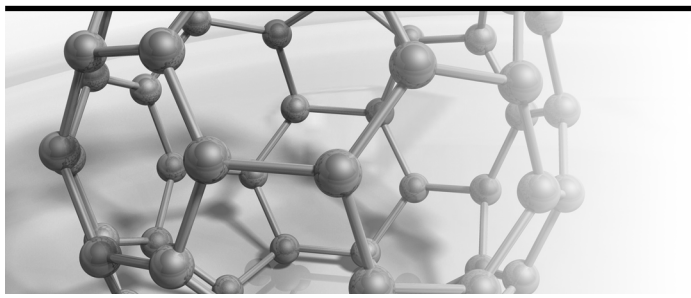

Foreword

When we originally set out to build Version 6.0 of the WebSphere® Business Integration (WBI) portfolio, Service-Oriented Architecture (SOA) and Business Process Management (BPM) were relatively new concepts with only early deployments. While a number of standards-based J2EE-based servers had anchored a pretty well-defined application server market, a wide variety of products and technologies, each with different technological roots, attempted to solve application integration, business integration, process management, and other related middleware problems.

At IBM®, we found ourselves with myriad products in the integration space as well. A strong messaging history with WebSphere MQ and WebSphere Message Broker anchored one key aspect of part of our integration strategy—connectivity. On the other end was WBI-Server Foundation (WBI-SF) as a strategic process engine, sporting an early version of the BPEL specification and built on the WebSphere Application Server Foundation. We also had MQ-Workflow and CrossWorlds'® Interchange Server (ICS) centered traditional workflow and advanced application integration, respectively. Each of these solutions had different development tools and different runtime execution architectures. A package suite called WBI Server 4.x, in fact, had MQ-Workflow (MQWF), Message Broker, and CrossWorlds' ICS all bundled in a single offering, providing choice (and maybe some confusion) to all customers wanting a WBI solution. These offerings were complemented by a set of adapters that came with the CrossWorlds acquisition as well as a WBI Modeler and WBI Monitor product that was an early view of BPM, with relatively strong affinity to MQ-Workflow.

The strategy within IBM's Software Group was focused around the WebSphere Application Server runtime and building out our portfolio of middleware using that strong and robust foundation. The new mission of the WBI team was to offer an integration solution that held the combined capabilities of WBI-SF, CrossWorlds' ICS, and MQWF, built upon the WebSphere Application Server Foundation. The foundation was to be counted upon for things such as configuration, administration, workload management, high availability, and

security. The idea was that this consolidated solution could provide an end-to-end set of integration capabilities. Customers would have fewer runtimes to manage if they needed to solve various middleware problems.

In the early days, we knew of this work as WBI Server Version 6.0. Through a variety of marketing and naming actions, this was renamed the WebSphere Process Server (WPS). So while WPS serves as a broad-spectrum integration server that goes beyond process, the name WPS has persisted.

One of the key goals of WPS and the associated development tools named WebSphere Integration Developer (WID) was to enable the creation and composition of solutions without mandatory coding and the complexity involved in traditional programming-language-based development. This paradigm would then allow a broader set of engineers, with less-intense J2EE skills and, in some cases, very limited Java skills, to contribute and build out solutions. This was and still is a clarifying and influencing premise of the product directions.

To make this objective become reality, it was clear from the early going that just putting some fancy editors on top of a J2EE programming model wouldn't be sufficient. Service Component Architecture (SCA) was invented and produced to provide a service invocation model and a service composition model that would abstract away many of the details associated with specific infrastructure tasks. While SCA, along with Service Data Object (SDO) as a consistent way to deal with data, helped solve the WBI challenge, there was a bigger plan behind SCA and SDO. These technologies, in conjunction with web services and related standards, would come to form the basis of IBM's SOA strategy.

As Business Process Management (BPM) has evolved, we have also evolved our integration products to support a BPM model. Our WebSphere Business Modeler and WebSphere Business Monitor products provide WPS with the complementary function needed for a complete BPM solution. Built on the same fundamentals as the rest of the WebSphere portfolio (Monitor is based on WebSphere Application Server, and Modeler is built on Eclipse), these products have evolved to provide first-class support for processes and services running WPS (and other environments).

WebSphere Process Server and the related WBI products started rolling out in 2005. Since then, a number of interesting things have occurred. We've seen the WBI products used for many new projects, some starting from BPM needs and requirements, while others have started from more technical roots centered around connectivity and application integration. The WBI product set has become the anchor of our SOA and BPM efforts. This has all happened at the same time as customers have begun migrating from the earlier WBI server—namely, MQ-Workflow, CrossWorlds' ICS, and WBI-SF.

At the same time, the Enterprise Service Bus (ESB) movement has come alive. In December of 2005, IBM released the WebSphere Enterprise Service Bus (WESB) product. This product is intended to provide ESB capabilities to a broad range of customers. Architecturally, the same foundational technology that makes up WPS is in WESB. In fact, there's enough software inside of every WPS to allow it to be an ESB.

The book you are about to read does a great job of providing the foundational concepts necessary to succeed with the WBI Version 6-based products, irrespective of whether you intend to capitalize on SOA or BPM opportunities or to meet other integration challenges that you face. The concepts are elaborated upon and supported by deep explanations of the various features and functions of WPS and the related products. Examples and patterns are used to demonstrate how real solutions can be constructed from the capabilities available.

Eric Herness
IBM Distinguished Engineer
WBI Chief Architect