

Microsoft  
Visual Studio */\*CODING完美世界\*/*



# 微软 Visual Studio 2010 技术专刊

《程序员》编辑部 主编

# /\*CODING完美世界\*/

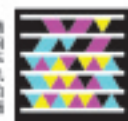
Code，创造了万事万物。几乎任何地方，均有它的存在，正如作为开发人员的你，眼中所见的无限可能。利用Visual Studio 2010的新工具，你设计、开发或是部署的工作方式，都将发生意想不到的变化；你心中的伟大构想，终将以前所未有的方式，付诸实现。

/\*CODE，无处不在\*/  
用Visual Studio 2010打造你的奇迹

立即登录 [VisualStudio.com](http://VisualStudio.com)  
或拨打咨询电话: 800-820-3800 / 021-96081318



用手机扫描



拍摄此标签，获取Visual Studio 2010最新消息  
如手机未安装二维码应用程序，请至<http://gettag.mobi>免费获取 \*信息及数据资费按相关标准收取

# 目录 Contents

---

## 卷首语

- 01 Hello, Visual Studio 2010

## 人物访谈

- 02 微软公司开发事业部Visual Studio团队总经理 Jason Zander专访：  
Visual Studio 2010是怎样炼成的？
- 06 微软(中国)有限公司开发工具与平台事业部大中华区总经理 谢恩伟观点：  
Visual Studio 2010的新机遇
- 08 微软Visual Studio商业软件部总经理潘正磊专访：  
Visual Studio 2010中国团队的研发内幕

## 新特性

- 10 简约而不简单的云开发平台  
——浅尝Visual Studio 2010云开发
- 12 三位一体打造高效酷炫软世界
- 15 在Visual Studio 2010 中实践Scrum
- 17 Visual Studio 2010 架构建模新特性
- 20 Microsoft在进化？  
——外包供应商对Visual Studio 2010测试架构的观感
- 22 Visual C++ 2010 新体验
- 24 免费午餐之后的选择  
——体验 Visual Studio 2010 中的并行开发
- 26 Windows Phone开发工具初体验
- 29 .NET 4和VS2010新技术亮点选析

## 业界观察

- 32 梦想创造可能  
——盘点微软 .NET 技术八年发展历程

策划：《程序员》编辑部

时间：2010.04.12



# Hello

## Visual Studio 2010

**相**信每一个有计算机编程经历的朋友，都记得一个经典的程序：“Hello World”。它可以说是世界上最简单的程序，无论是中文、英文、德文，无论是Basic、C++、Java……几乎所有国别所有编程语种的教材中，“Hello World”总是作为第一个TEST记录于书本之中。几乎每一个开发者，都以这个简单的打印程序为原点，踉踉跄跄地跨出计算编程的第一个步履，懵懵懂懂中，由此进入一个神奇的数字世界。同时，从“Hello World”词语本身的内涵来看，这是一个生命面对世界宣告自我存在的第一声呼唤；它如同一颗新生的种子，伴随着探索未知和构建世界的欲望，正不顾一切的冲破羁绊，实现自我的完善。所以“Hello World”，十分奇妙地匹配了软件开发者的先天精神禀赋：探索未知的好奇心、创造新世界的激情。

软件开发就是这样一种奇妙的职业，开发者通过输入一行行的代码，不仅可以创造自己的人生，而且会改变周围的世界。“工欲善其事，必先利其器”，那么软件开发者手中之剑，自然是开发工具了。所以不难理解为什么说2010年4月12日会是一个十分具有特殊意义的日子。因为业界瞩目的微软下一代软件开发平台——Visual Studio 2010将全球正式发布。这令无数中国的开发者翘首以盼，同时他们也难免心生揣测：Visual Studio 2010对于开发者究竟意味着什么？对软件工程的发展将有哪些影响？我们中国开发者的机会究竟在哪里？

