of a deployment environment.
- Platform alignment and currency:
  - This fix pack is the minimum supported fix pack level for the product on the Microsoft® Windows® Server 2008 R2 and Windows 7 operating systems.

New in WebSphere Process Server for z/OS, version 7.0

WebSphere Process Server for z/OS, version 7.0, includes the following new features:

- Enhanced capabilities for working with and administering human workflows.
  - Enables the business to react quickly to changing business requirements by allowing installation of new versions of a process, and allowing the migration of running processes to a new version, in order to use those changes immediately.
  - Supports additional human workflow scenarios, including parallel approval with voting and result aggregation.
  - Uses the versatility of the human task and workflow widgets in Business Space in additional scenarios such as human task, workflow, and escalation management. Uses innovative capabilities including on demand multi-column filtering and adaptive paging.
  - Provides richer capabilities for process administrators to manage in-flight processes, including modifying ownership of a process instance, and enhanced activity repair capabilities such as resetting timers and repairing correlation sets.
- Enhancements that empower users and accelerate productivity across all process roles.

  - Helps improve productivity with faster deployment of BPM solutions -- from WebSphere Business Modeler and the command line -- and faster iterative development with WebSphere Integration Developer.
  - Offers improved user experience for Interactive Process Design scenario with faster deployment.
  - Enhances operational visibility with new and improved Business Space widgets for better service monitoring and health and problem determination.
  - Improves problem determination with consistent fault handling across Service Component Architecture (SCA) bindings and cross-component trace enhancements.
  - Enhances operational flexibility with new and improved Business Space widgets for better module administration.
- Improved time to value for implementing and deploying BPM solutions.
  - Simplifies system installation, including easier cluster configuration.
  - Provides consistent, flexible, and independent BPM topology and database configuration and management.
  - Simplifies handling of runtime environment outages with support for unexpected service downtime with "store and forward" capability to queue events until service is restored.
  - Eases the process of loading or unloading static relationship data with a data import and export capability.
  - Uses Web-based forms rendered from Lotus® Forms Server, in addition to the existing Lotus Forms Client rendering capabilities.
- Enhancements designed to ease the effort of migrating from WebSphere Business Integration server solutions.
  - Enhances the maintainability of the generated BPEL from migrated WebSphere InterChange Server repositories.
  - Improves performance of WebSphere InterChange Server migration for improved user experience when migrating large repositories.
- Enhanced support for open standards.
  - Java™ enhancements including Java EE 5, EJB 3.0, JPA, Java SDK 6 support, and enhanced Java integration.
  - Web services enhancements including WS-Addressing, Attachments, Kerberos token profile, and WS-Policy support.
  - Interoperability with Open SCA, enhanced OSGi support, and enhanced XML fidelity.
- Platform alignment and currency.
  - Uses and extends WebSphere Application Server V7.0, providing enhanced standards support, simplified system installation and administration, and enhanced WebSphere MQ V7 integration.
  - Enables the use of Microsoft SQL Server 2008 as the underlying database for storing WebSphere Process Server program data (excluding Business Process Choreographer Explorer reporting capabilities).
  - On z/OS, delivers improved integration of the WebSphere Customization Tool with the WebSphere Process Server for z/OS installation experience, enhanced tool support to assist with the DB2® database creation process, and a Common Installer Framework that provides an integrated experience for all z/OS BPM products.
- Moving to V7.0 from an earlier version.

  - Extends and improves Information Center content to facilitate version-to-version migration.
  - Has common commands and a user interface for version-to-version migration utilities across WebSphere Dynamic Process Edition, WebSphere Business Services Fabric, WebSphere Process Server, WebSphere ESB, WebSphere Business Monitor, and WebSphere Business Compass.
  - On all platforms, WebSphere Process Server or WebSphere ESB environments running V6.0.2, V6.1.0, V6.1.2, or V6.2 can migrate to V7.0 in a single step.

What is new in the product information

Product information for WebSphere Process Server includes additions and enhancements in version 7.0

The WebSphere Process Server for z/OS, version 7.0 product information includes the following new features and enhancements:

- Enhanced welcome pages provide tabbed areas for learning, accomplishing tasks, and finding additional support.
- Enhanced capabilities for searching product information:
  - Linked search terms in "tag clouds" at the beginning of topics improve indexing for search and provide a useful, alternative means of navigating the documentation.
  - Information center software has been updated to improve the search engine indexing of product information.
- New migration information describes integrated, cross-product procedures, commands, and user interfaces for migrating business process management runtime products.
- New programming information describes how to program various components, applications, and business process management solutions. The information provides a cross-product, system-level perspective on important developer concepts such as Service Component Architecture (SCA) and the business object framework.
- Improved configuring information for WebSphere Process Server includes topics on common configurations, network deployment, database configuration, and Business Space.
- New scenarios describe products and key tasks in terms of high-level user goals and the business process management lifecycle.
- Updated software for the help system provides a more consistent user interface, bookmarking functions, and other features for managing information.
- Expanded documentation is provided for WebSphere Process Server widgets for Business Space.
- Links to resources outside of product documentation enrich the information in the information centers.
- New information about enterprise service bus functionality in WebSphere Process Server:
  - Tasks and tutorials about administering services and mediation policies with widgets
  - Topics about proxy gateways (including a new tutorial)
  - Updates to mediation policy information
  - Guidance for interoperability between SCA modules and Open SCA services

Product family overview

WebSphere Process Server is part of the IBM WebSphere Business Process Management platform and works with many other IBM products.

IBM WebSphere Application Server

WebSphere Process Server is based on the robust Java EE infrastructure and associated platform services provided by WebSphere Application Server. WebSphere Process Server for z/OS is built on WebSphere Application Server for z/OS. WebSphere Application Server includes a built-in JMS engine, for messaging between Java EE applications, and connectivity for messaging with WebSphere MQ. For more information about offerings, see the WebSphere Application Server documentation.

IBM WebSphere Enterprise Service Bus

WebSphere Process Server provides a fully converged, standards-based business process engine, using the full power of WebSphere Application Server. It also includes the same technology as WebSphere Enterprise Service Bus, providing the same enterprise service bus capabilities.

No additional license for WebSphere Enterprise Service Bus is required to take advantage of the enterprise service bus capabilities. However, you can deploy additional, purchased stand-alone licenses of WebSphere Enterprise Service Bus around your enterprise to extend the connectivity reach of the process integration solutions powered by WebSphere Process Server. For example, WebSphere Enterprise Service Bus can be installed closer to an SAP application to host an IBM WebSphere Adapter for SAP and to transform SAP messages before sending that information across the network to a business process choreographed by WebSphere Process Server.

IBM WebSphere Integration Developer

WebSphere Integration Developer is the development environment for WebSphere Process Server. It is a common tool for building service-oriented architecture (SOA)-based integration solutions across WebSphere Process Server, WebSphere Enterprise Service Bus, and WebSphere Adapters. For more information about WebSphere Integration Developer, see the WebSphere Integration Developer documentation.

IBM WebSphere Dynamic Process Edition

WebSphere Dynamic Process Edition is a bundle providing a comprehensive set of software offerings designed for enterprise-wide integration capabilities and service-oriented architecture (SOA). You can optimize business processes with dynamic capabilities for rapid change and customization. WebSphere Dynamic Process Edition is built on SOA-enabled products and capabilities that provide the foundation for integrating end-to-end business processes across an enterprise. It includes three products: WebSphere Business Modeler, WebSphere Business Services Fabric, and WebSphere Business Monitor. For more information about WebSphere Dynamic Process Edition, see the WebSphere Dynamic Process Edition product documentation library.

IBM WebSphere Business Services Fabric

WebSphere Business Services Fabric provides an end-to-end platform for the rapid assembly, delivery, and governance of industry-focused composite business services in an SOA. WebSphere Service Registry and Repository provides the technical service metadata underpinning, serving as a prerequisite for the WebSphere Business Services Fabric solution. For more information about WebSphere Business Services Fabric, see the WebSphere Business Services Fabric product documentation library.

IBM WebSphere Business Modeler

IBM WebSphere Business Modeler and IBM WebSphere Business Compass provide a comprehensive set of tools for business process modeling, Business Process Management (BPM) design, and modeling collaboration, to help your business users to document, visualize, analyze and design business process solutions.WebSphere Process Server and WebSphere Integration Developer include additional capabilities that make it possible to model, build, deploy, install, configure, run, and manage integration applications. WebSphere Integration Developer complements IBM WebSphere Business Modeler. For more information about WebSphere Business Modeler, see the WebSphere Business Modeler documentation

IBM WebSphere Business Compass

WebSphere Business Compass helps subject matter experts fully collaborate on their business processes and monitor solutions through a Web-based browser.

IBM WebSphere Business Compass Advanced Edition

WebSphere Business Compass Advanced Edition provides a package consisting of 10 copies of WebSphere Business Modeler Advanced, and one copy of WebSphere Business Compass.

IBM WebSphere Business Monitor

IBM WebSphere Business Monitor provides business users and managers with real-time, end-to-end views of business processes and operations. It provides customizable business dashboards that calculate and display key performance indicators (KPIs) and metrics derived from business processes, business activity data, and business events from a wide range of information sources, enabling you to proactively manage business problems or take advantage of business opportunities. WebSphere Process Server and WebSphere Integration Developer include additional capabilities that make it possible to monitor integration applications. WebSphere Integration Developer complements IBM WebSphere Business Monitor. For more information about WebSphere Business Monitor, see the WebSphere Business Monitor documentation.

IBM WebSphere Service Registry and Repository

WebSphere Service Registry and Repository (WSRR) is a system for storing, accessing and managing information, commonly referred as service metadata, used in the selection, invocation, management, governance and reuse of services in a successful service-oriented architecture (SOA). For example, it is where you store information about services in your systems, or in systems from other organizations, that you already use, plan to use, or want to be aware of. For example, an application can check the WSSR just before invoking a service to locate the service instance best satisfying its functionality and performance needs. WSRR also plays a role in other stages of the SOA life cycle. For more information about WebSphere Service Registry and Repository, see the WebSphere Service Registry and Repository documentation

Media Extender for WebSphere Process Server V7.0

Media Extender for WebSphere Process Server provides extended service mediation facilities that can be used as components in workflows that handle large multimedia files, making the workflow simpler. The mediation facilities form part of a solution to enable business transformation by linking business and content systems together for effective media, or other rich content, management.

