IBM ESB 产品之间的比较及应用场景: 第 1 部分，IBM ESB 产品之间的比较

马 国耀 (maguoyao@cn.ibm.com), 软件工程师, IBM
吴 宇 (william.wu.8156@gmail.com), 软件工程师, IBM

2008 年 11 月 06 日

本文首先介绍了企业级应用程序的发展以及 ESB 的定义；随后，分析了 ESB 在 SOA 解决方案中所起的作用，并比较介绍了三款 ESB 产品在支持实现一个 ESB 解决方案中所起的作用。

企业服务总线 ESB 的介绍

企业应用的发展概述

在介绍企业服务总线之前，有必要花一些笔墨来介绍企业应用架构的发展和变迁。企业级应用架构的发展经历了以下几个阶段：

1. 独立应用系统
2. EAI 阶段
3. SOA 阶段

独立应用阶段

20 世纪 60 到 70 年代，企业应用处于独立应用系统阶段，当时的企业应用是一种用来替代重复性劳动的简单设计，其目的是用计算机代替孤立的，体力性质的工作环节，将相关联的企业信息或数据管理起来。这些系统大部分是独立的系统——有独立的数据库、应用服务器、用户界面。因此有时候这类应用也叫“竖井型”的应用。

但是，随着业务和信息的不断扩展，独立应用系统渐渐不能满足企业对 IT 的需求，表现在大量的信息冗余，因为在建立一个新的应用的时候需要重新建立一套数据库；功能的重新设计，相似的功能存在于多个系统中；例如，客户信息在一个公司中可能有多个拷贝分别存在于多个数据库中，不同时期建立的应用系统所使用的技术也会不同。对于获取客户资料这样的功能，必然存在于多个系统中，而且在不同的系统中其实现方式可能是 Java/J2EE、Delphi、C/C++。

EAI 阶段

20 世纪 80 到 90 年代，一些公司或集成商意识到应用集成的价值和必要性。EAI 是一种将多个不同平台、用不同方案建立的异构的应用集成的一种技术和方法。它的目标包括以下几个方面：各个分离的系统间的相互通讯，消除信息孤岛，实现信息的共享。从功能的角度来看，EAI 包括信息接收、转换、翻译、路由、传播和业务流程管理。从架构上看有两种方式：Hub/Spoke 方式和 Bus 方式。

图 1 所示的 Hub/Spoke 结构使用一个中心代理（Hub）和多个适配器（Spoke）将 Hub 和应用连接起来。适配器负责将应用的数据格式转换成 Hub 可以理解的格式，Hub 将数据再转换成目标系统可以理解的格式，并执行消息的路由。Hub/Spoke 方式的弊端在于只有一个代理中心，当连接的应用种类增加或者消息量增大时，代理中心的性能将成为整个系统的瓶颈，在可扩展性方面也存在着一定的问题。

图 1 . Hub/Spoke 结构的 EAI 集成

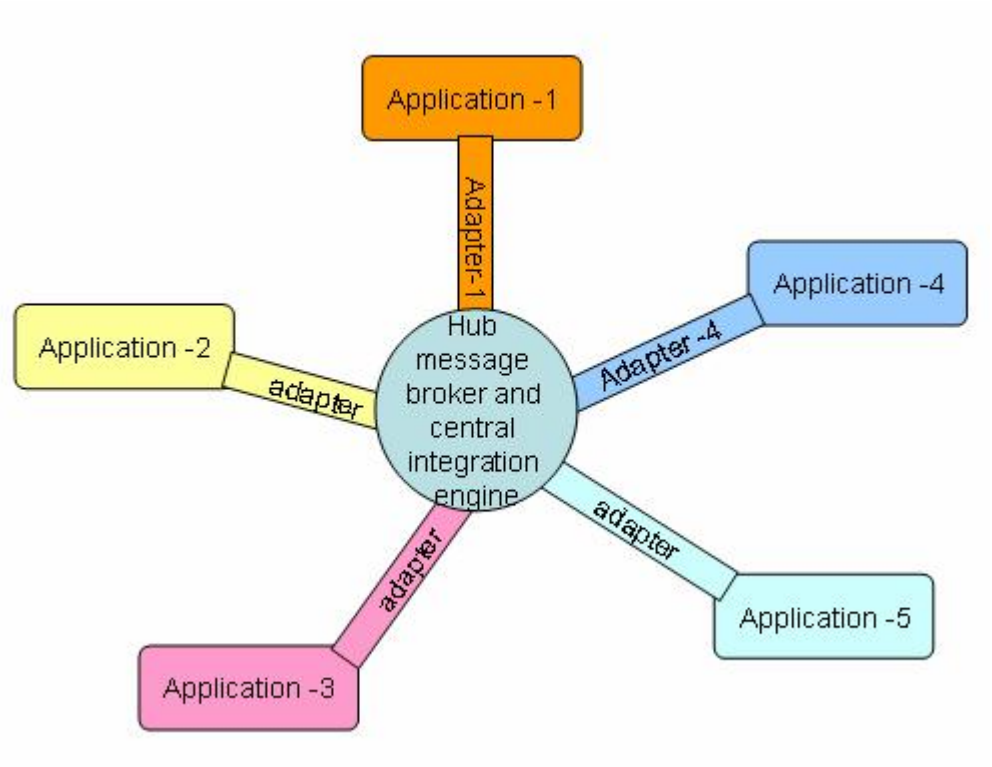
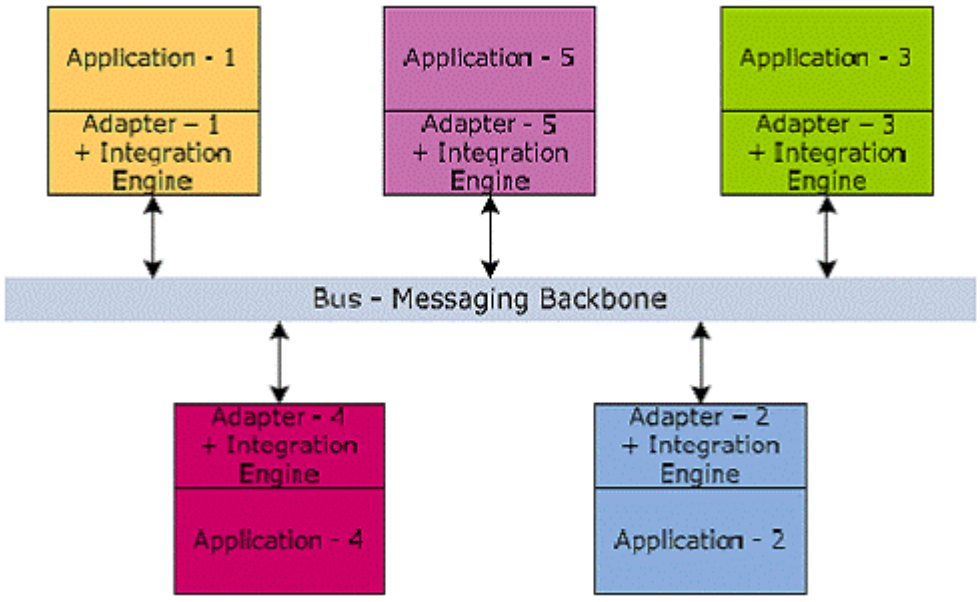