众所周知，微软每推出一款产品，总离不开它对当今世

界日新月异的商业和技术趋势、信息应用市场需求特征的战略判断；所以要想理解Visual Studio 2010的精髓，必须将它放在微软整个软件产品体系、世界整个技术应用发展潮流的坐标系中考量。为此《程序员》编辑部深入采访了几乎软件业内每一个热点领域的第三方资深专家。他们将就自己精通的领域，以“旁观者清”的状态，结合Visual Studio 2010 相关新功能，逐步地勾勒出一幅“下一代软件研发平台”的全景图。当然，我们同样会关注这款产品的“当事人”——微软公司，包括它的Visual Studio 研发团队。他们对行业的洞察、对产品的设想，以及中间经历的实现过程，相信都会令读者们对Visual Studio 2010产生更加全面和深入的理解。

微软将Visual Studio 2010 全球发布会的主题定为“CODING 完美世界——Life Runs On Code”。它寄托着这样的希冀：希望开发者们能通过Visual Studio 2010 构建一个绚烂多姿的世界，并实现完美的程序人生。这是否只是一个理想，还需要时间和实践来验证。根据微软官方透露的统计数据：Visual Studio 2010 Beta 2版本截止2010年1月底，在中国下载已超过10万人次，不仅居全球第二位，而且也刷新了国内下载非正式版软件产品的记录。这个数据无疑也寄托了广大中国开发者们对微软发布Visual Studio 2010正式版的期待：

## Hello Visual Studio 2010

# Visual Studio 2010 是怎样炼成的

——微软公司开发事业部Visual Studio团队总经理Jason Zander专访

■ 记者 / 常政 高松

**微**软公司将Visual Studio 2010正式版全球发布的日子定在2010年4月，而截止1月底，VS2010 Beta 2版本国内下载量已超过10万人次。这充分说明了这款开发利器，在中国开发者们心中的份量之重。毋庸置疑，VS2010的诞生意味着世界软件工程的重大变局，所以能否有效地把握VS2010的精髓，以迎接全新的机遇和挑战，对于中国的开发者尤为重要。为此，本刊记者第一时间采访了微软VS2010研发团队的总负责人Jason Zander。他将为大家揭开 Visual Studio 2010 的神秘面纱。

**记者：**您好，Zander先生。首先请您对数百万中国开发者做下自我介绍。

**Zander：**我是Jason Zander，微软公司开发事业部Visual Studio团队的总经理。作为总经理，我的团队主要负责Visual Studio Express、Professional和Visual Studio系列产品的开发，涉及的技术包括：编程语言、Javascript运行与工具、IDE与生态系统、Office与SharePoint工具、云工具、资源控制/工作项目跟进、高级架构、开发人员以及测试工具。

作为CLR的初始开发者之一，我在技术领域的主要工作包括：文件格式、元数据、编译器、调试/剖析，以及将系统集成到Windows和SQL Server等主要微软产品中。在担任Visual Studio总经理之前，我是.NET框架团队的总经理，并在微软开发过多个产品，其中包括SourceSafe、Repository和ODBC。1992年加盟微软之前，我在IBM公司的罗切斯特实验室致力于分布式SQL和SQL/400的开发工

作。我在密歇根州立大学获得计算机科学学士学位。业余时间，我喜欢和三个孩子一起玩耍，并在自家的木工房里做家具。我的博客地址是：<http://blogs.msdn.com/jasonz>。

**记者：**倍受全球开发者关注的VS 2010正式版将于4月12日发布，能否介绍一下微软开发本款新产品的主要情况？

**Zander：**微软在开发过程中乐于使用自己的产品——我认为指出这一点是非常重要的。开发事业部约有3000名员工，分别负责工具、.NET架构和运行等领域的工作。我们在世界各地都有开发团队，但是主要的技术开发工作集中在4个地点（美国华盛顿雷德蒙、美国北卡罗莱纳州Raleigh、美国夏威夷以及印度的Hyderabad）。TALM业务约有400名员工。微软的ALM产品包括：协作中心、Team Foundation Server（TFS），以及用于测试、架构和高级开发任务的各个专门版本的Visual Studio。将Visual Studio Professional作为主要开发环境的用户可通过授权并安装名为Team Explorer的TFS插件，在Visual Studio 2010 Ultimate或Premium（在这两个版本中可使用Visual Studio Team System的最新功能）团队项目中进行协同办公。