IBM WebSphere MQ

WebSphere MQ, available on more than 80 platform configurations, offers application integration connectivity and integrates many differing platforms, systems, and applications. It delivers heterogeneous messaging, extending your enterprise service bus with reliable message delivery. For more information about WebSphere MQ, see the WebSphere MQ product documentation library.

IBM WebSphere Message Broker

WebSphere Message Broker makes full use of the transport and connectivity options provided by WebSphere MQ and WebSphere Application Server, and allows additional routing and transformation capabilities to implement an integration-based enterprise service bus. For more information about WebSphere Message Broker, see the WebSphere Message Broker product documentation library.

IBM WebSphere DataPower® SOA Appliances

WebSphere DataPower SOA Appliances are easy-to-deploy network devices that simplify, help secure, and accelerate XML and Web services deployments. They extend core SOA infrastructure components such as WebSphere ESB, WebSphere MQ, WebSphere Message Broker, and WebSphere Process Server. For more information about WebSphere DataPower SOA Appliances, see the WebSphere DataPower SOA Appliances product documentation library.

IBM WebSphere Portal

WebSphere Portal provides web-based client access to applications, as well as to human tasks and business processes enacted by WebSphere Process Server.

For more information about WebSphere Portal, see the WebSphere Portal product documentation library.

IBM WebSphere Adapters

WebSphere Adapters allow for integration of existing Enterprise Information System infrastructure and applications that are deployed on WebSphere Process Server. WebSphere Adapters enable you to quickly and easily create integrated processes that exchange information between enterprise resource planning, human resource, customer relationship management, and supply chain systems.

Application adapters extract data and transaction information from cross-industry and industry-specific packaged applications and connect them to a central hub. Technology adapters provide connectivity to access data, technologies and protocols that enhance integration infrastructure. You can use the Adapter Development Toolkit to create custom adapters.

Some WebSphere Adapters are included components with WebSphere Integration Developer.

For more information about WebSphere Adapters, see the WebSphere Integration Developer documentation.

IBM Rational® Application Developer and IBM Rational Software Architect

WebSphere Integration Developer can be used in conjunction with Rational Application Developer, or Rational Software Architect, to create a unique, integrated, and powerful integration development platform.

For more information about these products, see the Rational Application Developer Information Center, and the Rational Software Architect Information Center.

IBM CICS® Transaction Gateway and IBM WebSphere Host Access Transformation Services

An IBM enterprise modernization portfolio that includes CICS Transaction Gateway and WebSphere Host Access Transformation Services allows you to extend existing applications for reuse in enterprise processes.

For more information about these products, see the CICS Transaction Gateway Library and the WebSphere Host Access Transformation Services (HATS) Information Center.

IBM WebSphere Extended Deployment

WebSphere Extended Deployment provides a WebSphere Process Server network deployment environment with the ability to adjust the resources between clusters in the environment to meet processing objectives that you define as policies. Because of the ebb and flow of application volumes, there can be insufficient processing power available to satisfy requests during peak periods, and it can be difficult to optimize resources so that critical applications get needed processing time.

Dynamic reapportioning of processing power at these times can help you meet business needs. WebSphere Extended Deployment dynamically removes resources from clusters with low application volumes and adds them to clusters that are servicing the applications that require the additional resources. The processing priorities are specified in WebSphere Extended Deployment as policies.

For more information about WebSphere Extended Deployment, see the WebSphere Extended Deployment Information Center.

IBM WebSphere Transformation Extender

WebSphere Transformation Extender is a powerful, transaction-oriented, data integration solution that automates the transformation of high-volume, complex transactions without the need for hand-coding. It performs transformation and routing of data from source systems to target systems in batch and real-time environments. The sources can include files, relational databases, message-oriented middleware (MOMs), packaged applications, or other external sources. After retrieving the data from its sources, the WebSphere Transformation Extender product transforms it and routes it to any number of targets where it is needed, providing the appropriate content and format for each target system. For more information about WebSphere Transformation Extender, see WebSphere Transformation Extender product library.

IBM WebSphere Business Events

WebSphere Business Events provides a way to manage business events flowing across systems and people, with the specific goal of providing timely insight and enabling response. When used in combination with WebSphere Process Server, business users can initiate an action (for example, initiating a business process) in response to a prescribed pattern of events.

IBM WebSphere Industry Content Packs

IBM WebSphere Industry Content Packs deliver WebSphere Business Process Management-based solutions, further solidifying, with the help of industry-standards-based SOA assets, improved time-to-value, consistency, and reuse. These assets, packaged under WebSphere Industry Content Packs, are useful across the lifecycle of WebSphere Business Process Management.

IBM WebSphere Service Registry and Repository Advanced Lifecycle Edition

WebSphere Service Registry and Repository delivers a robust system for storing, accessing, and managing service metadata used in the selection, invocation, management, governance, and reuse of services in an SOA. WebSphere ESB function is contained within WebSphere Process Server and can use metadata from WebSphere Service Registry and Repository to dynamically govern its behavior. In doing so, clients can better model, assemble, deploy, and manage their SOA to achieve their business goals with unprecedented levels of control.

IBM WebSphere MQ File Transfer Edition

WebSphere MQ File Transfer Edition adds file-specific features to the proven WebSphere MQ transport. It delivers a managed file transfer solution that enables the movement of files between IT systems with reliability and minimizes the need for programming.

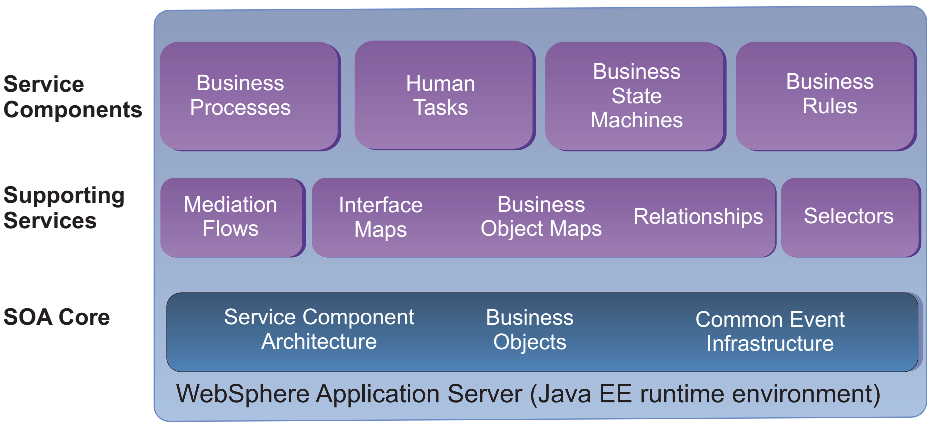
Architectural overview of WebSphere Process Server

WebSphere Process Server is a service-oriented architecture (SOA) integration platform built on a uniform invocation programming model and a uniform data representation model. It provides a standards-based business process engine, using the full power of WebSphere Application Server.

The base runtime infrastructure for WebSphere Process Server is WebSphere Application Server. The Service Component Architecture and business objects that are part of the SOA core provide the uniform invocation and data-representation programming models. The SOA core includes the Common Event Infrastructure for generating events for the monitoring and management of applications running on WebSphere Process Server.

Supporting services provide the foundational business object and transformation framework for WebSphere Process Server. Service components represent the functional components required for composite applications.

The combination of a powerful foundation (WebSphere Application Server and the SOA Core) and service components in WebSphere Process Server allows quick development and deployment of sophisticated composite applications that run on WebSphere Process Server.



One component-based framework addresses all styles of integration.

Figure 1. WebSphere Process Server component-based framework

Related concepts

Service Component Architecture
Service Component Architecture (SCA) enables a service-oriented architecture and is made available by many corporations including IBM. SCA is a platform and vendor independent programming model that provides a simple and consistent means for expressing business logic and business data as SOA services, regardless of technical implementation details. In this section, we examine SCA services and data objects.

Business objects
Data is exchanged between components in an application using business objects. A business object is the primary structure for representing business data in the WebSphere Process Server runtime. The underlying structure of a business object is an XML schema definition (XSD), and programmatic access to business objects is provided using business object interfaces in WebSphere.

Service-oriented architecture core

The service-oriented architecture core of IBM WebSphere Process Server provides both uniform invocation and data-representation programming models and monitoring and management capabilities for applications running on WebSphere Process Server.

Service-oriented architecture (SOA) is a conceptual description of the structure of a software system in terms of its components and the services they provide, without regard for the underlying implementation of these components, services and connections between components. WebSphere Process Server enables deployment of standards-based process integration solutions in an SOA. This means that a well defined set of business-level interfaces for the components can be created and maintained, shielded from lower-level technology changes. Loosely coupled integration applications that are based on SOA provide flexibility and agility. You can implement integration solutions independent of platform, protocols and products. For more information about SOA, refer to the Service-Oriented Architecture (SOA) from IBM Web site.

The Service Component Architecture and business objects that are part of the SOA core provide uniform invocation and data-representation programming models for applications deployed on WebSphere Process Server. The SOA core also includes the Common Event Infrastructure for generating events for the monitoring and management of applications on WebSphere Process Server.

The following diagram shows the WebSphere Process Server component-based framework.

One component-based framework addresses all styles of integration.

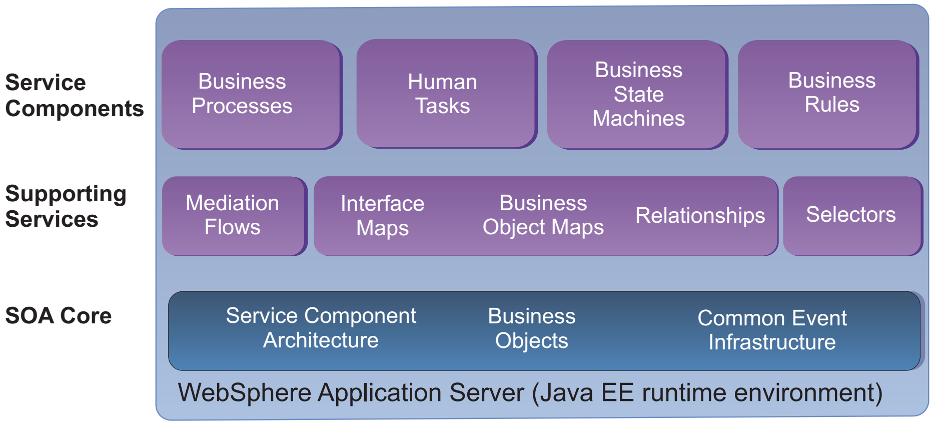


Figure 2. WebSphere Process Server component-based framework

Service Component Architecture

Service Component Architecture presents all elements of business transactions in a service-oriented way in the WebSphere Process Server runtime environment.