图 2 所示的 Bus 结构使用一个中心总线，应用程序通过 Adapter 将消息发送给总线，总线负责消息的路由，接受方的应用程序也有自己的 Adapter 来转换接受到的消息。Bus 结构和 Hub/Spoke 结构的最大区别在于在 Bus 结构中，Adapter 位于应用程序中，而 Hub/Spoke 结构中，Adapter 由 Hub 来统一管理。这样在 Bus 结构中，加入一个新的应用变得很简单，可扩展性得到了很大的提高，但是应用程序方的负担加重了。

图 2. Bus 结构的 EAI 集成

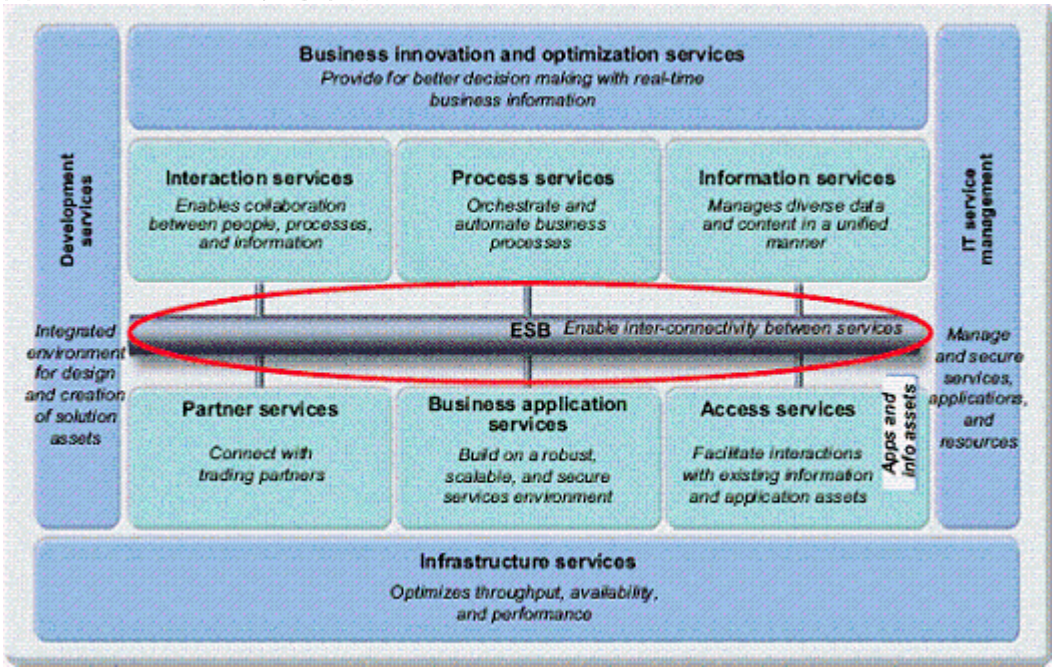


SOA 阶段

SOA 将应用资源看成一个个独立的，自包含并良好定义的服务，通过这些服务的组装，编排可以产生新的应用。每一个服务可以完成一个独立业务功能，并且不依赖于业务上下文或者其他服务的状态。服务的定义是标准的且被广泛支持的，比如 Web Service。在 SOA 的架构中，人们都用标准的方式来封装自己的服务，使得任何一个客户端程序都可以容易的和后台系统实施连接。而 ESB 是 SOA 架构中的一个核心基石，在几乎所有的 SOA 架构中，都将 ESB 放在核心的位

置。图 3 是 IBM SOA Reference architecture，从中我们可以看出 ESB 在一个 SOA 架构中的地位，对该图的详细解释不在本文介绍范围之内，有兴趣的读者可以参考一下 IBM SOA 专区的相关文章。

图 3. IBM 的 SOA 参考架构



下面我们来介绍一下 ESB。

什么是 ESB?

什么是 ESB？ESB 严格来说不是某一个产品，而是一种框架，设计模式。不同的提供商对 ESB 的理解也各有不同。

“ESBs are essentially integration systems, not SOA systems. SOA is about tearing down application silos, but integration systems reinforce those silos. [...] an ESB is especially good for bridging to legacy applications, and therefore it is a useful component in a services infrastructure ”

----Anne Thomas Manes, Research Director with Burton Group

“An Enterprise Service Bus (ESB) is a distributed middleware system for integrating enterprise IT assets using a service-oriented approach. ”

----Ron Ten-Hove , Sun Microsystems and JBI Spec Lead

“A Web-services-capable infrastructure that supports intelligently directed communication and mediated relationships among loosely coupled and decoupled business components. ”

--Gartner

IBM 对 ESB 是这样描述的：

“ An enterprise service bus (ESB) is a pattern of middleware that unifies and connects services, applications and resources within a business. Put another way, it is the framework within which the capabilities of a business' applications are made available for reuse by other applications throughout the organization and beyond. The ESB is not a new software product — it's a new way of looking at how to integrate applications, coordinate resources and manipulate information ”