## VS2010的核心价值

**记者：**即将发布的VS2010正式版能够完全实现你们最初设定的目标吗？

**Zander：**完全能够实现。很高兴告诉大家，微软的Visual Studio 2010是一个有助于简化从设计到部署等整个开发流程的集成环境。现在，开发人员和测试人员能够：

- 通过能将愿景变为现实的强大的原型、建模，以及设计工具，释放自己的创造力。

- 个性化自己的环境并将目标锁定为使用人数日益增长的平台，如：Windows 7、Windows Server 2008 R2、SQL Server 2008、Microsoft SharePoint，以及可加速编码流程并让开发者能够利用现有技能的云应用程序。

- 由于有了可让用户迅速找到并轻松修复错误、确保高质量解决方案的集成测试与调试工具，开发人员和测试人员能够更高效地工作。

而对于企业级用户，Visual Studio 2010 也提供了大受欢迎的热门新功能：

- **确保高质量编码：**IntelliTrace这样的新功能让测试人员和开发人员能够选择使用手动及自动测试方式，以及高级调试工具，以便维护源代码并查找错误。

- **通过集成提高简洁性：**IDE和编辑器的功能显著增强（如：Generate from usage）意味着：无论是建模、编码、测试，还是调试，开发人员都能利用现有技能来部署越来越多的应用程序类型。

- **释放创造力：**Silverlight和WPF的新拖放绑定功能、Windows 7和SharePoint的内置式工具，以及与创新技术的集成（例如：数据库、UML、Expression、ASP.NET、MVC和多核）使开发人员能够将远景目标变为现实。

总之，带有MSDN的Visual Studio Ultimate中强大的ALM功能，再加上Visual Studio Premium和Professional中强大的集成综合工具，将使开发人员



微软公司开发事业部Visual Studio团队总经理  
Jason Zander

把更多精力用在有创意的开发上，花在解码上的时间也会相应变少。

**记者：**在VS2010所拥有的这一系列热门新功能中，您认为最突出的优势是什么？

**Zander：**显然，并没有简单的答案，这取决于成熟团队目前的实践、技能和限制。采用ALM的三个最重要的优势在于：能够专注于客户的价值流、在流程中减少浪费、在企业利益相关者之间实现透明度，以利益各方在最关键的时刻做出明智的决策。汽车行业在过去半个多世纪所推崇的精益管理也发生过同样的变革。那些成功实现变革的公司从不起眼的小公司发展成为全球市场的领导者，而没有进行变革的公司则纷纷走向破产的深渊。

通过Visual Studio，我们已经采取了措施，专注于那些我们能够最大程度改进价值流、降低浪费、提供透明度的领域。我们提供了在线工具，以便帮助个人用户评估其成熟度、在投放市场之前发现潜在可改进的地方、降低成本并提高客户满意度。我们还对员工及合作伙伴进行了培训，以便他们利用这些理念帮助客户制定变革路线图。投资回报

率通常可以立即在项目中体现出来，但是改进可能会持续数年，因为这涉及企业运作方式的变革。

## VS2010 的挑战和经验

**记者：**在VS2010的整个开发过程中，您所面临的最大的挑战是什么？同时，您最珍视的经验是什么？

**Zander：**说起开发Visual Studio 2010时的挑战和经验，有几件事令人印象深刻。

首先，针对使用Team Foundation Server (TFS) 的小团队，我们在开发中进行了特别关注。TFS 2010引入了一个新的基础配置，在安装时即可使用。安装流程本身已经简化了，现在只需30分钟就能完成安装向导和配置。我们的意图是让那些可能并不需要所有功能并安装所有配置的小型团队能够更方便地使用TFS。基本配置可支持Windows 7、Windows Vista和SQL Server Express，并且不需要服务器操作系统、SharePoint或完整的SQL Server。