Service Component Architecture (SCA) is an architecture in which all elements of a business transaction, such as access to Web services, Enterprise Information System (EIS) service assets, business rules, workflows, databases and so on, are represented in a service-oriented way.

SCA separates business logic from implementation, so that you can focus on assembling an integrated application without knowing implementation details. The implementation of business processes is contained in service components.

Service components can be assembled graphically in the IBM WebSphere Integration Developer tools, and the implementation can be added later. The SCA programming model narrows what developers must know about Java and Java EE or other implementation in particular scenarios to a core set of language concepts that are familiar to all who develop business applications in other programming languages today. This allows developers to quickly and easily integrate technologies.

Developers switching from classical application development environments face a much smaller learning curve; they can quickly become productive with this programming model. The Service Component Architecture programming model also helps experienced Java EE developers be more productive.

Service Component Architecture supports several standard service implementation types:

- Java objects, which implement a Java class. As in the Java programming language, instances of Java components at run time are referred to as Java objects.
- Business process components, which implement a business process. The implementation language is the Business Process Execution Language (BPEL) and its IBM extensions.
- Human task components, which represent and implement a task typically performed by a person in a business process or an integration application.
- Business state machine components, which are used when applications work with artifacts that have a set of states. A state machine defines what the artifacts can do at a point in time.
- Business rule components, which determine the outcome of a business process based on a context and can be designed as if-then rules, decision tables, or decision trees. Business rules within a business process allow applications to respond quickly to changing business conditions. The rules are independent of the business process itself, and you can change them at any time without having to redo your process.

Service qualifiers govern the interaction between a service client and a service on the WebSphere Process Server runtime environment. Service qualifiers are quality of service specifications that define a set of communication characteristics required by an application for transmission priority, level of route reliability, transaction management, and security level. An application communicates its quality of service needs to a runtime environment by specifying service qualifiers. You can specify service qualifiers when wiring components in the assembly editor in WebSphere Integration Developer. These specifications, when running on WebSphere Process Server, determine how the clients interact with the target components. Depending on the qualifiers specified, additional required processing can take place at run time.

Importing and exporting capabilities within the Service Component Architecture define a service module's external interfaces or access points for WebSphere Process Server. Imports and exports can be either to other modules within the same application, or to other applications on enterprise information systems (EIS). This allows working with IBM WebSphere Adapters. For more information about imports and exports, see Service applications.

WebSphere Process Server solutions rely upon the underlying WebSphere Application Server capabilities for transaction, security, and workload management to provide a scalable integration environment.

For business processes, WebSphere Process Server offers support for transactions involving multiple resource managers using the two-phase commit process to ensure atomic, consistent, isolated, and durable (ACID) properties. This capability is available for both short-running flows (single transaction) and long-running flows (multiple transactions). You can group multiple steps in a business process into one transaction by modifying transaction boundaries in WebSphere Integration Developer.

Because not all service invocations support two-phase-commit transactions, WebSphere Process Server also includes recovery capabilities. If a failure occurs in the middle of running an integration application, the server detects it and allows an administrator to manage the failed event from the failed event manager.

Related concepts

Service Component Architecture
Service Component Architecture (SCA) enables a service-oriented architecture and is made available by many corporations including IBM. SCA is a platform and vendor independent programming model that provides a simple and consistent means for expressing business logic and business data as SOA services, regardless of technical implementation details. In this section, we examine SCA services and data objects.

Service Component Architecture programming
Service Component Architecture (SCA) provides a simple, yet powerful programming model for constructing applications based on service-oriented architecture (SOA).

The business object framework

Business objects define the data flowing between components defined in Service Component Architecture.

Part of the IBM WebSphere Application Server capabilities that are built into WebSphere Process Server, business objects provide a framework for data application development that simplifies the Java EE data programming model.

The business object framework, included in WebSphere Process Server as part of the Service-oriented architecture (SOA) core, provides a universal means of describing and exchanging data between Service Component Architecture services – for example, JDBC ResultSet and XML Schema described data.

A *business object* is a set of attributes that represent a business entity (such as Employee), an action on the data (such as a create or update operation), and instructions for processing the data. Components of the integration application use business objects to exchange information and trigger actions. Business objects are flexible because they can represent any kind of data. For example, in addition to supporting the data canonicalization model of traditional integration servers, they also can represent data returned from a synchronous EJB Session Bean facade or a synchronous business process, and then they can be bound to IBM WebSphere Portal portlets and JSF components.

Business objects are the primary mechanism for representing business entities, or documenting literal message definitions, enabling everything from a simple basic object with scalar properties to a large, complex hierarchy or graph of objects.

In WebSphere Process Server, the business object framework is made up of the following elements:

- Business object definition
- Business graph definition

- Business object metadata definition
- Business object services (service APIs)

A business object definition is the name, set of ordered attributes, properties, version number, and application-specific text that specify a type of business object. A business graph definition is the wrapper added around a simple business object or a hierarchy of business objects to provide additional capabilities, such as carrying change summary and event summary information related to the business objects in the business graph. A business object metadata definition is the metadata that can be added to business object definitions to enhance their value when running on WebSphere Process Server. This metadata is added to the business object's XML schema definition as well known xs:annotation and xs:appinfo elements. Business object services are a set of capabilities provided on top of the basic capabilities provided by Service Data Objects. Examples are services such as create, copy, equality, and serialization.

For more information about WebSphere Application Server Service Data Objects, see the WebSphere Application Server information center.

Related concepts

Business objects
Data is exchanged between components in an application using business objects. A business object is the primary structure for representing business data in the WebSphere Process Server runtime. The underlying structure of a business object is an XML schema definition (XSD), and programmatic access to business objects is provided using business object interfaces in WebSphere.

"Business object maps" on page 16
Business object maps are a way of relating business objects.

Business objects programming
Business objects are containers for application data, such as a customer or an invoice. Data is exchanged between components by way of business objects. The underlying structure of a business object is an XML schema definition (XSD), and programmatic access to business objects is provided via business object interfaces in WebSphere. Collectively, these aspects of the business object, its structural representation, its programmatic interfaces, and its behavior and manipulation within the service component architecture (SCA), are the business object framework, which provides a powerful, consistent means for describing and delivering business data in your solution.

Common Event Infrastructure in WebSphere Process Server

The Common Event Infrastructure is an embedded technology within WebSphere Process Server to provide basic event management services.

The infrastructure portion of the Common Event Infrastructure is included as part of the underlying IBM WebSphere Application Server capabilities in WebSphere Process Server. The event emitting capabilities are additional functions of WebSphere Process Server.

The *Common Event Infrastructure (CEI)* is the implementation of a set of APIs and infrastructure for the creation, transmission, persistence, and distribution of business, system, and network Common Base Events. A *Common Base Event* is a specification based on XML that defines a mechanism for managing events – such as logging, tracing, management, and business events – in business enterprise applications.

CEI provides basic event-management services, including consolidating and persisting raw events from multiple, heterogeneous sources and distributing those events to event consumers. It provides functionality for generation, propagation, persistence, and consumption of events representing service component processes. A standard, XML-based format, the Common Base Event model, defines the structure of these events. Each type of event used by the server contains a number of standard fields specific to a given type of event. In some cases, it contains an encapsulation of the business object data that is being used by the service component at a particular event point.

WebSphere Process Server uses events in the CEI almost exclusively to enable service component monitoring. You must configure the CEI server if you want to use event-related functions, but after that, you should not use CEI directly. Instead, use the existing services in WebSphere Process Server.

In WebSphere Process Server, a specially configured CEI Server -- which can be part of an existing process server or another server -- is used for all event-related services. You must first create and deploy several facilities that are used by the CEI Server, including an event database, a messaging engine, one or more enterprise applications, and a database driver.

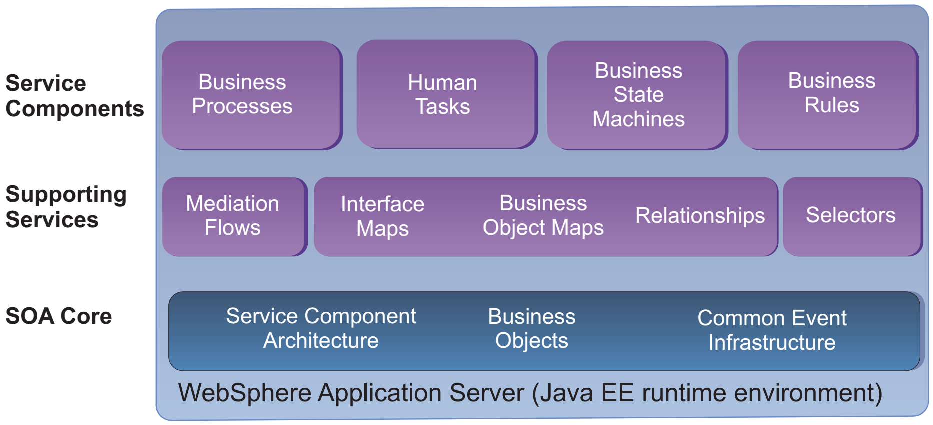
Related information

Administering the Common Event Infrastructure

Supporting services

Supporting services in IBM WebSphere Process Server address a number of transformation challenges for connecting components and external artifacts.

You can use mediation flows, interface maps, business object maps, relationships, and selectors to integrate applications running on the server. With WebSphere Process Server, you also can use business calendars.



One component-based framework addresses all styles of integration.

Figure 3. WebSphere Process Server component-based framework

Mediation flows

Mediation flows intercept and modify messages that are passed between existing services (providers) and clients (requesters) that want to use those services.

A *mediation flow* mediates or intervenes between an export and import to provide functions such as message logging, data transformation, and routing. Mediation flows are created in IBM WebSphere Integration Developer and deployed to WebSphere Process Server in either a module or a mediation module.

Related concepts

Enterprise service bus messaging infrastructure
WebSphere Process Server includes enterprise service bus capabilities. WebSphere Process Server supports the integration of service-oriented, message-oriented, and event-driven technologies to provide a standards-based, messaging infrastructure in an integrated enterprise service bus.