从 IBM 的立场来说，ESB 不仅仅是一个概念，而是一种中间件模式；它不是某个产品，而是一种全新的集成应用，协调资源和操纵信息的框架。

下面来介绍 ESB 或可以称为 ESB 的中间件产品保护一些特征，有些是必须的，有些是可选的：

1. **连接性**
   ESB 必须提供一种支持服务交互的桥梁，它必须支持多协议 (protocol) 之间的连接。不仅要提供对消息和面向事件的中间件的支持，还要提供和现有 EAI 技术的连接。连接性是 ESB 不可缺少的特征之一。
2. **服务交互**
   服务交互可以理解为 ESB 的一个目的之一，ESB 作为 SOA 架构的核心，必然要支持服务的交互，要在服务的请求者和提供者架起一个坚实的桥梁，让服务的请求者和提供者只需要关心各自的业务逻辑，而不需要在发布和消费服务的环节花很大力气。服务交互也是 ESB 的必备特征。
3. **集成**
   集成的概念是对于系统而言的，ESB 不仅要能集成那些很容易封装服务的系统，也要集成不能方便地封装服务的系统，例如 SAP, ERP, CRM, Siebel 等 EAI 系统、遗留系统。集成也是 ESB 的核心特征之一。
4. **消息处理**
   在集成的过程中，必须要面对的是消息处理，在不同的应用系统中，消息的描述格式是不一样的。在集成环境中，必须要提供一种统一的格式来处理系统间的交互，从 ASBO (Application Specific Business Object) 到 GBO(Generic Business Object) 之间的互转是 ESB 的核心特征之一。
5. **管理**
   对于一个具有 ESB 特征的产品，管理也是一个重要的方面。例如，当一个服务从一个地址切换到另一个地址，在结构等不发生任何改变的时候，ESB 产品应该提供一个方便的途径适应这种改变。
6. **QoS**
   对于服务交互来说，QoS 也是一个重要的特征，比如针对不同的服务请求者提供不同质量的服务响应。有些服务的请求需要在事务中完成，有些服务的交互需要保证其可靠性。一个 ESB 产品应该提供给开发者定义 QoS 的接口。
7. **安全**
   安全的必要性不言而喻，系统和系统之间的交互必然需要认证，授权，加密，签名等安全性。一个优秀的 ESB 产品应该提供可靠的，可灵活配置的安全支持。

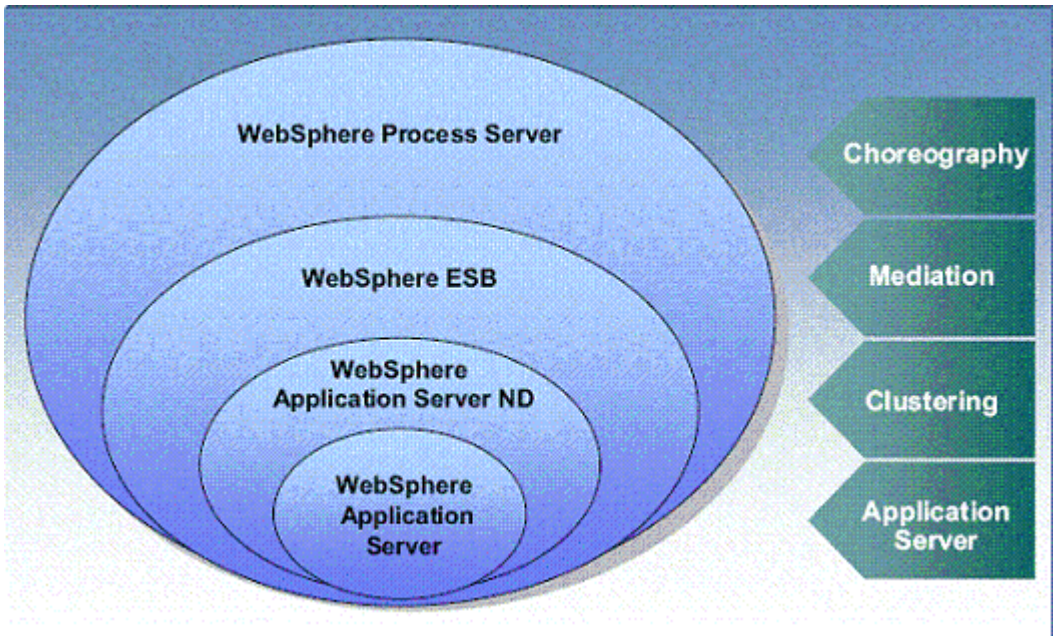
IBM 的 ESB 产品

IBM 有三款 ESB 产品：WebSphere ESB (WESB)，WebSphere Message Broker(WMB)，DataPower。这三款 ESB 产品都提供了 ESB 所必备的特征，但是它们各有侧重，WESB 主要构建与 WebSphere Application Server 之上，侧重于对标准协议和消息的支持，更适合于 J2EE，Web-Service 为主要特征的集成环境；WMB 提供了一个高级的 ESB，它构建于 WebSphere Message Queue 之上，提供了百种以上协议的连接和数据格式的转换机制。Datapower 是一款比较新的 ESB 产品，除了提供必备的 ESB 的特性之外，Datapower 更侧重于安全。众所周知，在 XML 的环境中，安全对于性能的影响是巨大的，Datapower 给企业 ESB 提供了强大的安全保障。下面分别介绍这三款 ESB 产品。

WebSphere ESB

从图 4 中可以看出 ESB 构建与 WAS ND 之上，它使用了 WAS ND 及 WAS 对于安全，用户注册表，事务，消息引擎的支持，在其之上增加了对服务集成、消息流处理、建模以及 ESB 编程模型的支持等等。从图中还可以看出 WebSphere Process Server 是构建与 WESB 之上，并扩展了服务编排和流程管理方面的支持。

图 4. WESB 在 WAS 产品线的位置



下面介绍在 WESB 上实现一个 SOA ESB 解决方案上的以下九个方面的特点，这九个方面的特点来源于上文中介绍的 ESB 的特性，或者特性的细化：

1. **消息转换**

   WESB 所处理的消息为 XML 格式的数据，对于非 XML 结构的数据 WESB 不能处理。对于 XML 结构的数据，在 WESB 的消息流中数据以 SMO(IBM 对 SDO 的扩展，参见参考资料部分了解 SDO 的规范) 形式存在，WESB 可以对 XML 消息树的内容进行修改，包括改变某个节点的内容，增加新的节点以及删除某个节点等等。

2. **支持的协议**

   WESB 支持符合 SOA 标准的协议，比如 SOAP/HTTP、SOAP/JMS、WSDL V1.1、UDDI V3.0，WebSphere MQ 等。也就是说 WESB 目前只支持 SOAP 方式来描述服务，传输协议可以是 HTTP、JMS 记忆原生的 WebSphere MQ 的连接。对于多传输协议的基础，建议使用 MB 来做 ESB 的解决方案，参考 MB 的介绍部分。

3. **消息路由**

   消息的路由在 WESB 中有良好的支持，开发环境 WID 中提供了一个节点专门来负责消息路由，WID 也提供了良好的对路由规则定义的开发支持，开发人员可以很容易的定制负责的路由规则。若要实现动态路由的功能，则需要和一个服务的存储单元中来动态的查找服务，目前 WSRR 是一款优秀的提供该功能的工具（参见参考资料部分了解 WSRR）。WESB 从 V6.1 开始提供了 Endpoint lookup 节点来支持 WSRR 的集成，简化了开发过程。如果要实现简单的动态服务路由的功能，则可把服务的定义存放在数据库中，在 WESB 中通过 DB lookup 来查找服务的 Endpoint, 然后注入到消息流中，WESB V6.1 之前的版本就已经支持与数据库的集成。

