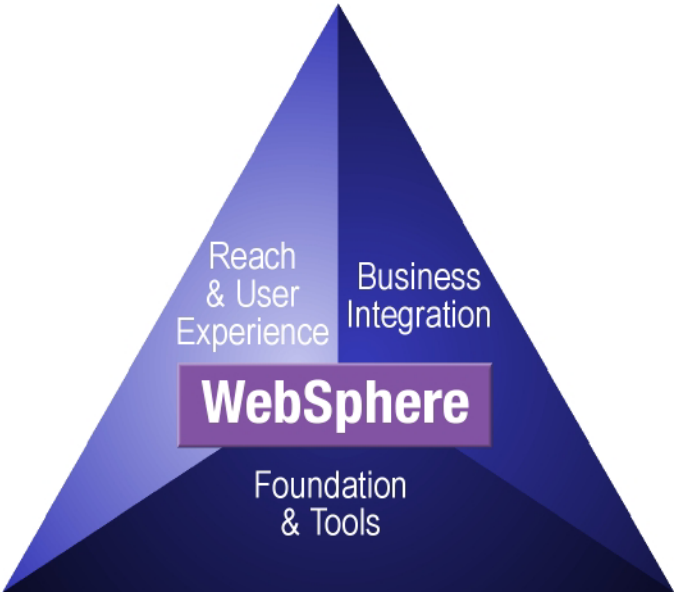
World Class Infrastructure for On Demand Business

WebSphere Application Server, Version 5



Comprehensive "Build-to-Integrate" Platform

Almost all new applications require integration with other applications and existing enterprise systems. Specific integration requirements include:

- Integrating and reusing existing IT assets within a dynamic e-business infrastructure
- Increasing developer productivity while composing integrated application
- Creating flexible applications that allow easy updates in the future
- Combining synchronous application server services with asynchronous messaging

Services Oriented Architecture

Version 5 of WebSphere Application Server and Studio fully implement and leverage a services oriented architecture based on Web services and J2EE standards. Because *WebSphere Application Server, V5* features an open approach to transforming any application asset into a modular service, these services are then accessible for reuse by other developers throughout the organization. WebSphere provides visual tools for developing business objects, business logic, and integration logic which are all compatible, using one repository for metadata and a single approach for defining data handling and transformation.

A services oriented architecture not only aids internal development and integration but also provides a standardized method for publishing encapsulated business services in order to drive B2B collaboration and new revenue streams. Examples Web services include stock quotes and charting, credit card verification and payment processing, integrated travel planning, and auctions.

WebSphere Application Server, V5 provides leading Web services capabilities including:

- Support for the latest Web services standards including SOAP 1.1[1], and access and security updates, all natively integrated with the J2EE programming model. SOAP is widely viewed as the backbone to a new generation of cross-platform; cross-language distributed computing applications, or Web services. The Apache SOAP architecture, which is incorporated in *WebSphere Application Server, Version 5*, defines a set of stable, published interfaces for component-oriented deployment.
- Intuitive interfaces for utilizing the latest Web services standards like SOAP (Simple Object Access Protocol) and WSDL (Web Services Description Language).
- Information connectivity via pre-built and tailored application adapters using the latest Web services standards for generation and service composition.

[1] SOAP allows for easy, standardized access to public and private registries and other Web services applications. SOAP is the Web services version of RPC (Remote Procedure Call), an XML-based protocol and encoding format for inter-application communications

- Web services-based invocation framework providing protocol flexibility and easy-to-use tools that generate Web services applications.

- UDDI (Universal Description, Discovery, and Integration) registry for use within the enterprise or between trusted parties. The UDDI registry provides a secure dynamic mechanism by which services can be published and consumed. WebSphere also supports use of the Public UDDI registries.[2]

- Support for Web services gateways. These gateways offer more security and protection by filtering Web services access to registries and other applications as suitable for B2B solutions.

WebSphere's services oriented architecture provides critical flexibility. Specific features in the application server products build on this foundation to meet customers' demanding integration requirements.

