

.NET与J2EE只能是敌人吗 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022_NET_E4_B8_8EJ2E_c104_253541.htm 目前多数企业内系统多是多层式的架构，可分为展示层、中介层与资料层。因此，整合便会在这几层之间产生多种连接点的组合。其中，中间层技术整合最为复杂，包括展示层到中介层（P to D）、中介层到中介层（D to D）等。过去几年间，许多厂商所建构的组件技术与标准即是用来协助于企业内部建立各种分布式系统，包括有：

- Distributed Component Object Model（DCOM）：微软让那些使用COM规格所撰写的组件可以进行分布式应用，并让组件在远程机器被呼叫。
- Common Object Request Broker Architecture（CORBA）：这是OMG（Object Management Group）所提出可跨越不同厂商进而统一分布式系统技术的规格。
- Java Remote Method Invocation（RMI）：Java v1.1.x的核心规格，允许用Java所撰写的组件可以被分散至其它机器或是行程中。虽然如此，这些技术基本上还是受限于企业内部，甚至是某些固定的平台之上。虽然微软提出COM Internet Services（CIS）技术，可让DCOM透过port 80沟通；另一方面，SUN也将RMI over Internet Inter-Orb Protocol（IIOP）纳入Java规格，但对于那些需要跨越企业内外网络，甚至是进行不同平台间的整合工程而言仍然不足。幸运的是，随着企业的需求日增与技术演进，现在我们已拥有多种选择可轻易地整合.NET与J2EE两大平台。在目前的技术中，两者的整合机制可分成三种类型：底层协定（Wire Level）这是走低阶协议以进行整合的第一种方式。当然，除了「苦工式」整合，也

就是自己建立socket或经HTTP通讯协议进行之外，技术人员也可考虑选用协力厂商的产品，例如：Intrinsyc Software的Ja.NET，或是JNBridge旗下的整合软件等。（前者当然是Java与.NET名称的整合，后者为Java与.NET桥梁的意思）。其中，「Ja.NET」可视为Java之上的.NET Remoting（编者按：Microsoft .NET Framework内的主要组件）的堆栈实作，而在Java平台上提供Ja.NET的执行时期模块（Run time），可支持TCP/IP、HTTP等沟通管道，也可同时支持SOAP或是二进制互通协议以提升沟通效率。透过此执行时期模块，.NET与Java/J2EE的数据类型不仅可以对应，还能进行双向的沟通。JNBridge也是类似架构，透过对应的执行时期模块与代理程序（proxy），.NET程序可以在不需要Java原始程序的状况下与这些组件进行互通、继承，并将其视为同一个程序内的.NET组件。以下为JNBridgePro的架构图：这类整合方式有诸多优点，包括更佳的互通效率、对象参考与生命周期的控制、支持回呼程序（call back）与事件（event），而能有更紧密的整合效益。但相反的，因为是较紧密型的整合，弹性也会变低。另外，这类整合也通常缺少动态寻找并新增服务的机制。一般来说，对于企业内部不同平台的整合仍是非常不错的选择。讯息队列或集线器（Message Queues或Hub）点对点的整合只适合初期项目，也许利用上述的底层协议方式，或是下文将会提及的Web services进行互通。但是当.NET有N个模块，J2EE有M个模块，要互通就需要建立「N*M」的点对点联机，复杂性与困难度将之提升。因此，当整合进行到一定规模，可以开始考虑采用类似讯息队列或是集线器等方式进行。目前可见软件，如MSMQ、IBM WebSphere MQ

、Microsoft HIS、BizTalk Server，或者是Mind Electric公司的GAIA等，都能有效的将整合数量如同集线器一样减至N M的状态。这类技术概念如同集线器，可以整合不同的接口或透过外挂的Adaptor增加对于不同接口的支持。以Microsoft BizTalk为例，微软与协力厂商所开发的Adaptor便超过一百个，其中包括SAP、Siebel、Java/J2EE、Web services、SQL Server、IBM WebSphere MQ等相对应的Adaptor. 换句话说，只要把先前.NET的N个模块与J2EE的M个模块各自透过Adaptor「安插」至类似BizTalk Server等具备「集线器概念」的服务器，即能整合与应用不同组件。由于不同平台之间的组件是非常松散结合的（loosely couple），相依性较低而适合N对M的整合以达到「服务导向架构」（SOA）的目标，这也是此类整合的诸多优点之一。例如，将一个.NET组件经Adaptor串接至某集线器概念服务器之后，将可用不同的方式存取此组件，也许是经由J2EE、或者是利用Web services，甚至是IBM的MQ Series.如此一来，对.NET组件开发者而言，完全不必担心未来使用这个组件的对象与技术平台为何。为满足进阶的需求，这类型服务器部分也内建安全性、交易、路由器等功能，导入成本当然很高，甚至个别的Adaptor也要分开购买，因而适合有大量整合需求的企业采用。网络服务（Web services）前述两种方式之外，以SOAP为基础的Web services进行异质平台整合，可说是最具弹性与成本优势的选择。虽然Web services的规格在WS-I等国际组织推动之下，仍是「现在进行式」，但对于.NET与J2EE两大平台进行基本整合与互通而言已是游刃有余。目前.NET与J2EE两大平台都有对应的Web services实作，包括：.NET：除了提供旧版本Web services支持能力

的Web services Toolkit与Microsoft Visual Studio .NET开发工具之外，几乎所有微软的产品都加入了Web services的支持，包括Microsoft Office System、Windows Server System...等，其它还有如Borland的Delphi 8 for .NET、C# Builder...等。J2EE：包括有Apache的Axis、IBM的WSTK和WSAD，以及Mind Electric的Glue...等。其中，Mind Electric将Glue称为Java Web services的「Turbo Pascal」，意思为用Java撰写Web services最简单、最容易入门的工具。除简单易用之外，Glue可单独运作或是外挂至不同的应用程序服务器，包括WebLogic、WebSphere、JBoss等，而其执行效率也比很多其它品牌的应用服务器所实作的Web services效率更佳。若从技术细节剖析，透过Glue可以将EJB对外包装成Web services，并可以和JASS进行安全性整合、透过JMS提供可依赖的讯息机制...等。因此如果只是想单纯的加入Web services支持，使用Glue会比升级应用服务器更划算。进行整合的阶段虽然上面介绍了众多不同整合的技术，但是一旦企业产生异质平台整合的需求，透过Web services先建立一个连接点对点的实验性项目是比较好的选择，一方面因为不同平台对应的技术已经非常成熟而开发容易，另一方面也是最节省成本而能清楚检视效益的方式。当然，如果不满足于互通的效率，或是希望更进一步的进行更紧密的整合，包括继承、双向沟通、数据型别的对应等，使用协力厂商所提供的低阶整合技术也是可以考虑的选择。一旦整合端点进行到超过一定数量以上，便该开始考虑讯息队列或是集线器概念服务器等的软件，将可以大幅增进异质平台整合的速度。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com