4. **对 Web Service 的支持**

   WESB 天生运行在 J2EE 的环境里面，对 Web Service 有着天然的支持。

5. **事件处理**

   在消息流中，我们需要跟踪消息各节点的状态，以满足统计和出错处理的要求，在 WESB 中，通过 CEI 机制来处理消息。（CEI 的介绍请参见参考资料部分）

6. **与遗留系统的集成**

   WESB 通过 Adapter 与遗留系统进行集成，支持 IBM Websphere Adapter 和 JCA Adapter，通过 JCA 我们就可以将遗留系统里面的服务和数据通过标准（SCA/SDO）的形式整合到集成环境中。

7. **安全方面的支持**

   WESB 没有对安全做特殊的处理，使用 WAS 的安全支持来实现 ESB 的安全。

8. **性能**

   WESB 是一个纯 Java 的应用，运行效率上有些限制，同时可以处理的消息流的数量级为几

十到几百之间。

9. **开发和部署**

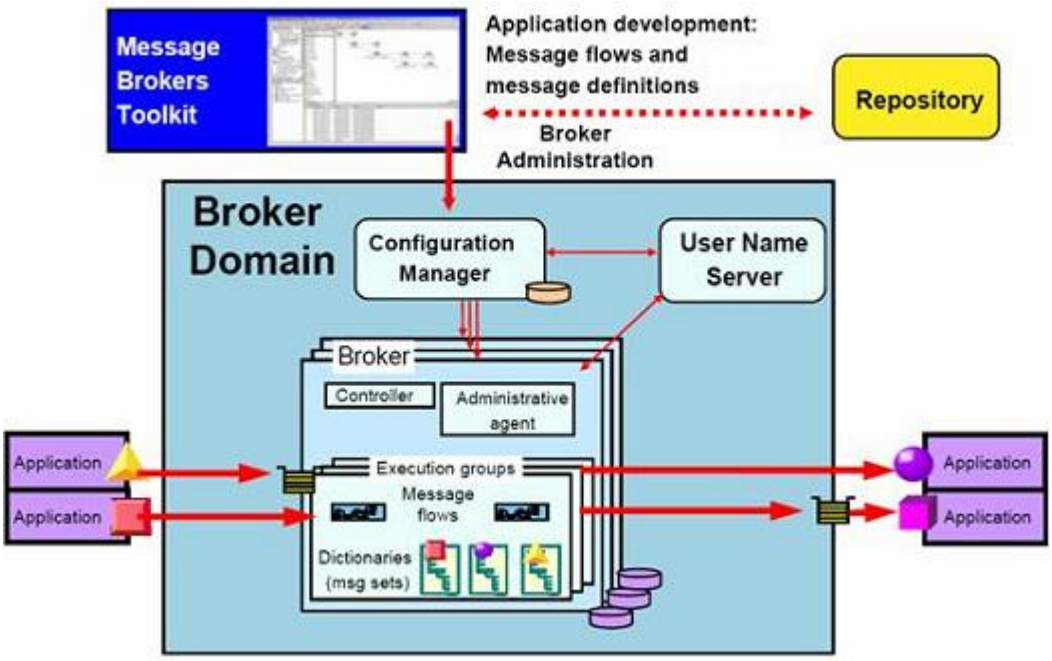
   开发工具是 WID，一个 ESB 的消息流在 WID 中被称为 Mediation Module，它是一个 J2EE 应用，开发和部署工作无异于普通的企业级应用。

Message Broker

WMB 是 IBM 的应用整合中间件，是 IBM ESB 架构重要的产品组成部分之一，用于企业应用整合领域。WMB 目前的版本是 6.1.2，它的前身是 MQSeries Integrator。本质上看，WMB 是 MQ 的衍生产品，它使用 MQ 作为内部通信的机制。然后，WMB 提供的接入方式远不止 MQ 一种，包括 JMS、HTTP(S)、SCADA 等常见的新一代接口规范。在消息转化过程中，WMB 能够识别 XML、C 结构、SOAP 等各种自定义的消息格式。

如图 5 所示，WMB 可以分为开发环境和运行环境两大部分。其中开发环境由开发工具（Toolkit）和调试环境（Rational Agent Controller）组成，运行环境是 WMB 核心，也叫 Broker Domain，由三部分组成，配置管理器（Configuration Manager）、用户名服务器（User Name Server）和代理（Broker）。

图 5. WMB 组件



- 代理（Broker）：消息代理是 WMB 的消息处理引擎，它提供 WMB 的所有运行服务，在 Windows 系统上它是一个系统服务，在 Unix 平台上表现为一个后台进程。应用系统利用与 MQ 的连接和队列将消息发送到消息代理。代理与代理之间，代理与配置管理器之间通过普通的 MQ 发生和接受类型的消息通道进行通信。在一个主机上我们可以创建一个或者多个 Broker，每个 Broker 会关联一个数据库，利用数据库存储 Broker 需要和相关的信息，每个 Broker 还需要一个队列管理器，多个 Broker 之间不能共享同一个队列管理器，每个 Broker 必须有自己特定的、唯一的队列管理器。每个 Broker 只能被一个配置管理器控制。
- 配置管理器（Configuration Manager）：配置管理器是整个 WMB 运行环境中的控制中心，它维护整个 Broker Domain 的配置信息，配置和管理所有代理，增、删、启动、停止消息流，所有的开发工具也是通过配置管理器来部署编译结果的。配置管理器也负责与用户名服务器联系，配置和管理各种用户权限。配置管理器和其他各个部件之间的接口是 MQ，所以配置管理器也必须是基于 MQ 队列管理器而运行的。
- 用户名服务器（User Name Server）：用户名服务器是 WMB 运行环境中的可选部分，它可以提供应用程序的接入认证以及订阅主题的访问控制服务。用户名服务器本身是不需要数据库，但必须依赖于队列管理器与其他组件通信。用户名服务器中的用户信息来自认证数据文件。
- 开发工具（Toolkit）：开发工具是基于 Eclipse 3.0 的集成开发环境，我们可以在其中开发消息流、消息集、ESQL 代码、Java 代码、映射规则等。通过内置的 MQ Client 与配置管理器连接，并将开发好的执行组件部署到相关的代理中。Eclipse 环境由不同的视图

（Perspective）组成，我们在 WMB 中常用的有代理开发视图、代理管理视图等。

- 调试工具（Rational Agent Controller）：这是 IBM Rational 开发工具标准的调试工具。一般安装在代理所在的服务器一边，开发工具可以连接 RAC，通过它来控制和调试消息流的运行。需要注意的是，Toolkit 可以并行开发，但是对于同一个执行组不能并行调试，因 RAC 的调试过程是排他的。在 WMB6.0 之后 RAC 不需要再单独安装。