Mediation flow
Mediation is a way of mediating or intervening dynamically between services. A *mediation flow* implements a mediation.

Service message objects
Service message objects (SMOs) provide an abstraction layer for processing and manipulating messages exchanged between services.

Interface maps

Interface maps reconcile the differences between components that have different interfaces.

Note: Interface maps are deprecated as of WebSphere Process Server version 7.0.

Interface maps are supporting service components in WebSphere Process Server that resolve and reconcile differences between interfaces in other Service Component Architecture (SCA) components to enable them to communicate. The interface map captures a first-class pattern that allows module designers in IBM WebSphere Integration Developer to reconcile differences across multiple interfaces using transforms and other rudimentary operations. Interface maps are deployed on WebSphere Process Server as part of modules, also called SCA modules.

Related concepts

Interface map
An *interface map* resolves differences between the interfaces of interacting components.

Transforming data
Flexible and scalable business integration and SOA-based solutions often require support for different types of transformations. When you are integrating services, you often need to transform the data into a format that the receiving service can process. Typically, interfaces and operations of disparate services are not identical, and the message from the source needs to be transformed into a format that can be accepted by the target. Another possibility is that you might want to manipulate the data that you are moving between the source and target.

Business object maps

Business object maps are a way of relating business objects.

Business object maps are supporting service components in IBM WebSphere Process Server that assign values to the target business objects service components based on the values in the source business objects service components. One business object becomes the source and another becomes the target. The business object map maps the source and target. Business object maps support 1-to-n, many-to-1 and many-to-n mappings among business objects. This includes mapping the business data and the aspects associated with the business object, such as verb.

Developers create and edit the business object maps in IBM WebSphere Integration Developer. During run time, the maps resolve how data is represented between the source and target business objects. You can monitor map events during run time in WebSphere Process Server.

Related concepts

Transforming data
Flexible and scalable business integration and SOA-based solutions often require support for different types of transformations. When you are integrating services, you often need to transform the data into a format that the receiving service can process. Typically, interfaces and operations of disparate services are not identical, and the message from the source needs to be transformed into a format that can be accepted by the target. Another possibility is that you might want to manipulate the data that you are moving between the source and target.

"The business object framework" on page 13
Business objects define the data flowing between components defined in Service Component Architecture.

Relationships

Relationships are services used to model and maintain associations between business objects and other data.

Relationships are supporting services in IBM WebSphere Process Server applications that establish an association between data from two or more data types.

A *relationship* is an association between two or more data entities in the business integration system. Most often, these entities are business objects. Relationships are used to transform data that is equivalent across business objects but is represented differently.

In WebSphere Process Server, relationship manager is a tool for manually manipulating relationship data to correct errors found in automated relationship management or to provide more complete relationship information. In particular, it provides a facility for retrieving and modifying relationship instance data. Relationship manager allows you to configure, query, view, and perform operations on relationship runtime data, including participants and their data. You create relationship definitions with relationship designer. At run time, instances of the relationships are populated with the data that associates information from different applications.

Related concepts

Administering relationships
The relationship manager is a tool for manually controlling and manipulating relationship data to correct errors found in automated relationship management or provide more complete relationship information. In particular, it provides a facility for retrieving as well as modifying relationship instance data.

Selectors

Selectors provide flexibility at points in the processing of service components during run time.

Selectors, also called selector components, are supporting services in IBM WebSphere Process Server that take one invocation and allow different targets to be called based on the selection criteria.

A *selector component* is a component that provides a means of interposing a dynamic selection mechanism between the client application and a set of target implementations.

Selectors allow additional flexibility beyond business rules. *Business rules*, a fundamental part of a businesses, drive the general processing of an application, invoking certain services to get the data through the application. For example, a rule may be: Two weeks before school starts, offer a back-to-school special price on our school-related merchandise. A selector takes one invocation and allows different targets to be called based on the selection criteria. For example, if the time is just before school starts, then the previous back-to-school offer would be called. However, if the season is the just as school ends, then a get-your-kids-ready-for-summer offer would be called.

The application is portable because it calls the same thing all the time. The business rule never changes. The actual processing differs (and calls different service components) because of the selector.

Related concepts

Overview of selector components
As businesses change, the business processes that drive them must change, too. Some of those changes may require that certain processes return different results than as originally designed without changing the design of the process. The selector component provides the framework for that flexibility.

Business calendars and timetables

Timetables, also called business calendars, define available time for a business year in an organization, allowing for office hours and holidays. Human tasks and business processes use the available time defined in timetables to schedule timeouts, when a task expires, when it is deleted, or when it is overdue. The Business Calendars widget, available with Business Space, provides an environment for you and all runtime users of the application to view and modify the time intervals, depending on security roles.

A timetable is a schedule of times that indicates availability (such as Monday to Friday). For example, an organization can define its business days and holidays in a timetable that includes the following criteria: working hours of 9:00 a.m. – 5:00 p.m. Monday through Friday, with New Year's Day, Memorial Day, Labor Day, Thanksgiving Day and Christmas Day as holidays.

Timetables are created in WebSphere Business Modeler, and business calendars are created in WebSphere Integration Developer. Both are deployed to WebSphere Process Server as XML artifacts in Service Component Architecture (SCA) modules. Modules containing business calendars and timetables are deployed the same way as other modules – as an enterprise archive (EAR) file – either from WebSphere Integration Developer using the unit test environment or from the administrative console.

The timetable, or business calendar, capability can be used by business processes and human tasks. Business processes use it to schedule timeouts. Human tasks use it to schedule when a task expires or when it will be deleted or is overdue.

At run time, a specified business calendar, or a timetable, is loaded for components that are clients of the service. It uses the current time and the delta to calculate the time for the components. For example, if a claim is identified as overdue if not completed by 3 working days, and the claim is assigned to an employee on a

Friday, May 16, before a public holiday, the process will not be overdue until 6 days later, on Thursday, May 22 – taking into account when the office is closed for Saturday, Sunday, and the public holiday.

During runtime, if you have configured Business Space, you and all users of the applications can use the Business Calendars widget to view and edit timetables. This includes business calendars that were created in WebSphere Integration Developer and timetables created in WebSphere Business Modeler that were deployed to WebSphere Process Server version 6.2. For business calendars developed in WebSphere Integration Developer version 6.1.2, if you want them to be available in the Business Calendars widget, you must import modules into WebSphere Integration Developer version 6.2 and then deploy them to WebSphere Process Server version 6.2.

Each timetable has security roles associated with it: owner, reader, and writer. Users who have the owner role can modify timetables they own in the Business Calendars widget, and they can grant writer and reader roles to other users by using the Security Roles widget in Business Space. Users who have writer roles can modify timetables by creating and modifying time intervals in the Business Calendars widget. Users who have reader roles can view timetables and time intervals but cannot modify them.

The business calendar schema is flexible enough to allow multiple types of timetables. In the flat model, all of the metadata is in one timetable file. In a hierarchical model, you can build small timetables that are complete on their own and then build a top level timetable that references other timetables.

Timetables have dates with offsets based on Greenwich Mean Time (GMT). For example, if a timetable is designated for working hours of 9 a.m. to 5 p.m. in New York, it is set with a GMT offset of GMT-5, which keeps the working hours the same, even if the module is moved to a server in California. In a flat timetable, all the dates use the same offset. In a hierarchical timetable, which references other timetables, the individual timetables can have different GMT offsets.

The scope of a business calendar, or a timetable, is the module into which it is deployed.

For more information about creating business calendars in WebSphere Integration Developer, see "Working with business calendars" in the WebSphere Integration Developer documentation.

For more information about using the Business Calendars widget, see the online help for the Business Calendars widget in Business Space.

Service components

All integration artifacts running on IBM WebSphere Process Server (for example, business processes, business rules, and human tasks) are represented as components with well defined interfaces.

In the Service Component Architecture (SCA), a *service component*, also called an SCA component, defines a service implementation. Service components each have an interface and can be wired together to form a module deployed to WebSphere Process Server.

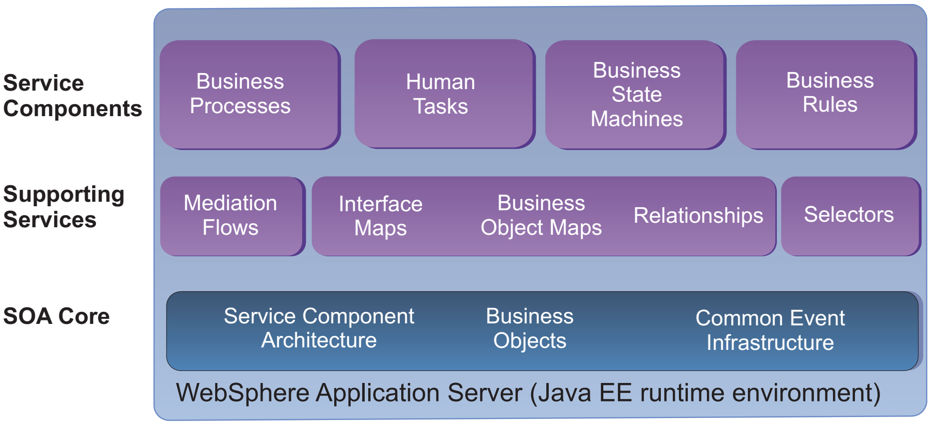
This creates a flexible runtime environment and enables changing any part of an application without affecting the other parts. For example, it is possible to replace a human task representing an approval with a business rule representing automatic approval – by replacing the service components in the assembly diagram – without changing either a business process or the caller of the business process.

Service components can interact with existing applications, using the following programming constructs:

- Java Beans
- Enterprise Java Beans
- Web services
- JMS messages

In addition, service components can interact with other applications on enterprise information systems (EIS) with IBM WebSphere Adapters.

On top of the runtime infrastructure of supporting services and the service-oriented architecture (SOA) core, WebSphere Process Server offers a variety of ready-to-use SCA components that can be used in integration applications. Mediation flows are implemented in an SCA component (a mediation flow component) but for WebSphere Process Server modules they provide a supporting service role.



One component-based framework addresses all styles of integration.

Figure 4. WebSphere Process Server component-based framework

Related concepts

Service components
A service component configures a service implementation. A service component is presented in a standard block diagram.

BPEL process
A *BPEL process* component implements a business process.

Human task
A *human task* component implements a task done by a person. It represents the involvement of a person in a business process.

