

Java、.NET，为什么不合二为一？PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/245/2021\\_2022\\_Java\\_E3\\_80\\_81NE\\_c104\\_245771.htm](https://www.100test.com/kao_ti2020/245/2021_2022_Java_E3_80_81NE_c104_245771.htm) 十二年之前，Sun公司默默宣布了一种可以使网页更动感和更富有活力的编程语言及其环境。当然，目前Java语言已经成为了一种普遍使用的工具，不仅仅用于为网页添加更多的动态效果，还包括创建和生成这些网页（透过servlets和JSP技术），提供一个用于事务性过程和商业逻辑的平台（透过EJB技术，即Enterprise Java Beans），访问消息系统（透过JMS技术，即Java Message Service），访问关系型数据库（透过JDBC技术），甚至于访问不同的主机（透过Java Connector API技术）。但这个故事还远远没有终结，每天，以Java为中心的社区通过开源的努力和大量的项目变得越来越强大，甚至于官方的Java平台也不断地通过Java Community Process这样一个开放性的国际组织来进行构建、成长和对自身进行增强。六年之前，微软大张旗鼓地宣布了一系列崭新的编程语言和应用用于各种开发场景下的环境。在此之后，.NET已经出现了两个发行版本。每一个主要的发行版本都会对运行时和三款主流的语言（C#，C和Visual Basic）产生巨大的改变，同时也会为客户层和服务层带来许多新特性，如事务的支持（System.Transactions），泛型的支持（同时支持C#和Visual Basic），目录服务支持，管理（WMI）等等。这个故事也远远没有终结，微软甚至计划将一系列新技术应用到其下一个发行版本中（NetFX 3.0，随Vista发行）。一个急速增长的社区也依然在不断扩大，并用开源和商业的新项目以及新构想增强了.NET环境。在这

几年中，在Java和.NET环境之间产生了大量的讨论，大多数的讨论都强烈地倾向于两者中的一方，这几乎没有产生任何作用。毕竟，诸如“我的编程语言比你的编程语言要好”或者“我的平台比你的平台运行得要快”，乃至“你们很逊”这类的话题或许在鸡尾酒会和小组会议上是一个你来我往的颇为有趣的话题，但是这些话题对于引导一个有意义的软件开发是没有任何成效可言的。在经历了立场和姿态上的对立以及互相攻击以后，当尝试使.NET和Java共同工作和对此进行有意义的讨论时，这些对话却又转向了一些诸如“Web服务”、“企业服务总线”或“面向服务的体系架构”等繁杂的词汇中，而没有任何实在的展示。当越过这些高层的讨论去关注底层的细节时，对话中经常出现的又是SOAP、WSDL和WS协议，或者通过消息交换数据，或者在CLR中实现JVM，或者在JVM中实现CLR等。换句话说，来解释这些流行的用语即“你大步迈进并讨论这如何去解决这个问题，但是却从来没有真正得讨论为什么你要这样做”从历史的角度看，关于Java/.NET互操作性的讨论降低到了体系结构的次要位置，仅次于“按需”主题也就是说这种互操作的发生仅仅应该在一个企业同时使用.NET和Java系统，并且需要在它们之间进行对话时。尽管如此，在这个讨论中关于动机问题的讨论被忽视了为什么开发人员想要同时使用Java和.NET？尽管可能不需要这样做。从表面上看来，这是一个危险的主题。因为不是由于对某个平台“不能”做什么的暗示而招致完全的义愤，就是任何关于某个平台可能在某方面“优于”另一个平台的说法都会导致偏爱或无知的谴责。（这甚至忽略了一个基本的问题，即指出这里的“优于”的定

义是什么)。与其无视或躲避这个话题，不如直接面对它。这样的谴责和批评是不应该被忽略的，事实上我们应该欢迎它们，并将其作为一个大讨论的一部分，这个大讨论将解答何时、何地以及如何做出这些决策。尽管这样，我们认为开放式的讨论，时刻检查思想，允许读者形成自己的、批判的观点依然非常重要。本文作为关于Java/.NET互操作性主题系列文章中的一篇，也正是基于此观点进行的。\*\*\* 当在大脑中思索什么是Java和.NET都做的好的方面时，有好几个场景会浮现于我们的脑海并值得我们向前探索。Office客户端，J2EE服务器 微软的Office产品，无论好坏，即使在有电脑的历史以来不是最流行（这里所说的流行是指安装在更多的电脑主机上）的软件平台，也必然是最流行的软件平台之一。目前，Office产品已经迎来了第二个十年，作为一个强大的平台，用户可以输入，维护和查看广泛的、不同来源的数据。对于那些目前依赖于J2EE后台服务器的用户来说，既然有相当数量的数据，那么将这些数据转入Office平台来实现更加简单的管理和考察将是一个很好的考量。更让人感兴趣的是来考察Office平台利用过程无关的通信工具、实现利用Spring或其他轻型Java容器中业已实现的商业逻辑的方式。在2006年8月发行的MSDN杂志发表了数篇关于Office开发的文章（并为此强烈建议任何对于Office编程能力不熟悉的人将此作为背景材料），在以“使用Office作为一个开发平台的须知”为题的一篇文章中，用一个图表展示了Office平台的全部能力。这里我们没有卷帙浩繁地列出完全名单，而是用一块区域简单列出Office可以与Java平台进行良好互动的几点特性：\* 外部自动化。由于COM自动化技术的强大，COM自动化的后继者

， Visual Studio Tools for Office ( VSTO ) ，这个主要的Office ，包括Word、 Excel、 Outlook 和其他应用程序等，组件可以被外部的应用程序接口所驱动，因此，各种Office文档就可以通过一些通用语言从外部创建。拿Excel的强大的图表和计算功能或者Word的强大的编辑和拼写检查功能来说，考虑在Java应用程序如何结合这些功能来实现何种新功能将十分有趣，在服务器上（如一个Web应用程序可以驱动Word来创建一个顾客邮件或者打印由J2EE服务器传入的包含特定数据元素的报告文本，就像使用Velocity引擎填充模板生成HTML的方式一样），或者是在客户端，利用Eclipse富客户端平台，一个已经实现可以作为COM自动化组件的宿主（事实上，Eclipse可以在一个安装有 Office的Windows操作系统里创建Word文档）。当用户仅仅需要查看Word文档而不是创作Word文档时，这就显得尤其重要。微软提供了一个免费的Word查看程序，如果Java的Web应用程序负责创建Word文档，然后通过HTTP协议在网络上传送，这样就可以提供一个比HTML更加丰富的格式。 \* 插件。Office也可以提供插件功能，一些软件组件作为插件在Office的内部运行，通常的情形是将它们自身作为一个主菜单项或者一个上下文菜单。一个。NET组件可以将其自身注册为一个Excel电子表格应用程序插件，使用一些形式的Java互联技术（Web服务，远程调用工具包或其他过程内宿主）来连接Java的商业组件用于验证、数据获取或存储。比如，很多公司使用Excel作为一个发票和/或会计解决方案，而且他们可能使用了一些Java组件作为一个简单的进出公司会计系统的方式。这个会计系统一般是庞大的、基于Java技术的一个应用程序包，运行在一个企业级的服务

器上，通过EJB中的会话Bean提供的Java连接器进行访问。

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](http://www.100test.com)