下面从九个方面来介绍 WMB 在实现一个 SOA ESB 解决方案上的支持的特点：

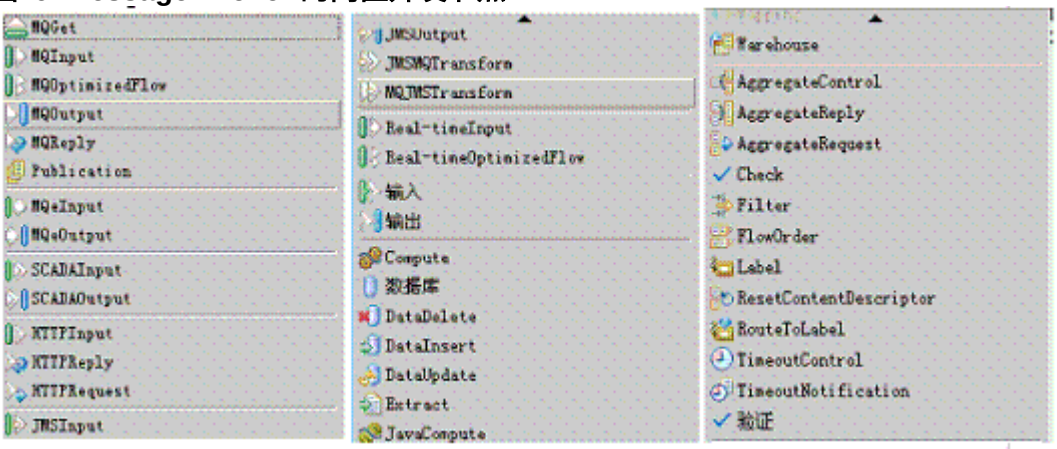
1. **消息转换**

   WMB 在消息处理方面功能非常强大，对 XML 格式的消息或非 XML 格式的消息，如 C Header，COBOL 等都有很好的支持。实际上，通过开发消息对应的消息集（Message Set），可以在消息流中对任意格式的消息进行修改。WMB 提供了内置的 mapping 和 database 节点，用户可以通过图形化的方式方便的实现消息的转换或与数据库的交互。

2. **支持的协议**

   WMB 支持所有 WESB 支持的传输协议，除了常用的 HTTP、JMS 等，还对 FTP、Socket、Mobile、Telemetry、Biztalk 和 Tuxedo 等有良好的支持。WMB 与 MQ 有着紧密的联系，对 MQ 的支持不在话下。WMB 内置的功能节点对这些协议提供了很好的支持，仅需配置即可，如图 6 所示。

   图 6. Message Broker 的内置开发节点

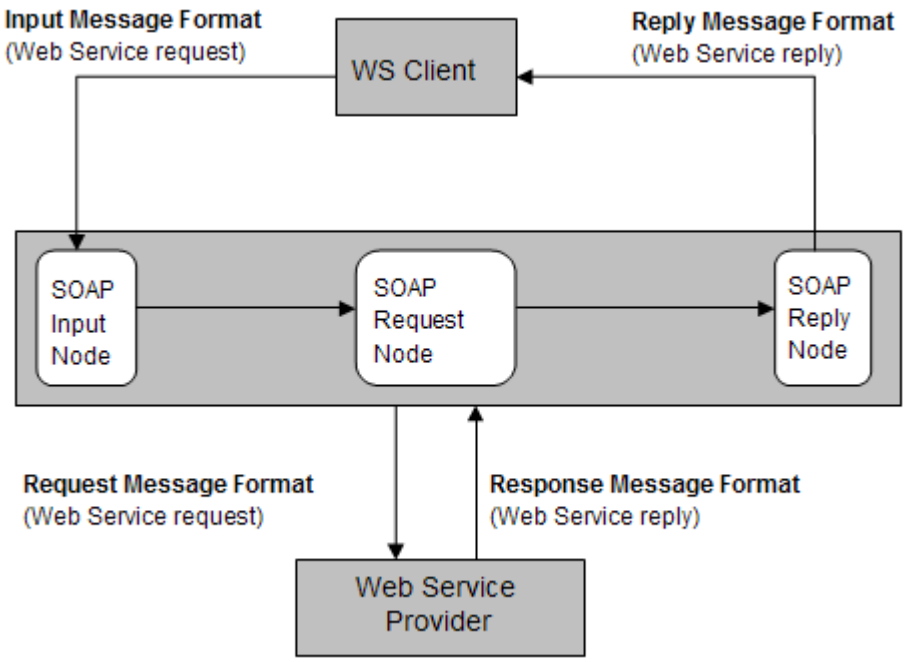


3. **消息路由**

   WMB 提供了很多功能强大的内置节点支持消息的路由，如 Filter 节点、Label 节点等，在新版本的 WMB 中又引入了 Router 节点，该节点几乎和 WESB 中的 Router 节点一样。若需要实现动态路由，可以使用 WSRR 作为服务的存储，WMB 和 WSRR 有很好的集成，通过 RegistryLookup 和 EndpointLookup 我们可以在消息流中实现动态路由。另外需要指出的是，WMB 可以提供一个消息输入，多个消息输出的功能，可以实现一个消息同时路由到多个输出端。

4. **对 Web Service 的支持**

   在 WMB 中，消息流可以作为 Web Service 暴露出去供外部调用，也可以作为客户端调用外部提供的 Web Service。WMB 不仅提供了内置的 SOAPRequest、SOAPInput 等节点实现对 Web Service 的支持，而且对 WS 扩展，如 WS-Security 和 WS-Addressing 也有良好的支持。



5. **事件处理**

   在 WMB 中我们可以通过 Trace Service 来记录所发生的事件。Trace 分两种，一种是 User Trace 记录消息流级别的事件，另一种是 Service Trace，可以记录整个 WMB 的事件，如 Broker 的部署执行。WMB 提供了 Trace 内置节点，可以实时的记录流程中消息内容的变化。

6. **与遗留系统的集成**

   WMB 对遗留系统有良好的支持。对 SAP，PeopleSoft 等大型的 EIS 系统，直接提供了内置的节点，通过 JCA Adapter 配置的方式和 EIS 系统交互。对于比较特别的遗留系统，如 CICS、VSAM 等，WMB 提供了丰富的 SupportPacs，客户可以下载并安装。

7. **安全方面的支持**

   WMB 本身提供了两个层次上的安装，一个是部署时安全性，管理部署 bar 文件到 Broker 以及运行 WMB 管理命令的权限控制；另一个是运行时安全，涉及的权限控制包括发送消息到相应的消息流，以及消息流可以访问哪些 MQ 资源和非 MQ 资源，如数据库系统。