Integrated Connector Architecture

New Java-based applications frequently need to integrate existing legacy assets with new business logic written in Java. The ability to create new applications which incorporate a variety of enterprise resources quickly and easily is a key requirement across a variety of industries. Businesses are looking for new ways to reduce the amount of complex underlying coding required to create these dynamic applications without sacrificing transactional integrity. For example, what if an insurance or telecommunications company which stores a multitude of customer data utilizing IBM CICS now needs to integrate this information within their J2EE environment. How does it efficiently accomplish this task?

WebSphere Application Server, V5 delivers a productive environment for visually creating dynamic application adapters that can be easily integrated with others within complex, multi-mode transactional schemes. WebSphere Application Server leverages the J2EE Connector Architecture (JCA) which provides a consistent way of connecting to and communicating with a wide range of enterprise systems and applications, and extends it to provide advanced transaction coordination..

The J2EE Connector Architecture defines function that *WebSphere Application Server* provides and which back-end system vendors (e.g., IBM CrossWorlds, SAP, PeopleSoft, Siebel, Oracle, and/or third-party connector developers) can use to plug into J2EE.

JCA defines a set of service contracts that a connector developer can expect will be available to the adapter at application runtime. The three services defined in 1.0 and implemented in WebSphere *WebSphere Application Server, V5* include:

- **Connection management** enables *WebSphere Application Server* to create and manage connections to back-end systems. *WebSphere Application Server* also implements connection pooling, since connections to back-end systems are expensive.

- **Transaction management** supports transactional access to underlying resource managers. This service enables the transaction manager provided within the EJB server to manage transactions across multiple back-end systems.

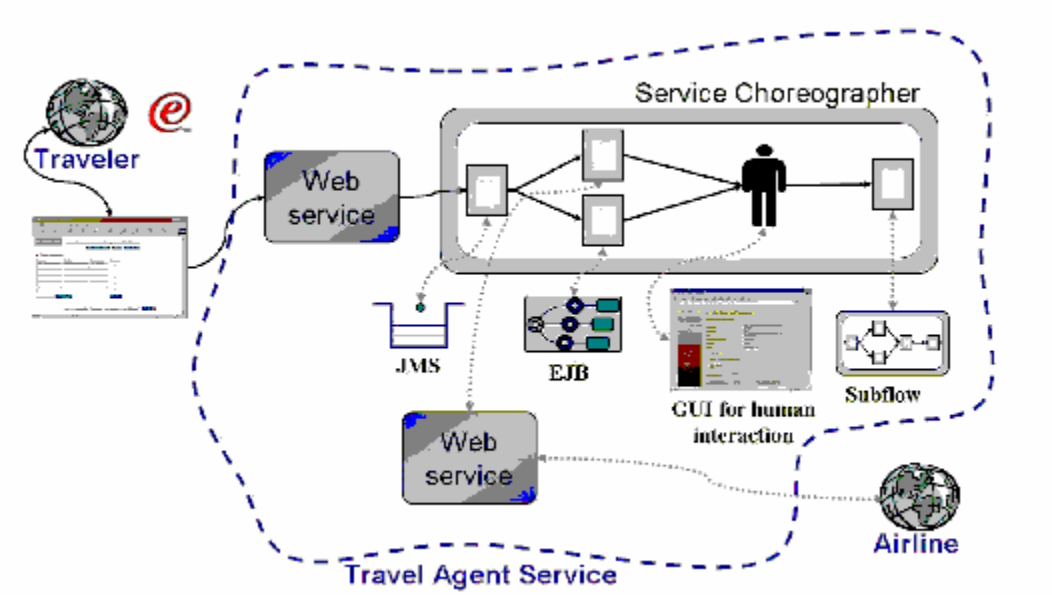
[2] Public registries enable anyone to publish their services and for other, effectively anonymous users to locate them and understand how to use them. Normal access restrictions will still apply from a technical and business perspective. These are operated currently by IBM and Microsoft®, and will be shortly joined by HP and SAP.

- **Security management** enables the developer to define security between the EJB server and the back-end system. The specific security mechanism that is used is dependent on the security mechanism provided by the back-end system.

Becauase *WebSphere Application Server,* V5 implements a services oriented architecture and sufaces that architecture to developers through a visual tool, connectors can be easily accessed and integrated into workflows via the visual WebSphere Studio integration perspective.