其次，有人问我们，为什么UML现在成了Visual Studio的一部分。过去十年，UML往往被认为是MDA（模型驱动架构）的一部分。我们并没有沿着老路走。相反，我们希望关注编码的开发。UML类图可视化编码是Visual Studio 2005的组成部分。在Visual Studio 2010中，我们引入了更多功能。例如，用户可右键点击一个方法来生成一个序列图，该序列图将确切展示该方法是如何被调用的。我们还能超越UML，将那些符号中并不包含的问题显示出来。最好的例子就是架构分层，在那里你不仅能够生动地看到逻辑组件之间的依赖关系，还能执行分层规则，以阻止反模式（如循环混乱）随着时间的推移，无意间进入你的团队项目。

最后，在听到客户反馈之后，微软决定在Visual Studio 2010中引入一个简化的SKU结构，这将使用户更果断地做出采购决策。客户只要购买了MSDN和Visual Studio 2010，就能期

待产品带来更大价值。

根据开发人员的具体需求，每个级别的版本都有其各自的优势。未来，客户将从Visual Studio 2010的三个主要版本中选择自己所需：

- 带有MSDN的Microsoft Visual Studio 2010 Ultimate:

是一套适用于团队的综合应用程序生命周期管理工具，确保从设计到部署的质量。

- 带有MSDN的Microsoft Visual Studio 2010 Premium:

是一套适用于开发人员的完整工具组，提供可扩展、高质量的应用程序。

- 带有MSDN的Microsoft Visual Studio 2010 Professional :

是一套适用于基础开发任务的基本工具，使开发人员能够轻松实施其理念。

我们建议开发人员访问[Microsoft.com/visualstudio](http://Microsoft.com/visualstudio)，从而了解更多关于SKU的详细信息，以及其它可用Visual Studio产品（如：Team Lab Management）的信息。

**记者：**我们注意到VS2010团队本次采用了敏捷研发模式。能否和读者分享一下你们敏捷应用的具体情况？

**Zander：**对于Visual Studio 2010，我们在敏捷工具上进行了大量投资。新的Excel Agile Planning Workbook让团队可以轻松地采用Scrum等敏捷的软件开发方法。Excel Agile Planning Workbook提供与TFS的双向同步，而团队可以用它创建和管理用户故事及产品特性列表，评估团队的效率并细分项目工作。产品特性列表让团队能够规划各项工作并跟踪进度。此外，Team Foundation Server 2010附带针对Agile Software Development v5.0流程模板的Microsoft Solutions Framework。此流程模板可以定制，它定义了一套工作项目、报告和控制面板，供团队在规划和跟踪项目时使用。

**记者：**在整个VS2010研发过程中，您觉得最值得回忆的故事是什么？

**Zander:** 一个有趣的故事发生在购买计算机的时候。我们当时努力说服销售代表卖给我们最低质量（指的是性能）的上网本，以便在2月份推出的RC版本上测试性能。有趣的是，销售代表却卖力地向我们推荐一款速度更快但价格更低的计算机。但是，我们的目标是看看低性能计算机是否能够很好地支持RC版本。最终，我们购买了性能较低的计算机，但性能仍表现良好！

## 软件开发的未来

**记者：发布VS2010之后，你们接下来会关注什么项目？**

**Zander:** 这个问题很难回答，但对我们最重要的是用户反馈。用户的反馈在各个Visual Studio版本的开发中起着重要作用。

云可能是很好的机会。在进行云开发时，架构师和开发人员必须在目前的开发实践中融入下一代的想法、设计、编码和测试。Visual Studio提供各种关键的创新，让云计算——软件加服务——成为现实。