8. **性能**

   WMB 底层是使用 C++ 开发的，在性能上相对于 WESB 有很大的提高。同时可以处理的消息数量级为几千到几万之间。

9. **开发和部署**

   开发工具是 WMB Toolkit，我们开发的消息流和消息集被打成 bar 文件通过配置管理器部署到 Broker 中。

总的来说，WMB 是 IBM 久经考验的一款消息中间件产品，为异构的 IT 环境提供了统一的连接和转化，其优势如下：

- 利用 WebSphere MQ 作为企业消息传递主干，提供了很好的并发性和可靠性
- 不仅支持各种标准协议，而且支持和 WebSphere 企业应用程序适配器进行集成
- 支持多种数据格式之间的转换，包括 XML、遗留系统、行业标准和自定义消息格式
- 针对大量数据处理进行了优化，极大提高了处理速度。

Datapower

DataPower 是一个硬件产品，看起来像一个盒子。目前 IBM 有三款 Datapower 产品供用户选择。按照进入市场的先后次序，它们分别是：Datapower Accelerator XA35，Datapower Security Gateway XS40， Datapower Integration Appliance XI50，见图 7。

图 7. IBM 的 Datapower 产品线

- XA35 的主要侧重点是 XML 加速，众所周知，在 SOA 的环境中，XML 是应用最广泛的，而对 XML 的解析，加密，解密这些操作让应用服务器来做的话，这是一个很大性能开销，而 XA35 就是要替应用服务器的分担这些 XML 相关的工作，让应用服务器专注于处理业务逻辑；
- XS40 的主要侧重点是安全，即负责提供安全的企业网关，及对 XML 攻击的防范。
- XI50，从名字上看，主要侧重点是集成功能，是一款高级 ESB 产品，提供了多种协议和数据格式的路由和解析的功能。

虽然三款产品各有侧重，但是三款产品的功能也具有包含关系，XS40 包含 XA35 的功能，XI50 包含 XS40 的功能。

我们依然从以下九个方面来介绍 Datapower 在实现 ESB 解决方案中的特点。

1. **消息转换**

   Datapower 对 XML 消息有强大的支持，但是 Datapower 绝不仅仅支持 XML，我们可以在 Policy 中使用 Transformation 节点来对消息进行任意我们需要的转换，其原理是使用 XSLT 来实现的，开发人员定义自己 XSLT Stylesheet 并在 Transformation 节点中指定，Datapower 负责转换。

2. **支持的协议**

   Datapower 支持和以下传输协议，HTTP，HTTS，WebSphere MQ WebSphere JMS，TIBICO EMS，FTP Poller, FTP Server, NFS 等等。 Datapower 的 MPGW 就是一个处理不同协议的应用系统的互联的服务对象。

3. **消息路由**

   Datapower 支持对服务和消息的路由，根据消息流中的上下文连接将消息动态的分发到不同的消息提供者。但是 Datapower 的动态路由和 WESB 以及 MB 的动态路由还是有区别的，Datapower 的动态路由需要由开发者定义路由的 Map，而 WESB 和 MB 支持在消息头的属性里动态的设置 Endpoint 的地址。目前 Datapower 可以和 WSRR 集成来定义 WS-Proxy（Datapower 中的一种服务对象），但不支持直接和 WSRR 联合实现动态访问 Endpoint 的功能。

4. **对 Web Service 的支持**

   Datapower 的 XMlFirewall 和 WS-Proxy 提供了强大的对 Web Service 的支持，而且 datapower 提供了细粒度的对 Web Service 的控制，可以从服务级 (Service)，端口级 (port)，绑定级 (binding)，操作级 (operation) 来对消息体进行控制。此外，对 WS-Security 也提供了强大的支持。

5. **事件处理**

   Datapower 中可以通过 Probe 的方式来跟踪消息流的中间状态，在 Probe 中，可以看到消息流的每个节点的消息内容。Probe 一般用于开发调试过程，在生产模式下一般不使用，因为使能 Probe 会牺牲一定的效率。Datapower 不支持与 CEI 类似将消息发送到其他应用系统的机制。

6. **与遗留系统的集成**

   Datapower 不支持和 Adapter 的连接，若要与遗留系统的集成，则需要通过其他中间件转换在遗留系统和 Datapower 之间做而桥梁来连接。

7. **安全方面的支持**

   Datapower 的强大之处在于其对安全方面的强有力的支持，它提供对 XML-attack 的原生支持（关于 XML-attack 的知识参见参考资料）；此外，Datapower 可以对 Web Service 提供细粒度的安全支持，包括加密 (Encryption)，解密 (Decryption)，签名 (Sign) 和确认 (Verify)，以及 HTTS 方面的支持。这些支持在 Datapower 上开发起来都异常简单。

8. **性能**

   Datapower 无疑是三款 ESB 产品中性能最高的，对 XML 的处理速度达到线速，下图是一

组测试结果。如果去处网络传输在其中的比例，对 XML 的处理性能所提高的倍数可达到上百倍。

Software Solution		DP Solution	
Avg. response time (TimeP / N)	XML processing time (TimeX)	Avg. response time (TimeP / N)	XML processing time (TimeX)
106 ms	96 ms	19 ms	11ms

9. **开发和部署**

   Datapower 的没有相应的开发工具，但是提供了 Web GUI 的管理控制台和 CLI 方式的管理支持。我们在 Web GUI 下开发消息流，开发即部署。

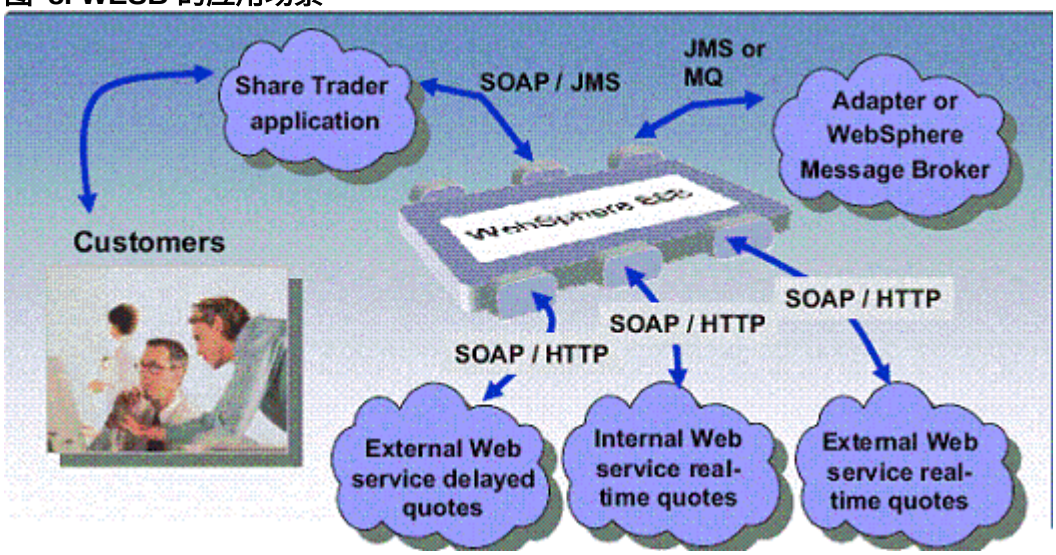
   高速的 XML 处理能力和强大的安全支持，是 Datapower 作为 SOA 应用中的 ESB 的重要特色，这个特点使得 Datapower 成为一款举足轻重的 ESB 产品。