Compose long-running workflows

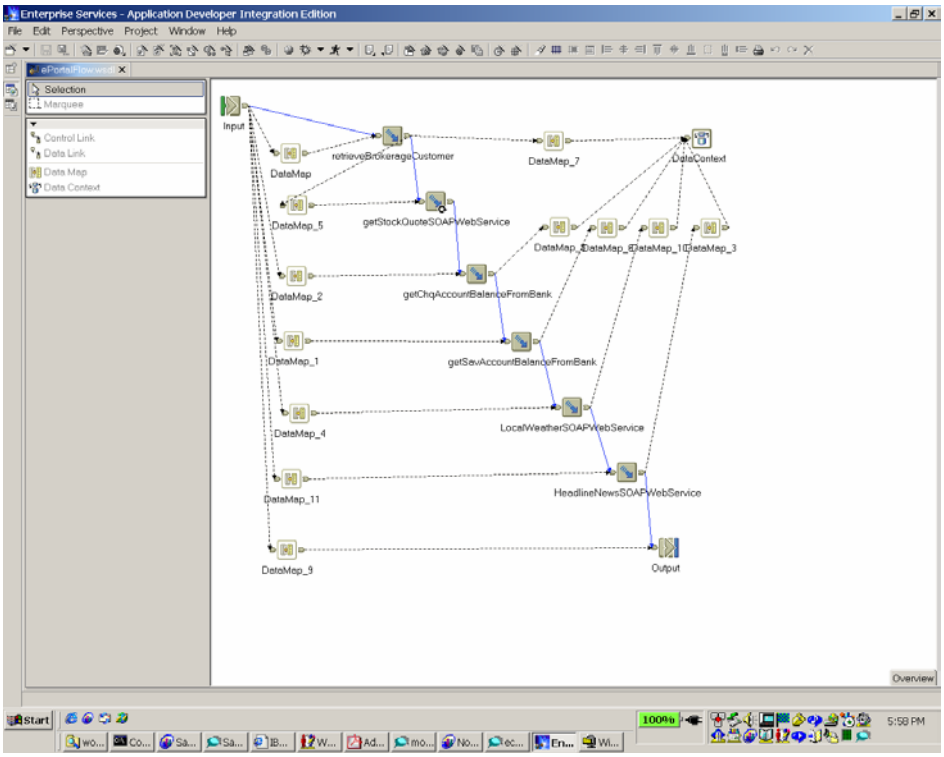
WebSphere Application Server, V5 provides the ability to integrate Web services into long- running workflows which leverage new and existing applications. Being able to easily choreograph application interactions allows for real-time adjustments in the future. With new *WebSphere Application Server, V5* features, applications can be generated with adaptable intra-application flows and behaviors that can be changed dynamically via human interactions or rules engines. The figure below outlines an example of how a multitude of services could be combined and choreographed within an integrated workflow.



The diagram above outlines WebSphere Application Server workflow capabilities. Long running business processes (macroflows) encompass shorter running, synchronous "activities" (microflows). These long running flows can be exposed as services which process a series of activities which can be reversed if one or more activities fail. For example, a service called BookTravel could combine a series of activities called Bookcar, Bookflight, and Bookhotel. If one of these activities fails, the entire service is "rolled back" and undone. These services could also support human interaction as an activity. Multiple services could be combined into a single service. This flexible programming model promotes the future reuse of existing services.

WebSphere Studio provides a visual services choreography tool to create and assemble services into workflows. WebSphere Studio also provides application usage profiling and a business rules engine that can be populated dynamically for controlling a Web services-oriented architecture--as well as EJB transactions.

The following figure displays WebSphere Studio Application Developer with visual workflow composition capabilities.



Application Flexibility

By implementing a services oriented architecture, *WebSphere Application Server and WebSphere Studi0, V5* facilitate the creation and maintenance of flexible applications. The products provide specific tools to help customers and solution providers exploit this capability and make changes to applications on the fly without re-coding.