首先，Visual Studio 2010提供了为应用软件创建详细的架构模型所必需的工具。及早关注架构可以防止在后期编程时出现众多严重的编程错误。对于在多个层（演示、应用、数据，可能都在本地和/或远程）上存在的应用软件，全面详细地了解各部分如何连接至关重要。不幸的是，由于没有生命周期管理工具，很多应用软件在设计时都没有完整的架构模型。

云应用也将必须在传统IT防火墙之外的环境——经常是第三方网络——中保证质量和可靠性，并且与第三方服务和数据相连。这极大地增加了软件测试的复杂程度。同样，高效的生命周期管理和使用Visual Studio Test功能可以为这些场景设计测试流程，创建工作项目以跟踪这些场景中的所有变量，并且让团队的每个成员都能够轻松地访问有关这些场景的报告。

最后，云模式为地理上分散的开发与部署带来大量机会，但同时也会造成源代码和不同版本的管理十分混乱。Visual Studio的核心是Team Foundation Server (TFS)，它为位于各个地点和不同公司的开发团队提供单一的服务器解决方案。

Visual Studio Ultimate/Premium为应用程序生命周期管理提供完整的解决方案，让用户得以开发能够在多种软件加服务模式部署的应用程序和服务。这还利用了.NET框架，让Windows软件能够在各种设备上运行，比如PC、Web、移动设备以及电视、游戏机和Windows Media Center等家庭娱乐系统。

还有新的用户界面，这毫无疑问是很重要的技术趋势，而微软一直在这方面居领先地位。已经在Xbox上展示的Natal项目就是一个很好的例子——这种游戏界面让用户成为游戏杆，而不再需要手持设备。在语音等很多情况下，新用户界面确实需要实现我们的软件加服务愿景。你需要具备极佳本地计算能力的软件，以及在云中不断积累的全球数据库以调节识别的精确度。

Visual Studio 2010已经支持多点触控的托管与本地代码的开发。随着更多的新界面功能出现在我们的平台上，我们将在未来版本的Visual Studio中为开发人员提供这些服务。

**记者：能否向中国的开发者们预测一下软件开发与工具的未来？**

**Zander:** 在回答这个问题时，我想谈一下我对应用程序生命周期管理（ALM）的看法。未来五年，微软认为ALM会超出目前的边界，作为一个类别不断成长和扩展，最终成为所有开发工作的事实标准，这是因为越来越多的机构意识到需要管理开发过程，能够跟踪和报告开发工作，并能够让不同的业务部门和IT实施部门在多个项目上进行更好的合作。ALM工具会扩大应用范围，而其中的关键是在生命周期

的各个阶段进行协作。这将不仅受到软件开发周期的限制，还将扩展到包含生命周期的方方面面——从业务价值的定义与识别到软件和系统的交付与维护到业务价值的支持。ALM目前被认为是支持软件交付的线性过程，而我们认为未来会变成环形，随着企业不断变更战术和战略以及IT部门的应变，这个过程将充满反复和变化。

ALM是动态IT的基石，让IT更加灵活、更具成本效益且更加高效，通过流程自动化、降低复杂性和提高对业务需求的响应能力来优化企业的系统和人员的协作。ALM就像胶水，让IT部门能够通过软件加服务的方式把现有和新的系统投资整合起来，实现差异化并帮助用户开发集成且敏捷的基于服务的解决方案，从而实现新的应用并提高业绩。通过工具和技术把项目管理、运营和工程团队联系起来，推动信息流动、报告、治理和项目可见性，提升工程和用户体验，从而增强整个IT业务的效率。