State machines
A state machine is an alternative way of creating a business process. A state machine is suited for processes related to changing states rather than a flow of control. A state defines what an artifact can do at a point in time. A *state machine* is an implementation of this set of states.

Business rules
Business rules complement business processes and state machines. If there is condition with a variable, for example, a *business rule* can change the value in that variable at run time. Created by a visual programming language, a business rule makes a decision based on context. The decision can be simple or complex. Business rules are nonprocedural and the rules can be changed independently of an application.

Business processes

Business processes are service components that provide the primary means through which enterprise services are integrated.

A *business process* is any system or procedure that an organization uses to achieve a larger business goal. When you break it down, you see that a business process is a series of individual tasks, and each task is executed in a specific order. As an integral part of applications running on IBM WebSphere Process Server, business processes provide the primary means through which enterprise services are integrated.

Business process components implement a fully supported Web Services Business Process Execution Language (BPEL) engine. WebSphere Process Server includes Business Process Choreographer; both are installed on WebSphere Application Server. You can develop and deploy complex business processes in a simple development model with sophisticated support for long and short running business processes in a highly scalable infrastructure. You can either create BPEL models in WebSphere Integration Developer, or import them from a business model you created in WebSphere Business Modeler.

Web Services Business Process Execution Language (BPEL) is used to choreograph the flow of business processes. Business process integration services build on BPEL4WS version 1.1 and add major capabilities of the upcoming WS-BPEL version 2.0 specification.

Human tasks

Human tasks are service components that can be used to either assign work to employees or to invoke other services.

A *human task* is a unit of work done by a human that often involves interaction with other services, and thus becomes a task within a larger business goal.

The Human Task Manager, available in WebSphere Process Server, supports creation and tracking of tasks during run time. Existing LDAP directories (as well as operating system repositories and the WebSphere user registry) can be used to access user and group information. WebSphere Process Server supports multi-level escalation for human tasks including e-mail notification. It also includes a Web client to manage human tasks, and a set of Java Server Faces (JSF) components that can be used to create custom clients or to embed human task functionality into other Web applications.

Human task service components allow role-based task assignment, invocation and escalation.

Business state machines

Business state machines are service components that allow you to represent business processes based on states and events instead of a sequential business process model.

Business state machines specify the sequences of states, responses, and actions that an object or an interaction goes through in response to events.

You create and edit business state machines in IBM WebSphere Integration Developer, and you monitor them during run time in IBM WebSphere Process Server.

Business rules

Business rules are service components that declare policy or conditions that must be satisfied within your business.

A *business rule* is a representation of how business policies or practices apply to a business activity. It is anything that controls the behavior of, or imposes a structure on a business practice. A rule can enforce business policy, establish common guidelines within an organization, or control access in a business environment.

Business rules make business processes more flexible. Because business rules determine the outcome of a process based on a context, using business rules within a business process allows applications to respond quickly to changing business conditions.

Business rule authoring is supported with IBM WebSphere Integration Developer. IBM WebSphere Process Server includes the business rules manager, a Web-based runtime tool for business analysts to update business rules as business needs dictate, without affecting other components or Service Component Architecture (SCA) services.

Deployment environments in WebSphere Process Server

WebSphere Process Server allows you to manage the deployment environment for your Service Component Architecture (SCA) modules as one collection of servers. WebSphere Application Server capabilities included with WebSphere Process Server provide elements for this collection of servers.

The WebSphere Process Server environment includes a layout of interconnected servers, or topology, which supports SCA modules of your service applications.

This topology consists of one server process running on one computer system, or it can consist of multiple server processes running on multiple computer systems. A *server process* is a runtime environment for components that are deployed as SCA modules. In WebSphere products, including WebSphere Process Server, a server process is a Java Virtual Machine (JVM).

If your environment consists of one server process on one system, the server process that is set up is called a *stand-alone server*. A stand-alone server does not have interconnections with other server processes, it has a capacity that is limited to the resources on that one computer system, and it does not include failover support. It is also the easiest environment to set up.

If your environment consists of multiple server processes, you most likely set up those processes as a *clustered* environment in a *cell*. A cell is a management domain of a distributed computing environment consisting of SCA modules and the resources needed to support them. A *deployment environment* is an environment in which server processes, typically on different physical computer systems, are managed together. One deployment manager can manage multiple deployment environments.

Using a deployment environment with clusters provides the following benefits:

- Ease of management: You can have one view for configuring SCA modules, one view of the server processes that support the SCA modules, and one point of control for runtime actions for the SCA modules such as starting, stopping, creating, and deleting.
- Workload balancing: By running application images on multiple servers, a cluster balances an application workload across the servers in the cluster.
- Processing power for the application: You can add processing power to your application by configuring additional server hardware as cluster members supporting the application.
- Application availability: When a server fails, the application continues to process work on the other servers in the cluster thereby allowing recovery efforts to proceed without affecting the application users.
- Maintainability: You can stop a server for planned maintenance without stopping application processing.
- Flexibility: You can add or remove capacity as needed by using the administrative console.

By nature, the z/OS environment is clustered and can provide a highly scalable environment without all the complexities of separate clustered servers. Separate clustered servers can be used for multisystem nodes to provide additional availability.

Related tasks

General steps for implementing a deployment environment
After designing a deployment environment, you will perform specific tasks to make that design a reality. Regardless which method you use to implement the deployment environment, you will perform the same general steps.

Business Space powered by WebSphere

WebSphere Process Server includes Business Space powered by WebSphere, which provides a common interface for application users to create, manage and integrate Web interfaces across the IBM WebSphere Business Process Management portfolio.

Business Space is a browser-based graphical user interface that lets application users customize content from products in the WebSphere Business Process Management portfolio. Application users (business users) are the users of the applications deployed on WebSphere Process Server.

Business Space provides an Asynchronous JavaScript™ and XML (AJAX) interface using mashup technology to enable business users to create and customize human task-centric user experiences. Mashups are Web pages created by combining Web applications (widgets), which mix together Web content to create novel interfaces. The widgets communicate with the WebSphere Process Server runtime using Representational State Transfer (REST) interactions with common Web formats, such as JavaScript Object Notation (JSON) and XML data.

Business users can customize Business Space widgets to view the runtime business data according to their preferences. Business Space administrators can create new spaces (collections of pre-configured pages), and mashup page content, in addition to the predefined scenarios that are shipped with Business Space. For example, these mashups act on specific business needs for the enterprise, such as assigning people to various tasks or adjusting business rules for different outcomes.

Business Space is shipped with WebSphere Process Server, Enterprise Service Bus, WebSphere Business Monitor, and WebSphere Business Compass. Templates for predefined scenarios are enabled in Business Space when each product is installed. Business Space also includes information from WebSphere Business Services Fabric that business users can view and modify.

The following diagram shows the Business Space framework and products in the WebSphere Business Process Management portfolio. The top layer represents Business Space. The middle layer contains products that directly contribute content for Business Space: WebSphere Business Monitor, WebSphere Process Server, WebSphere Enterprise Service Bus, WebSphere Business Compass, and WebSphere Business Services Fabric. The bottom layer represents products that indirectly contribute content through one of the middle-layer products: WebSphere Integration Developer and WebSphere Business Modeler.

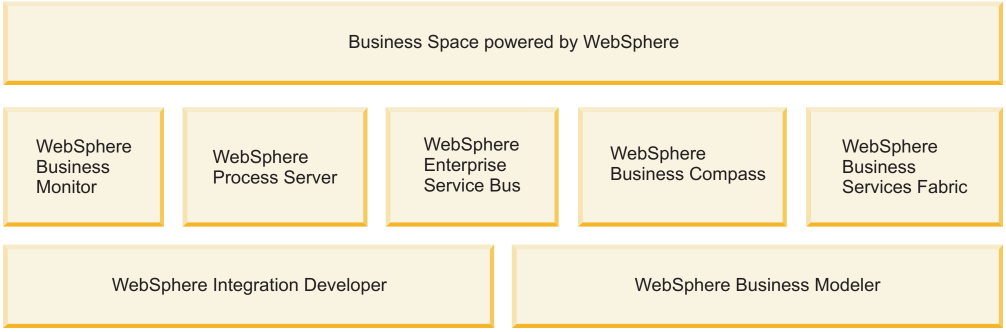

Business Space powered by WebSphere
WebSphere Business Monitor
WebSphere Process Server
WebSphere Enterprise Service Bus
WebSphere Business Compass
WebSphere Business Services Fabric
WebSphere Integration Developer
WebSphere Business Modeler

Figure 5. Business Space and Business Process Management products

Business Space provided with WebSphere Process Server includes templates that you can use to create spaces, or collections of widgets for working with runtime artifacts.

Templates include Advanced Managing of Human Tasks and Workflows, Solution Administration, Service Administration, and Problem Determination. You create a space using templates by clicking **Manage Spaces** → **Create Space**.

Business Space also includes a Business Configuration page with a Business Rules widget to work with business rules.

In addition to the widgets related to WebSphere Process Server applications, Business Space also includes Google Tools widgets and other widgets for viewing presentations, documents, spreadsheets, Web sites, and RSS feeds.

The Business Space framework is installed with WebSphere Process Server. Configure Business Space with the administrative console Business Space Configuration page.

To use the Managing Tasks and Workflows widgets, you must configure Business Process Choreographer. For more information, see "Configuring Business Process Choreographer" in the WebSphere Process Server information center.

After you have installed and configured Business Space, users of your runtime environment can open it from the following URL: http://*host*:*port*/BusinessSpace, where *host* is the name of the host where your server is running and *port* is the port number for your server.

If your team is working with a WebSphere Portal environment, you can configure Business Space for WebSphere Portal. For more information about this configuration, see the related task "Configuring Business Space for WebSphere Portal."

Related concepts

Administration widgets

Human Task Management widgets

Business Space templates

Related tasks

Configuring Business Space

Enabling Business Space widget endpoints on the administrative console

Configuring widgets for WebSphere Portal

WebSphere Adapters

WebSphere Adapters provide a service-oriented approach to integration with Enterprise Information Systems (EIS).

WebSphere Adapters are compliant with Java EE Connector Architecture (JCA 1.5). JCA is the Java EE standard for EIS connectivity. EIS Import and EIS Export provide SCA components with the uniform view of the services external to the module. This allows components to communicate with the variety of external EIS systems using the consistent SCA programming model. WebSphere Adapters are assembled in WebSphere Integration Developer from imported RAR files. They are then exported as an enterprise archive (EAR) file and deployed on WebSphere Process Server.