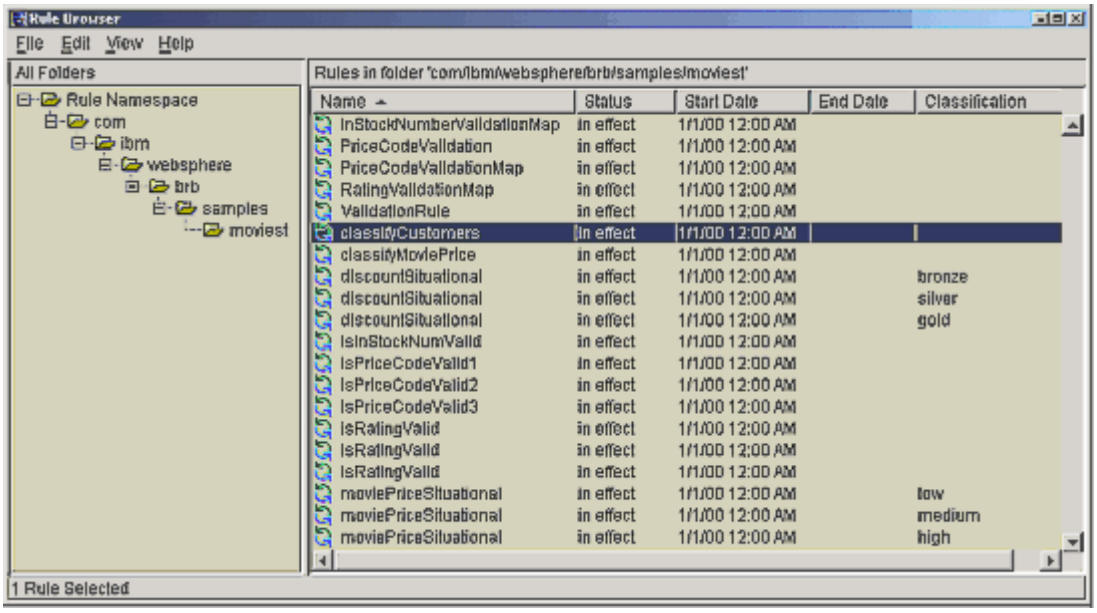
Many businesses today across different industries rely on various business policies and rules as part of their business processes. For example, the insurance industry relies heavily on regulations – both government and self imposed -- which determine insurability. Meanwhile, the telecommunications industry constantly faces regulatory policy changes. Businesses need to be responsive to these changes. *WebSphere Application Server, V5* provides a business rules framework and an internationalization framework to help with these types of scenarios.

Business Rules Framework

The WebSphere business rules framework provides the ability to define, execute, manage, and schedule the rules that encapsulate variable business policy. Any discrete unit of business logic can be

expressed as an externally managed rule. Developers initially create or select a rule that will be triggered from an application. Business analysts or administrators can maintain the rule thereafter without programmer involvement. This framework decreases maintenance and testing costs, increases the consistency of business practices, allows for reuse of policies across business processes, and provides the ability to identify and correct conflicting rules across the business.

The following figure displays the WebSphere Application Server business rules console. Business analysts can define when rules should take effect as well as end dates and rule classification.



Businesses can use this rules framework to enact changes to risk classification policies dynamically and respond to a myriad of state and regional regulatory changes in real time. It can also be used to proactively create and manage service levels. For example, businesses can dynamically classify their best customers and then offer these customers the highest levels of service. Overall, the business rule framework provides the ability to gain control over where and how business rules are defined, and exactly when critical business rule changes should take effect.

Internationalization Framework

WebSphere's internationalization framework provides the ability to extend applications to global constituencies with ease. It provides the framework to display content using appropriate local language conventions and currencies and to account for client time zones as time-sensitive transactions are processed. The WebSphere internationalization framework eliminates months of programming costs and significant maintenance IT costs typically associated with expanding into international markets.

Businesses trying to take advantage of global opportunities now have the ability to deliver intelligent applications that account for customer's language, geography, formatting rules, sorting algorithms, and time zone.

