

VisualBasic的未来预告 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/244/2021_2022_VisualBasi_c97_244441.htm 对于面向对象编程语言的宠儿Visual Basic，微软将作如何的改进？概述：下一个版本的Microsoft Visual Basic 主要有以下三方面的改进：Web Forms、Web services和面向对象的语法方面的改进。WebForms使得经验丰富的Visual Basic用户可以象现在编写单机程序一样简单地开发网络应用程序。通过SOAP 接口，Web services让你在可以联网的任何地方配置你所设计的组件。另外，几个在面向对象的语言方面的关键性的改进使得Visual Basic的代码象C 一样具有灵活性，这几方面的改进包括继承性、多态性和重载。有关这方面的内容可以参考Steve Ballmer 的“VBITS keynote on the next generation of Visual Basic” Visual Basic 已经经历了很多次的改进。然而从它诞生以来，我就一直喜欢它的一点是：就它的核心而言，你仍然可以象1991年一样的编写你的程序。当然，和那时相比它的软件包已经有了很大的增强，但是这些改进一般是补充性的，并没有模糊作为编程工具本身具有的目的，这个目的就是：使Visual Basic能更简单快捷地用于设计、编写和调试出优秀的面向对象的应用程序。当前使用的Visual Basic 6.0 版本引入WebClasses作为一种简化手段，用于配置健壮的网络的应用程序。事实上，WebClasses提供了大量的途径可以通过常见的工具把程序移植到网络上。（关于Visual Basic 6.0在网络下的可伸缩性的详尽讨论可以参见Ted Pattison的“Advanced Basics column”（Microsoft Internet Developer于1999年十月出版发行）最近了解到了下一个版本的Visual Basic计划

采取的一些新的改进。其中主要的改进是在存储容量方面，开发者可以使用的存储容量扩大了三倍。下一个版本的Visual Basic计划将采用Visual Studio环境中叫Web Forms的特性。Web Forms代表着一种全新的组件化的网络解决方案。Web services将成为一种新的基于XML的方案，它通过标准的网络协议发布中间层的事件处理功能。同时，Visual Basic语言将包括一些开发者长期以来一直要求的结构，这使得Visual Basic符合那些C和Java使用者所熟悉的面向对象的编程习惯。在下一个版本的Visual Basic的测试版中，这些改进将会被公布。下一个版本的Visual Basic所发生的变化是令人惊喜的。如果你想获得它们所带来的那些好处，那就使用它们。如果你不想，你可以理直气壮的使用你目前仍然使用的。然而，了解在象Visual Basic这种比C和Java容易使用的多的语言中，也可以实现C和Java所实现的功能，是有好处的。未来的发展趋势这种预览式的介绍你留下了什么样的印象呢？这个问题问得很好，但是你可以找到问题的答案。在过去的一年中，可以明显的看到ASP开发的变化，这些开发程序常常由一些易读的ASP脚本组成，在这些脚本的基础上运行整个程序。由于ASP是对整个脚本代码进行解释执行的，在对各组件进行组装时，人们逐渐发现这种技术的固有的局限性。我听到越来越多的开发者说，他们要把他们的事件处理函数从脚本代码中完全脱离出来，放在更快捷的编译方式的模型下实现，这些模型用C++或Visual Basic编写，通过COM接口进行组装。对于你所能想到的各种理由，Visual Basic都是能够满足的。使用Visual Basic来设计组件实际上并不比使用VBScript或JScript?困难多少。你可以编写执行起来更快的代码，并且

很容易就能达到你的要求。当下一个版本的Visual Basic发布后，你可以使用Visual Basic来生成面向网络的对象，这种对象和ASP兼容。总之，走组件组合的路线不管是现在还是将来都会被认为是最好的选择。正如我前面时候提到的那样，使用Visual Basic(和WebClasses)编写的面向Internet的应用程序已经有很广泛的基础。问题是，大部分的基于WebClasses的应用程序并没有经过很好的设计。它们没有很好地区分应用程序的不同的层次，把中间层的过程和基于DHTML的用户界面混淆了。下一个版本的Visual Basic将引入WebClasses，它是经过精心挑选后确定的网络开发的工具。因为它更具有scalable、更强大、而且是真正的language-agnostic。它在Visual Studio的所有的工具中起作用。如果你注意多层开发的一些基本规则，你可以很容易地完成这个转变。特别要注意，把中间层过程和显示层过程分开。强烈推荐在做这些工作时，参考Windows? DNA 2000的体系结构。核心的事件处理功能必需在中间层完成，你可以使用各种你所喜欢的编译语言编写的用于实现这些功能的各个组件。然后，这些组件组装在一个ASP脚本文件中，这样各组件就可以协同工作了。如果你把大部分的逻辑运算放在事件对象中而不是脚本中的话，那就是最理想的了。它不仅对将来向Web services转变是一个好的主意，它也是一种值得效仿的实践。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