三款产品的比较

从上面的介绍我们可以看出，三款产品都提供了 ESB 的必须的功能，但各有侧重：

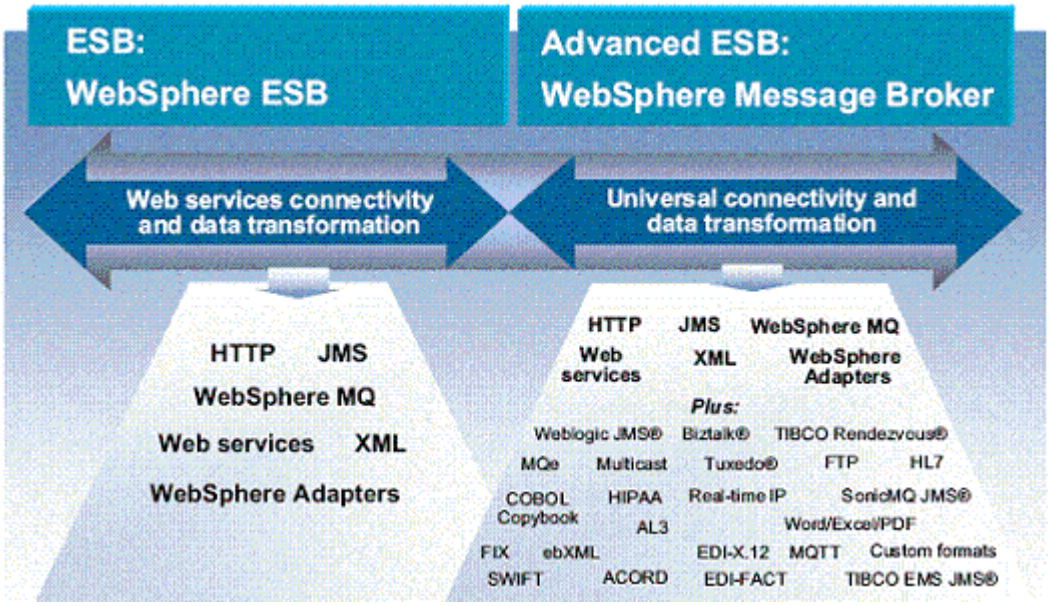
- WESB 是一个轻量级的 ESB，侧重于标准协议，SOAP,JMS 等应用的基础，构建于 WASND 基础之上，提供了和 J2EE 应用很好的集成功能；同时由于 WESB 是一个纯 Java 的应用服务器，在性能上也是相对较差一点的 ESB，可并发执行的 Mediation Flow 的数量级在几十个左右。WESB 适合使用于对性能要求不是很高，且遵循标准协议的 SOA 整合环境中。WESB 的优势是提供了和流程服务器 WPS 以及 J2EE 服务器 WAS 良好的整合。WESB 的应用场景见图 8：

图 8. WESB 的应用场景



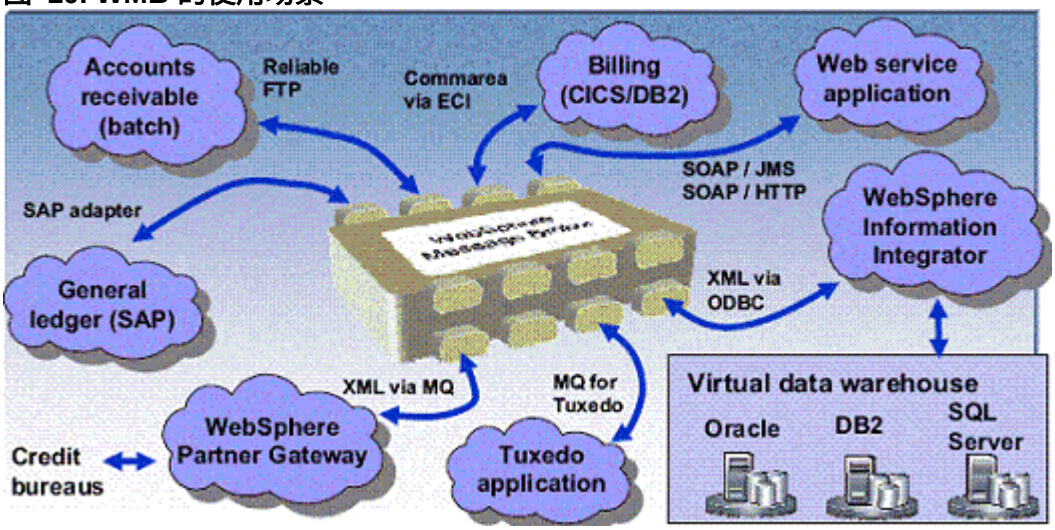
- WMB 是一款高级的 ESB，提供了比 WESB 多很多的传输协议，数据格式的支持，它所支持几乎大部分常用的数据格式和协议。并且 WMB 提供了良好的扩展功能，开发人员可以在 WMB 基础上开发自己的数据格式解析的节点。WMB 使用 C/C++ 编写，在处理性能上比 ESB 也要高出很多倍，可并发执行的流可以达到上百个或上千个。从图 9 可以看出 WESB 和 WMB 在支持的协议上的区别。