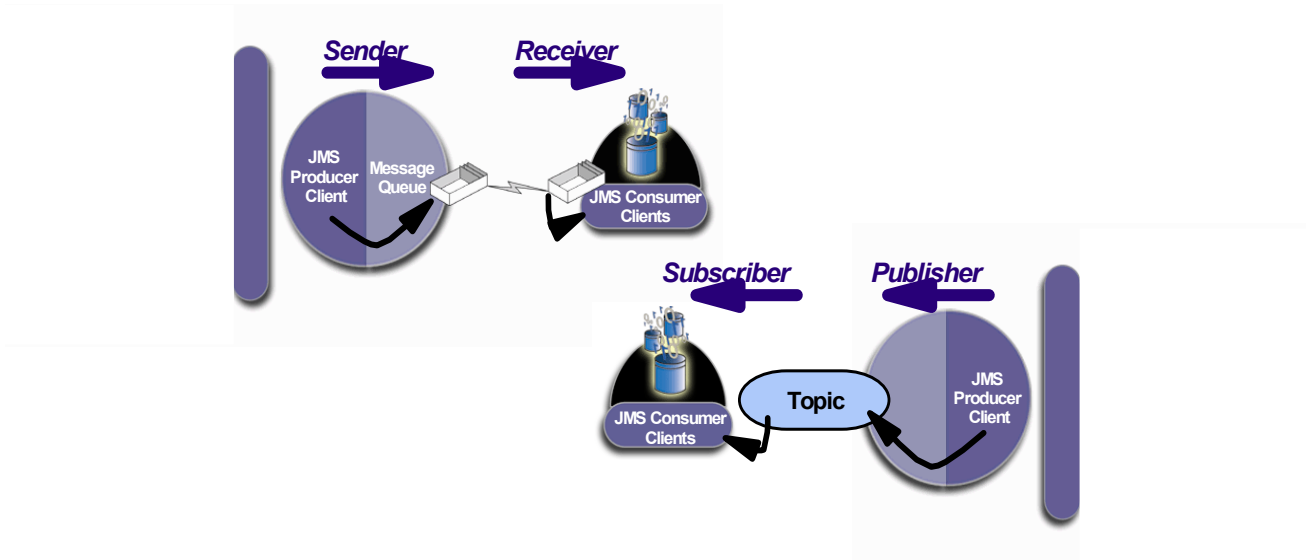
The internationalization and business rules frameworks in WebSphere Application Server, V5 improves the end-user customer experience through increased personalization while improving IT resource utilization through decreasing programming costs and increased application flexibility. Both of these frameworks have been submitted to industry standards bodies for consideration as future standards. Both address outstanding customer problems today.

Integration of asynchronous messaging services

A powerful part of "build-to-integrate" is building new applications that initiate and respond to asynchronous invocations, conversations, and broadcasts. *WebSphere Application Server V5* delivers extended services for mixed synchronous and asynchronous transactional environments as part of the native J2EE 1.3 and Web services environment. A comprehensive, self-contained Java Message Service (JMS) implementation that encompasses queue management and publish/subscribe components is included in the product.

Java™ Messaging Service relies on concepts established in the application messaging market for a number of years. JMS is based on the concept of a JMS "consumer" application (for example an Enterprise JavaBean™ or EJB™) listening for the arrival of a message on a queue, and then executing some business logic based on the message (for example updates to an online catalog), and possibly put a response (such as an acknowledgement) on a response queue. A new specification in J2EE 1.3, EJB 2.0 Message-Driven Beans includes the concept of a "listener" interface with the EJB container which monitors the appropriate queues.

The following figure depicts a publish/subscribe process through a JMS framework.



WebSphere Application Server, V5 also implements a publish/subscribe message broker, where messages are published to a broker against an identifying topic. The messages are then distributed to subscribers whose registered subscriptions match the published messages. All of this is done within an environment that provides fault-tolerant clustering and load balancing, as well as full support for distributed transactions for a more robust implementation of JMS--offering high-performance publish/subscribe technology.

WebSphere Application Server, V5 adds value on top of the core EJB 2.0 specification by delivering container-managed messaging as a way of simplifying the development of these asynchronous applications. Container-managed messaging shifts responsibility for interacting with the messaging services to the EJB container, allowing EJBs to exploit messaging facilities without making explicit JMS calls. Container-managed messaging is analogous to container-managed persistence, where data in entity EJBs can be persisted to the database tables without developers having to implement any database calls.