WebSphere Adapters include the following:

- IBM WebSphere Adapter For Email

- IBM WebSphere Adapter For Flat Files
- IBM WebSphere Adapter For FTP
- IBM WebSphere Adapter for JDBC
- IBM WebSphere Adapter for JD Edwards EnterpriseOne
- IBM WebSphere Adapter for Oracle E-Business Suite
- IBM WebSphere Adapter for Siebel Business Applications
- IBM WebSphere Adapter for SAP Software

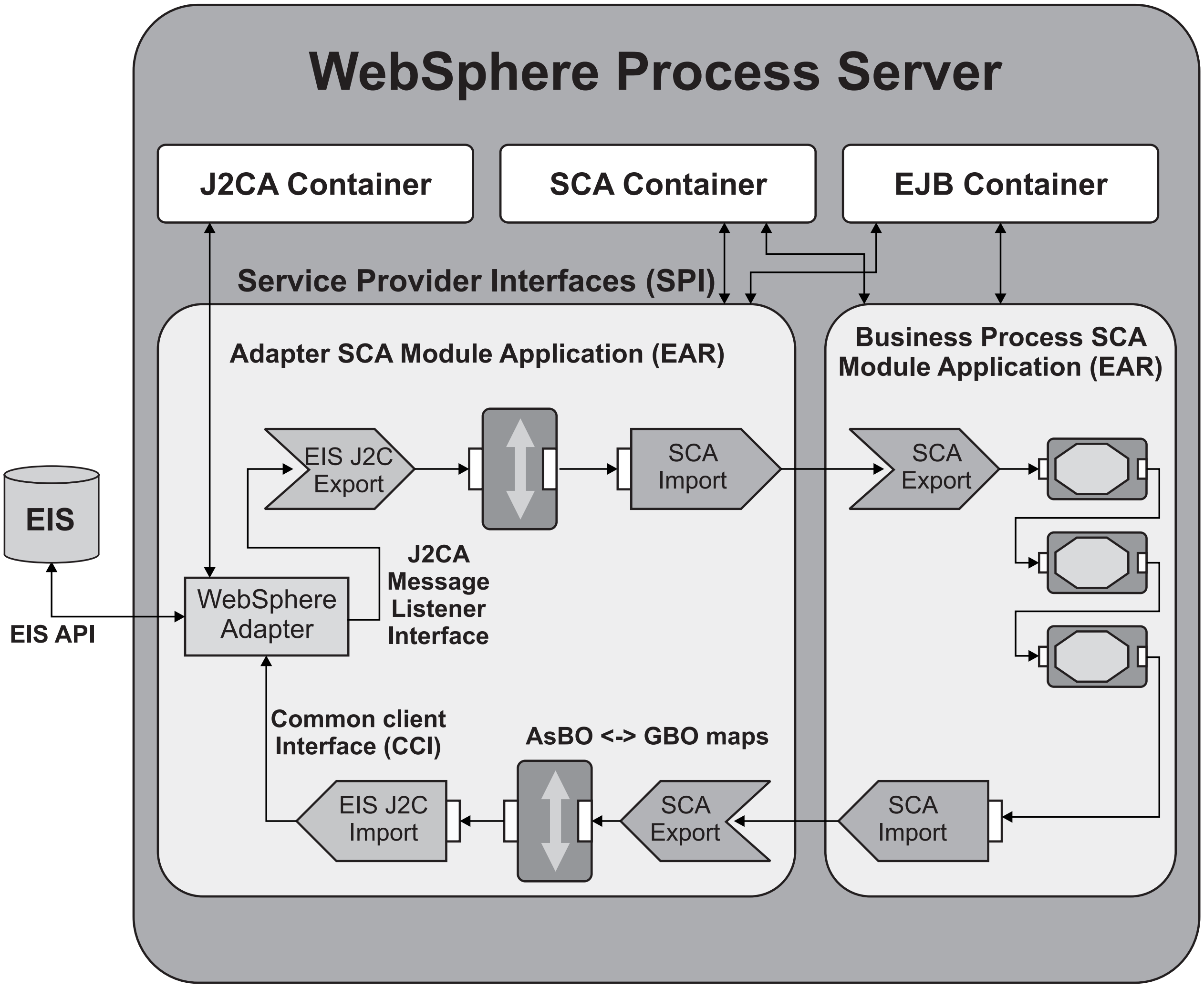


Figure 6. Detailed schematic of a WebSphere Adapter

Development and deployment of applications on WebSphere Process Server

Options for development and deployment of integrated applications on WebSphere Process Server include working in the WebSphere Integration Developer development environment, working with Service Component Architecture APIs, and enabling the applications in a test or production server environment.

Developing service applications

IBM WebSphere Integration Developer is the separate development environment for WebSphere Process Server. In addition to the WebSphere Integration Developer development environment, Service Component Architecture APIs are published for developers. You can also develop some service components using other application development tools then import them into WebSphere Integration Developer for modelling, editing, testing, and packaging for deployment into WebSphere Process Server.

Within WebSphere Integration Developer, you can use an assembly editor to group services into *modules* and specify which services are exposed by the module to outside consumers. The modules are then connected to form complete integration solutions. You encapsulate integration logic within modules so that a change to services within a module will not affect any of the other modules in the solution if the interface of the changed module stays the same.

Modules, also called Service Component Architecture (SCA) modules when deployed to WebSphere Process Server, determine what artifacts are packaged in enterprise archive (EAR) files that are deployed to the runtime environment.

For more information about developing modules for use with WebSphere Process Server, see Developing for WebSphere Process Server.

For more information about using WebSphere Integration Developer to develop integration applications, see the WebSphere Integration Developer documentation.

Deploying service applications

Deploying is the act of enabling your applications – your SCA modules – in either a test or a production environment. While the concept of deployment is the same for both environments, there are a few differences between the deployment task in each environment. Because it is best to test any changes to your SCA modules on a test server before committing them to the production environment, use WebSphere Integration Developer to deploy the modules into a test environment, and to package them as a standard enterprise application package, for deployment into WebSphere Process Server.

Use WebSphere Process Server to install and deploy the applications into a production environment. In WebSphere Process Server, you can use the standard WebSphere administrative console to deploy and manage the components of service integration packages. For more information about deploying applications into WebSphere Process Server, see Deploying modules.

If you need to deploy many application files, which means installing many SCA modules, you may want to use a batch file. For more information about batch files, see “Deploying applications using Apache Ant tasks”.

Related tasks

Deploying a module
You can deploy a module or a mediation module, as generated by WebSphere® Integration Developer, into a production WebSphere Process Server environment using these steps.

Deploying applications using Apache Ant tasks
ANT tasks allow you to define the deployment of multiple applications to WebSphere Process Server and have them run unattended on a server.

Related information

Developing for WebSphere Process Server
Developing (programming) for WebSphere Process Server includes developing business integration solutions and developing modules for a service-oriented programming model. Developing also includes programming client applications for business processes and tasks.

Migration to WebSphere Process Server

In this release, you can migrate your installed applications and profile configurations from prior versions of IBM WebSphere Process Server and IBM WebSphere Enterprise Service Bus to WebSphere Process Server version 7.0 using version-to-version migration tools. Version-to-version migration requires that you install the new version of the product alongside the older product, then run migration tools to migrate the existing applications and configurations to the new product. You cannot use updates (in-place upgrades) to migrate from previous releases to WebSphere Process Server version 7.0.

In addition, you can migrate applications and configuration data from certain IBM products that existed before WebSphere Process Server, such as WebSphere InterChange Server, WebSphere Business Integration Server Express®, WebSphere Studio Application Developer Integration Edition, and WebSphere MQ Workflow.

Related concepts

Migrating: Heritage products
You can migrate applications and configuration data from certain IBM products that existed before WebSphere Process Server.

Related information

Migrating: Version-to-version
With version-to-version migration, applications can be migrated to the new version of the product by redeploying them, by updating the applications using the authoring tools and redeploying them, or by using the runtime migration tools which preserve all the configuration information and automatically redeploy the applications.

Administration of applications on WebSphere Process Server

Administering IBM WebSphere Process Server involves preparing, monitoring, and modifying the environment into which Service Component Architecture (SCA) modules are deployed as applications and resources, and working with the applications and resources themselves.

For more information about administering applications, see the *Administering WebSphere Process Server* PDF file.

WebSphere Process Server offers several interfaces for administering the runtime environment:

- Administrative console

  The *administrative console* is a browser-based interface where you can monitor, update, stop, and start a wide variety of applications, services, and resources for the applications running on WebSphere Process Server. The administrative console can also be used to work with relationships and to locate and resolve failed WebSphere Process Server events.

  The administrative console also provide administration capabilities for WebSphere Application Server and other customer-defined products. The WebSphere Process Server administrative console is part of the Integrated Solutions Console framework in general, and the WebSphere Application Server administrative console in particular. As a result, many administrative tasks (for example, setting security, viewing logs, and installing applications) are the same for both WebSphere Process Server and WebSphere Application Server.
- Command-line tools

  Command-line tools are simple programs that you run from an operating system command-line prompt to perform specific tasks. Using these tools, you can start and stop application servers, check server status, add or remove nodes, and other tasks. The WebSphere Process Server command-line tools include the serviceDeploy command, which processes .jar, .ear, .war and .rar files exported from a WebSphere Integration Developer environment and prepares them for installation to the production server.
- WebSphere administrative (wsadmin) scripting program

  The wsadmin scripting program is a non-graphical command interpreter environment that enables you to run administrative options in a scripting language and to submit scripting language programs for execution. It supports the same tasks as the administrative console. The wsadmin tool is intended for production environments and unattended operations.
- Administrative programs

  A set of Java classes and methods under the Java Management Extensions (JMX) specification provide support for administering Service Component Architecture (SCA) and business objects. Each programming interface includes a description of its purpose, an example that demonstrates how to use the interface or class, and references to the individual method descriptions.
- Business Process Choreographer Explorer

  Business Process Choreographer Explorer is a stand-alone Web application that provides a basic set of administration functions for managing business processes and human tasks. You can view information about process templates, process instances, task instances, and their associated objects. You can also act on these objects; for example, you can start new process instances, repair and restart failed activities, manage work items, and delete completed process instances and task instances.