图 9. WESB 和 WMB 的比较



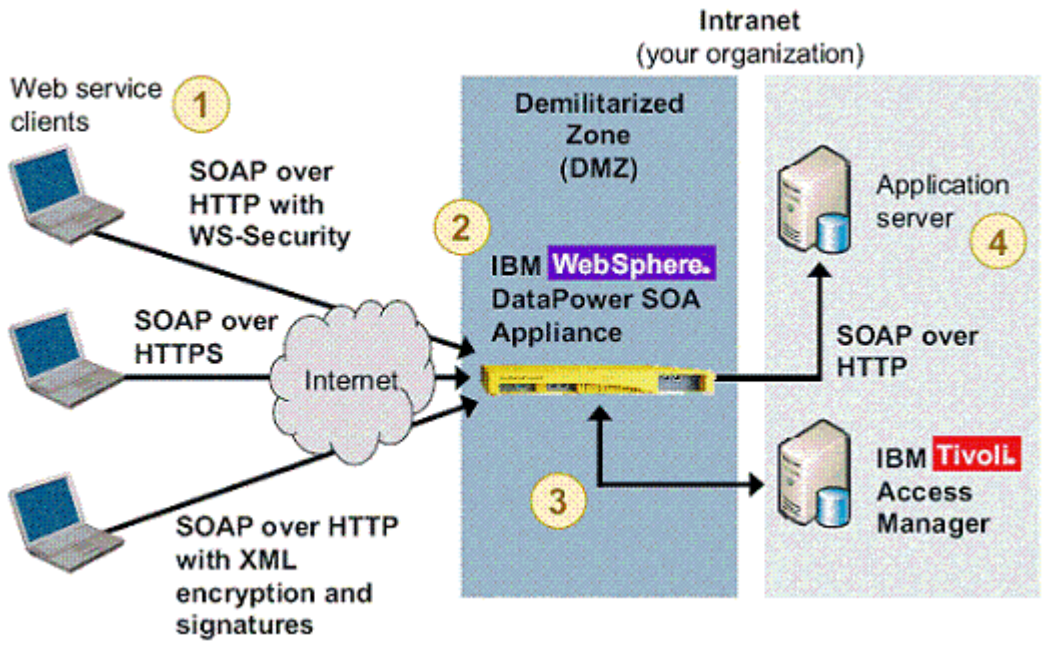
WMB 应用于对性能要求相对较高，多种复杂协议存在的集成环境中。另外，WMB 构建于 WebSphere Message Queue（WMQ）之上。 WMQ 提供了对异步消息提供了可靠的传送机制，比较适合于信息传输量较大，信息交互频繁的场景中。如图 10 所示：

图 10. WMB 的使用场景



- Datapower 是 SOA 中的又一重要的 ESB，在 WESB 和 WMB 中都是用软件来实现 XML 解析和安全支持的，而 Datapower 使用硬件的 XML 解析和加速器，在性能上有了很大的提高。在很多 SOA 的环境里，安全和性能是天平的两端，增加安全势必要牺牲性能；而提高性能则需要牺牲安全。所以在安全和性能要求都比较高的环境中，Datapower 是一首选 ESB，因为 Datapower 可以在实现高性能的同时也保证安全。图 11 是一种 Datapower 的应用场景：

图 11. Datapower 的使用场景



总结

基于上文对三款产品做了详细的比较，我们可以做如下总结，WESB 适用于 J2EE 环境下，对性能要求不是很高的，标遵循标准协议的 SOA 集成；WMB 应用更复杂的集成环境，表现为数据格式多种，传输协议多样，性能要求很高；而在安全和性能要求都很高的应用场景下，选择 Datapower 无疑是最好的选择。下面的图表再次对文中的描述进行总结。

ESB 功能特点	WESB 的支持	MB 的支持	Datapower
消息转换	XML	XML、非 XML	XML、非 XML
支持的协议	HTTP,JMS, WMQ 等	多达上百种	介于前二者之间
消息路由	强大，灵活	功能强大，灵活	灵活度比前二者稍弱
Web Service	强大的支持	支持 WS 扩展	强大的支持
事件处理	CEI，可以和外部事件消费系统监控	Trace Service	用于调试 Probe
遗留系统的集成	Adapter	丰富的 SupportPac	特定的遗留系统
安全	依赖 WAS 的安全	部署和运行时两个级别的安全	超强的安全支持
性能	几十到几百每秒	几千到几万每秒	达到线速
开发和部署	WID 集成开发环境	WMB Toolkit	WebGUI

在本文的第 2 部分中，我们将用一个简化的实际案例来描述三款 ESB 产品在实现上的差异。

参考资料

学习

- CEI 的介绍：从这里您可以了解 Common Event Infrastructure 相关的内容。

- EAI vs. SOA vs. ESB： 本文向您讲解了这三个概念之间的关系和相关的内容。
- WebSphere Enterprise Service Bus 的产品文档中心： 在这里您可以找到所有和 WESB 产品相关的资源。
- WebSphere Message Broker 的产品文档： 在这里您可以找到所有和 WebSphere Message Broker 产品相关的资源。
- WebSphere Enterprise Service Bus 资源中心： 这里拥有 developerWork 中所有和 WESB 相关的内容，让您可以更全面地了解和学习这个产品。
- IBM developerWorks SOA and Web services 专区 提供了大量的文章，以及关于如何开发 Web 服务应用程序的初级、中级和高级教程。
- 使用 IBM SOA Sandbox 进行试验！通过 IBM SOA 进行实际的亲手实践来提高您的 SOA 技能。
- IBM SOA 网站 提供 SOA 的概述，并介绍 IBM 是如何帮助您实现 SOA 的。
- 了解关于 developerWorks 技术事件和网络广播 的最新消息。请特别关注以下 SOA 和 Web 服务技术讲座：
  - SCA/SDO:To drive the next generation of SOA
- 访问 Safari 书店 ，浏览有关这些技术主题以及其他方面的书籍。
- 查看快速的 Web 服务按需演示 。
- 获取 SOA and Webservices 专区的 RSS。（了解关于 RSS 的更多信息。）

获得产品和技术

- 使用 IBM 试用软件 开发您的下一个项目，可下载或索取 DVD 光盘。

讨论

- 参与 developerWorks Blog，从而加入到 developerWorks 社区中来，其中包括以下与 SOA 和 Web 服务相关的 Blogs：
  - Sandy Carter 的 Service Oriented Architecture -- Off the Record
  - Ali Arsanjani 的 Best Practices in Service-Oriented Architecture
  - Bobby Woolf 的 WebSphere SOA and J2EE in Practice
  - Eoin Lane 博士的 Building SOA applications with patterns
  - Kerrie Holley 的 Client Insights, Concerns and Perspectives on SOA
  - Simon Johnston 的 Service-Oriented Architecture and Business-Level Tooling
  - Sanjay Bose 的 SOA, ESB and Beyond

作者简介

马国耀，2007 年毕业于北京大学信息科学技术学院，现在 IBM CDL 的战略合作部担任软件开发工程师。对 SOA，企业集成，消息中间件等方面都有很广泛的爱好。曾发表过使用 Datapower 实现ESB文章。业余时间喜欢围棋，五子棋，扑克等网络游戏，旅游和网上聊天业是他的爱好。你可以通过 maguoyao@cn.ibm.com 和他联系。

吴宇：毕业于上海交通大学，现在在 IBM BPTSE 部门担任软件开发工程师，对 SOA 和企业集成应用方面有浓厚兴趣，你可以通过 william.wu.8156@gmail.com 和他联系。