Finally, *WebSphere Application Server, V5* minimizes costs by maximizing utilization of existing computing resources in supporting line of business applications. WebSphere provides the ability to process workloads through parallel processing. These capabilities allow for scheduling of high priority work processing adding speed and productivity. Background tasks can be automated and scheduled to process during low traffic off hours.

In summary, *WebSphere Application Server, V5* provides a comprehensive "build-to-integrate" platform. WebSphere Application Server and Studio provide integrated visual development tools and dynamic frameworks to leverage existing skills and assets as well as build new agile applications. Based on an open services oriented architecture, integration of existing and new assets is greatly simplified. Applications built with WebSphere Application Server and Studio provide excellent user experience, increased business flexibility and efficient use of IT resources.

Highly Integrated Application Development

The workflow capability is one example that demonstrates the power of the WebSphere integrated development and deployment environments. WebSphere Studio provides a visual tool for composing new applications out of a set of enterprise services – whether they are Web services, EJBs or legacy resources – enabling this new application to be tested immediately, and subsequent changes to be implemented, without disruption to the running production applications.

An integrated application development and deployment platform provides direct positive impact on ROI by:

- Enabling rapid time to market for new applications, from concept through debug and test into production – regardless of the deployment server platform or operating system
- Making it easy to expand and adapt applications as business needs changes
- Facilitating re-use of existing assets and skills when creating new applications
- Driving productivity for both the individual developer -- --through wizards and templates – and across the development team by supporting different developer roles across the organization.

Build Quickly, Expand Easily

WebSphere Studio, V5 integrates development of Web sites (HTML, CSS); dynamic Web applications (JSP, XML, JDBC); and J2EE applications with a WebSphere Application Server unit test environment. The integrated tools and server enable a tight develop, test, debug cycle for both functional and performance testing by each developer.

In addition, by utilizing Eclipse Technolgy -- an open, industry supported software platform for application development -- IBM *WebSphere® Studio, V5* provides an integrated environment for the many different developer roles throughout the application life cycle. The result is higher quality, lower cost, more flexibility and faster time to value.

By leveraging a services oriented architecture, *WebSphere Application Server, V5* simplifies the development and ongoing adaptation of dynamic applications.

Increase Productivity

The time required to roll out new applications is a key concern across all industries, and improved developer productivity is clearly a way to address this. One way to vastly improve productivity is to reduce the need for handcrafted programming. This can be accomplished through powerful frameworks that absorb much of the work involved in development, or through tools that generate code used by the runtime.

WebSphere delivers a combination of frameworks and tools that work together to provide best practice implementations, a realization of the industry's best architectures. With *WebSphere Studio, V5*, developing and maintaining applications is largely facilitated through *visual programming*. As discussed in the "Build-to-integrate" section of this paper, the WebSphere Studio services choreography feature enables a developer to create and combine basic services into workflows or applications. Wiring these interactions together in a visual fashion makes it easier for developers to create applications, and to preserve the flow structure of the application when underlying service implementations change over time. *WebSphere Studio, V5* also provides Web site developers with the ability to visually map a series of actions in order to generate a dynamic Web page.

Still other productivity gains come from the inclusion of wizards, automated project task lists and sample application templates.

Maximize Development Assets

Integral to information and data connectivity is building new applications that integrate multiple back-end systems requiring data transformation and transactional integrity. *WebSphere Application Server, V5* delivers productivity through an open approach to transforming any application asset into a modular network-accessible service, which can be easily identified and reused by other.

WebSphere Studio Assets Analyzer is a powerful member of the WebSphere Studio family that can be used to analyze existing application assets – from Web pages, to Java components, to host assets such as COBOL, PL/1 and JCL. Using the knowledge store built from the analysis of your assets, WebSphere Studio Asset Analyzer helps you identify reusable components; understand the impacts to changing them; and helps you prepare them for broader use as services accessible by new and existing business applications. The result: you maximize the value of exiting assets that have been running your business for years (if not decades) and continue to leverage the skill and experience of all of your developers.

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
11-02