  Business Process Choreographer Explorer also includes an optional reporting function. You can use Process Choreographer Explorer reporting function to create reports on processes that have been completed. You can then use these reports to evaluate the effectiveness and reliability of your processes and activities. You can also use the reporting function to view the status of running processes.
- Business rules manager

The business rules manager is a Web-based tool that assists the business analyst in browsing and modifying business rule values. The tool is an option of WebSphere Process Server that you can select to install at profile creation time or after the initial installation of the server.

Related concepts

The administrative console
The administrative console is a browser-based interface used to administer applications, services, and other resources at a cell, node, server, or cluster scope. You can use the console with stand-alone servers and with deployment managers that manage all servers in a cell in a networked environment.

Related information

Administering WebSphere Process Server
Administering WebSphere Process Server involves preparing, monitoring, and modifying the runtime environment into which applications, their modules, and resources are deployed, and managing those applications, modules, and resources within the runtime environment.

Administrative control of mediation processing

You can control mediation flows between service requesters and service providers, administratively.

You can control mediation flows by changing module properties. The module properties set the values of mediation primitive properties.

Module properties

You can change the properties of Service Component Architecture (SCA) modules that contain mediation flows. You can make changes in the following ways:

- When you install an application.
  - From the administrative console
  - Using an administrative command
- When you administer an application.
  - From the administrative console
  - Using an administrative command

The properties that you can change are those that have been promoted from WebSphere Integration Developer. Any property that you promote is also a dynamic property (which means that it can be overridden at run time using a mediation policy).

The module properties displayed on the administrative console, can do the following:

- Change the values of properties in a mediation flow.
- Provide default values for mediation flows that use mediation policies. (Promoted property values are used when there are no suitable mediation policy values. For further information, see the mediation policy model.)

Promoted properties always have a name, a type, and a value; you can change the value, administratively.

In addition, promoted properties can belong to a group (property groups are introduced in Version 6.2). Property groups can do the following:

- Separate multiple properties that have the same name. An administrator sets property values inside groups. You might have one group for properties on the request flow, and another group for properties on the response flow.
- Set multiple properties (of the same type) under one name. If the integration developer promotes two properties with the same alias name and group, then the administrator can set their values together. You might have logging for the request flow and the response flow, and want to set both at the same time.
- Map to a namespace in a mediation policy.

Security on WebSphere Process Server

IBM WebSphere Process Server provides a runtime security infrastructure and mechanisms based on IBM WebSphere Application Server security.

Securing the WebSphere Process Server environment involves enabling administrative security, enabling application security, creating profiles with security, and restricting access to critical functions to selected users.

Related information

Securing WebSphere Process Server and applications
Security of WebSphere Process Server and applications depends on securing the runtime environment and securing applications.

System monitoring on WebSphere Process Server

You monitor events in WebSphere Process Server to assess problem determination, to tune performance, and to measure the effectiveness of your business processes.

WebSphere Process Server event monitoring capabilities include performance monitoring and service component monitoring.

Monitoring performance: Performance measurements are available for service component event points, and are processed through the Performance Monitoring Infrastructure (PMI) and the Tivoli® Performance Viewer.

You can monitor specific performance measurements for a given event, such as the number of times the event is invoked or the length of time it takes for that event to complete from start to finish. You can also monitor events, and later view their contents, either by viewing the events in a log file or by querying the events stored on the event database. In both cases, you can temporarily specify an event point or points to monitor in order to spot problems with the application logic or with system performance.

Monitoring service component events: WebSphere Process Server monitoring can capture the data in a service component at a certain event point. These events are formatted in a standard called the Common Base Event. You can have the process server publish these events to the logging facilities, or you can use the more versatile monitoring capabilities of a Common Event Infrastructure server database to store and analyze these events.

Some applications that run on the process server include event points that are monitored continually after the application is deployed. You can do this if you are a business analyst and want to observe the effectiveness of the business processes you have modeled and implemented in the applications you deployed on the process server. This allows you to use products, such as IBM WebSphere Business Monitor, to create customized panels -- or "dashboards" -- to view key business process metrics.

Related information

Monitoring WebSphere Process Server
Monitoring WebSphere Process Server enables you to assess performance and evaluate the overall progress of service components that make up the applications deployed on your system.

Samples

Samples help you learn how to accomplish your goals with WebSphere Process Server.

WebSphere Process Server samples are available from the Samples Gallery, which you can install with the product.

WebSphere Process Server samples are also included in the Business Process Management samples at http://publib.boulder.ibm.com/bpcsamp/index.html.

Installing and accessing the Samples Gallery

Samples of integration application artifacts are available in the Samples Gallery.

About this task

The Samples Gallery contains samples of simple artifacts such as those generated by IBM WebSphere Integration Developer and deployed on IBM WebSphere Process Server. Other Business Process Management samples are available at http://publib.boulder.ibm.com/bpcsamp/index.html.

To install and view the WebSphere Process Server Samples Gallery, perform the following steps.

Procedure

1. Unload the contents of the WebSphere Process Server for z/OS installation media. The samples package is loaded onto the system when you unload the installation media. Run the install script to create the symlinks to the files in the samples directory

   The samples are installed in the `*install_root*/samples/bin` directory.
2. Start the server.
3. The Samples Gallery can be accessed directly from a web browser with the link `http://xxxxxxxx:9080/WESBsamples/en/index.html`, where xxxxxxxx represents a host name or IP address of the target z/OS system and 9080 is the port name assigned for the default host. The actions of deploying and installing each sample into the server is part of the samples themselves. WebSphere Process Server samples are initially listed as installable samples in the Samples Gallery.
4. Start the Samples Gallery and click **Refresh**. You can deploy any of the installable samples by following the instructions that appear in the browser window. The samples that you previously deployed are listed as installed samples that you can run by selecting an option in the browser window.
5. Run each of the z/OS deployed samples by following the instructions on the screen.

Business Process Management samples

Business Process Management samples demonstrate features developed in IBM WebSphere Integration Developer and deployed on IBM WebSphere Process Server. They help you work with various product features to develop your own applications.

Business Process Management samples are available at http://publib.boulder.ibm.com/bpcsamp/index.html.

Standards compliance

WebSphere Process Server is compliant with several government and industry standards, including accessibility standards, information processing standards, software download security standards, and internet protocol standards.

Accessibility

IBM strives to provide products with usable access for everyone, regardless of age or ability.

Accessibility features for WebSphere Process Server

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility features

The following list includes the major accessibility features in WebSphere Process Server. The accessibility features include the following functions:

- Keyboard-only operation, except in Business Space powered by WebSphere.
- Interfaces that are commonly used by screen readers.

Operating system features that support accessibility are available when you are using WebSphere Process Server.

Tip: The WebSphere Process Server Information Center is accessibility-enabled for screen reader software, including IBM Home Page Reader. You can operate all features using the keyboard instead of the mouse.

Keyboard navigation

This product uses standard Web browser navigation keys.

(For information about supported Web browsers, see the WebSphere Process Server System Requirements at http://www.ibm.com/software/integration/wps/sysreqs/.)

Interface information

- Installation

  You install WebSphere Process Server for z/OS using scripts, and you can install the WebSphere Process Server Client silently from the command line. This silent installation meets accessibility needs..
- Administration

The administrative console is the primary interface for interacting with the product. This console is displayed within a standard Web browser. By using an accessible Web browser, such as Microsoft Internet Explorer, administrators are able to:

- Use screen-reader software and a digital speech synthesizer to hear what is displayed on the screen
- Use voice recognition software, such as IBM ViaVoice®, to enter data and to navigate the user interface
- Operate features by using the keyboard instead of the mouse

You can configure and administer product features by using standard text editors and scripted or command-line interfaces instead of the graphical interfaces that are provided.

When appropriate, the documentation for specific product features contains additional information about the accessibility of the features.

Vendor software

This product includes certain third-party software not covered under the IBM license agreement. IBM makes no representation about status of these products regarding the Section 508 of the U.S. Federal Rehabilitation. Contact the vendor for information about the Section 508 status of its products. You can request a U.S. Section 508 Voluntary Product Accessibility Template (VPAT) on the IBM Product accessibility information Web page at www.ibm.com/able/product_accessibility.

Related accessibility information

See the IBM Accessibility Center for more information about the commitment that IBM has to accessibility.

Federal Information Processing Standards

Federal Information Processing Standards (FIPS) are standards and guidelines issued by the National Institute of Standards and Technology (NIST) for federal government computer systems.

WebSphere Process Server relies on IBM WebSphere Application Server for all cryptographic functions, which are compliant with Federal Information Processing Standards.

FIPS are developed when there are compelling federal government requirements for standards, such as for security and interoperability, but acceptable industry standards or solutions do not exist. Government agencies and financial institutions use these standards to ensure that the products conform to specified security requirements. For more information about these standards, see the National Institute of Standards and Technology at http://www.nist.gov/.

WebSphere Application Server integrates cryptographic modules including Java Secure Socket Extension (JSSE) and Java Cryptography Extension (JCE), which have undergone FIPS 140-2 certification. In the WebSphere Application Server documentation, the IBM JSSE and JCE modules that have undergone FIPS certification are referred to as IBMJSSEFIPS and IBMJCEFIPS.

For more information, see "Configuring Federal Information Processing Standard Java Secure Socket Extension files" in the WebSphere Application Server information center. When you enable FIPS, several components of the server are affected including the cipher suites, the cryptographic providers, the load balancer, the caching proxy, the high availability manager, and the data replication service.

Related information

Configuring Federal Information Processing Standard Java Secure Socket Extension files

Internet Protocol Version 6

WebSphere Process Server relies on WebSphere Application Server for all Internet Protocol Version 6 compatibility.

IBM WebSphere Application Server Version 7.0 and its JavaMail component support Internet Protocol Version 6 (IPv6).

For more information about this compatibility in WebSphere Application Server, see "JavaMail support for Internet Protocol 6.0" in the WebSphere Application Server documentation.

For more information about IPv6, see www.ipv6.org.

Related information

JavaMail support for Internet Protocol 6.0

www.ipv6.org

Globalization

WebSphere Process Server is globalized: It has multicultural support, and the user interface and documentation are translated into multiple languages.

Multicultural support means that WebSphere Process Server supports the cultural conventions of multiple languages and geographic regions. These conventions include the use of various writing systems and sort orders; various formats for date, time, numbers, and currency; and various keyboard layouts.

Translations are provided for the following national languages:

- Brazilian Portuguese
- Czech
- French
- German
- Hungarian
- Italian
- Japanese
- Korean
- Polish
- Russian
- Simplified Chinese (GB18030 compliant)
- Spanish
- Traditional Chinese

WebSphere Process Server provides partial translations (Human Task Manager and Business Process Choreographer Explorer) for the following national languages:

- Arabic
- Hebrew

Because WebSphere Process Server is built on WebSphere Application Server, you can get information about developing and assembling globalized applications in the WebSphere Application Server information center. See "Learn about WebSphere programming extensions" in the WebSphere Application Server documentation.

Bidirectional language support

WebSphere Process Server supports bidirectional languages, through bidirectional enablement. Bidirectional enablement is a mechanism for accurately displaying and processing bidirectional script data inside components either bundled with WebSphere Process Server (for example, Web-based tools such as the Common Base Event Browser or the business rules manager) or supported by it (for example, service components).

WebSphere Process Server processes all bidirectional language data in the logical, left-to-right format, which is the Windows standard bidirectional language format. It processes data passed to internal components, stores data, and outputs the data in that format. WebSphere Adapters and other Enterprise Information Systems (EIS), must convert the data into this format before sending the data to be processed by WebSphere Process Server. Because the data output by WebSphere Process Server is also in the logical, left-to-right format, the receiving application must convert it to the correct bidirectional format required by the external EIS.

The following table shows the attributes and settings that must match the Windows standard bidirectional format.

Table 1. Bidirectional language format string values

Letter position	Purpose	Allowable values	Default value	Meaning
1	Ordering schema	I	I	Implicit
		V		Visual
2	Orientation	L	L	Left to right
		R		Right to left
		C		Contextual left to right
		D		Contextual right to left
3	Symmetric swapping	Y	Y	Symmetrical swapping is on
		N		Symmetrical swapping is off
4	Shaping	S	N	Text is shaped
		N		Text is not shaped
		I		Initial shaping
		M		Middle shaping
		F		Final shaping
		B		Isolated shaping

Table 1. Bidirectional language format string values (continued)

Letter position	Purpose	Allowable values	Default value	Meaning
5	Numeric	H	N	Hindi (National)
		C		Contextual
		N		Nominal

For data coming from an external component that does not enforce bidirectional support, such as Web services or connectors that are not enabled for processing bidirectional data, you can use example bidirectional application programming interfaces (APIs), based on the IBM Java Development Kit (JDK) to create APIs that transform the data from an external source into the supported bidirectional language format and that transform data sent from WebSphere Process Server to an external EIS into the bidirectional format used by that specific EIS.

To create APIs that transform string objects, see "Transforming string objects from one bidirectional language format to another."

To create APIs that transform data objects, see "Transforming data objects from one bidirectional language format to another."

Note: The locale setting of the user interface (browser) defines the bidirectional language display and edit format.

For more information about bidirectional language, see the technical articles on IBM developerWorks, available at www.ibm.com/developerworks/websphere/library/techarticles/bidi/bidigen.html.

Related tasks

"Transforming string objects from one bidirectional language format to another"
For data coming from an external Enterprise Information System (EIS), you can create APIs that transform string data into the supported bidirectional language format and that transform data sent from WebSphere Process Server to an external EIS into the bidirectional format used by that specific EIS.

"Transforming data objects from one bidirectional language format to another" on page 38
For data coming from an external Enterprise Information System (EIS), you can create APIs that transform Service Data Objects into the supported bidirectional language format and that transform data sent from WebSphere Process Server to an external EIS into the bidirectional format used by that specific EIS.

Related information

Learn about WebSphere programming extensions

www.ibm.com/developerworks/websphere/library/techarticles/bidi/bidigen.html

Transforming string objects from one bidirectional language format to another

For data coming from an external Enterprise Information System (EIS), you can create APIs that transform string data into the supported bidirectional language format and that transform data sent from WebSphere Process Server to an external EIS into the bidirectional format used by that specific EIS.

Before you begin

For more information about bidirectional language support, see Globalization. Use the table in Globalization to determine the correct value for either the input string or output string to use when transforming string data from one format to another.

To create an API to transform the bidirectional language format of string objects, perform the following steps.

Procedure

1. Include all bidirectional classes that contain the bidirectional engine implementation. For example:

   ```
   import com.ibm.bidiTools.bdlayout.*;
   ```

2. Define the strings to contain the data object to transform, and the input and output format values.

   The input format is the bidirectional format in which the string object is currently stored. The output format is the bidirectional format in which you want to store the string object. For example:

   ```
   String strIn = new String("Hello world");
   String formatIn = "ILYNN";
   String formatOut = "VLYNN";
   ```

3. Call the BidiStringTransformation function. For example:

   ```
   String strOut = BiDiStringTransformation(strIn, formatIn, formatOut);
   ```

   ```
   String BiDiStringTransformation(String strIn, String formatIn, String formatOut) {
   ```

   a. Test if the input string is null. For example:

      ```
      if (strIn == null) return null;
      ```

   b. Perform the transformation. For example:

      ```
      BidiFlagSet flagsIn;
      BidiFlagSet flagsOut;
      formatIn = formatIn.toUpperCase();
      formatOut = formatOut.toUpperCase();

      if (formatIn != null)
          flagsIn = new BidiFlagSet(formatIn.toCharArray());
      else
          flagsIn = new BidiFlagSet();

      if (formatOut != null)
          flagsOut = new BidiFlagSet(formatOut.toCharArray());
      else
          flagsOut = new BidiFlagSet();

      if (flagsIn.equals(flagsOut)) return strIn;
      String strOut = BiDiStringTransformation(strIn, flagsIn, flagsOut);
      return strOut;
    }
      ```

Related concepts

"Globalization" on page 35
WebSphere Process Server is globalized: It has multicultural support, and the user interface and documentation are translated into multiple languages.

Transforming data objects from one bidirectional language format to another

For data coming from an external Enterprise Information System (EIS), you can create APIs that transform Service Data Objects into the supported bidirectional language format and that transform data sent from WebSphere Process Server to an external EIS into the bidirectional format used by that specific EIS.

Before you begin

For more information about bidirectional language support, see Globalization. Use the table in Globalization to determine the correct value for either the input string or output string to use when transforming DataObject-type data from one format to another.

To create an API to transform the bidirectional language format of data objects, perform the following steps.

Procedure

1. Include all bidirectional classes that contain the bidirectional engine implementation. For example:

   ```
   import com.ibm.bidiTools.bdlayout.*;
   ```

2. Include all the classes you need to manipulate the DataObject-type object. For example:

   ```
   import commonj.sdo.DataObject;
   import commonj.sdo.Type;
   import commonj.sdo.Property;
   ```

3. Define string variables to contain the different types of strings that a DataObject-type object contains. This filters the attributes of `type String` while recursively transversing the DataObject. For example:

   ```
   String STRING_STR_TYPE = "String";
   String NORM_STRING_STR_TYPE = "normalizedString";
   String TOKEN_STR_TYPE = "token";
   String LANG_STR_TYPE = "language";
   String NAME_STR_TYPE = "Name";
   String NMTOKEN_STR_TYPE = "NMTOKEN";
   String NCNANE_STR_TYPE = "NCName";
   String ID_STR_TYPE = "ID";
   String IDREF_STR_TYPE = "IDREF";
   String IDREFS_STR_TYPE = "IDREFS";
   String ENTITY_STR_TYPE = "ENTITY";
   String ENTITIES_STR_TYPE = "ENTITIES";
   ```

4. Define the function that verifies if the type of a property is String. For example:

   ```
   private static boolean isStringFamilyType (Property property) {
       boolean rc = false;
       if ((property.getType().getName().equalsIgnoreCase(STRING_STR_TYPE)) ||
       (property.getType().getName().equalsIgnoreCase(NORM_STRING_STR_TYPE)) ||
       (property.getType().getName().equalsIgnoreCase(TOKEN_STR_TYPE)) ||
       (property.getType().getName().equalsIgnoreCase(LANG_STR_TYPE)) ||
       (property.getType().getName().equalsIgnoreCase(NAME_STR_TYPE)) ||
       (property.getType().getName().equalsIgnoreCase(NMTOKEN_STR_TYPE)) ||
       (property.getType().getName().equalsIgnoreCase(NCNANE_STR_TYPE)) ||
       (property.getType().getName().equalsIgnoreCase(ID_STR_TYPE)) ||
       (property.getType().getName().equalsIgnoreCase(IDREF_STR_TYPE)) ||
       (property.getType().getName().equalsIgnoreCase(IDREFS_STR_TYPE)) ||
       (property.getType().getName().equalsIgnoreCase(ENTITY_STR_TYPE)) ||
       (property.getType().getName().equalsIgnoreCase(ENTITIES_STR_TYPE)))
          rc = true;
        return rc;
   }
   ```

5. Define the recursive function that applies the bidirectional transformation on the entire DataObject.

   Note: The code logic includes the following assumptions:

- Bidirectional transformation is applied on the properties of string-type only.
- The properties of type string in the DataObject are stored in one bidirectional format.

For example:

```
DataObject BiDiDataObjTransformationBO(DataObject boIn, String formatIn, String formatOut){

   Type type;
   Property property;

   if (boIn == null) return null;

   type = boIn.getType();
   List propertyList = type.getProperties();
   for (int propertyNumber = 0; propertyNumber < propertyList.size(); propertyNumber++){
         property = (Property) propertyList.get(propertyNumber);
         String propertyName = property.getName();
```

a. Skip all non-string properties. For example:

```
if (!isStringFamilyType(property))
      continue;

if (property.isContainment()) {
     if (property.isMany()) {
           List childsList = boIn.getList(property);
```

b. Recursively call the transformation to handle child objects. For example:

```
              for (int childNumber = 0; childNumber < childsList.size();
  childNumber++){                 BiDiDataObjTransformationBO(connectionContext,
((DataObject)childsList.get(childNumber)),formatIn, formatOut);
          }
      } else {
```

c. Recursively call the transformation to handle child objects of any contained business objects. For example:

```
         BiDiDataObjTransformationBO(connectionContext,
 ((DataObject)boIn.get(property)),formatIn, formatOut);
    }
} else {
```

d. Transform the simple string attributes. For example:

```
         String str = BiDiStringTransformation(
             (boIn.getString(propertyName),formatIn, formatOut);
         boIn.setString(propertyName, str);
     }
   }
   return boIn;
}
```

Related concepts

"Globalization" on page 35
WebSphere Process Server is globalized: It has multicultural support, and the user interface and documentation are translated into multiple languages.

Printed in USA