有多种因素推动着ALM在未来五年的增长。软件交付的质量继续是关键因素，而随着业内整体质量的提升其重要性会越来越高。可预测性要求和规章制度压力迫使机构需要更好地跟踪软件开发和交付，推动机构采用更加正式的可跟踪流程。日益流行的离岸和全球开发只有通过好的ALM工具才能高效地控制，确保所有参与者高效协作，还需要实施集中协作，让各个地方的人都能够围绕单一的共享资产工作。我们还看到生命周期中引入了其他人士，包括业务流程分析师、设计师和UX专家、部署与运营团队，甚至终端用户，他们都将参与到意义更广泛的软件开发之中。

**记者：谢谢接受《程序员》的专访。最后，对于即将使用VS2010的开发人员，您最想说的一句话是什么？**

**Zander:** Visual Studio 2010有益于所有开发人员——不管组织规模、开发人员的偏好或项目需求有何不同。■



# 《程序员》

## 非常春天非常有礼

还等什么现在就开始规划你的知识大道吧，  
相信我们的携手相伴会让明天更加灿烂。

在春季预定阳光，让你的思想绽放，相信总有一天我们会用青春的代码改变世界。

《程序员》在这个春季为答谢新老读者在十年间对杂志的关注与陪伴，特举行春季订阅特惠活动。



即日起至6月30日，凡订阅《程序员》杂志全年或以上的读者均会获得精美礼品。礼品有限，送完为止。

订阅一年期  
送T恤一件



订阅两年期

- 办公助手(笔记本+精美便笺)
- 《压力下的角逐》\*
- T恤2件
- 三种礼品任你选



订阅三年期

2G U盘+T恤一件



《程序员》订阅价格：**180元/年**

老读者续订除赠送上述礼品外，还**加赠**一期《程序员》。  
机会不容错过，赶快行动吧！

注：本活动解释权归《程序员》杂志社所有

详情请登录：<http://book.csdn.net/programmer/>



\* 赠品介绍：《压力下的角逐》

微软 Xbox360 和索尼 PS3 处理机芯片首席设计师戴维·希普，为您揭秘复杂项目的管理技巧和过程，电子巨头的利益与政治角逐，告诉你 PS3 与 Xbox 诞生的真相……

杂志社读者服务部订阅：

咨询电话：010-64351436 010-51661202-176/501 传真：010-64348545

Email：reader@csdn.net 在线咨询 MSN：reader@csdn.net

在线订阅：<http://book.csdn.net/programmer/>

邮局汇款：

汇款地址：北京市朝阳区酒仙桥路 14 号兆维  
工业园 B3 区 2 门 1 层

收款人：《程序员》读者服务部

邮编：100015

银行汇款：

开户行：中国银行将台路支行  
户名：北京创新乐知广告有限公司  
帐号：802119627508091001

Programmer  
程序员

# Visual Studio 2010 的新机遇

■ 文 / 谢恩伟

**中**国是全世界最重要的开发工具市场之一，这主要体现在两个方面：首先，中国拥有150万专业开发人员，占全球第二位。其次，中国拥有250万IT专业人员，很多高级IT专业人员在程序编译或者办公软件的自动化方面都需要用到相应的工具，所以IT专业人员群体对微软具有非常重要的意义。同时，由于政府政策的扶植，软件

外包以及服务行业正在迅速拓展，中国的软件业正面临着前所未有的发展机遇。在这种形势下，Visual Studio 2010的发布，将会迎来一个新机遇。

## 中国软件开发的三大趋势

分析CSDN在《2009年中国软件开发开发者年度调查报告》中的数据，可

以发现中国的软件开发正呈现出三大重要趋势。

第一个趋势是中国的软件业正表现出极大的Web开发潜力。这通过国内众多门户网站的建设情况也能得出结论。从微软的角度来看，Web应用也有很大的发展空间。现在业界流行的RIA概念，旨在让互联网的应用更加丰富多彩。在这方面，微软也作了很多努力，不仅涉猎到Web的应用，而且还关注如何将Web应用做到像桌面应用软件那样为用户带来丰富多彩的体验。

第二个趋势就是云计算。从CSDN的调查数据同样可以看到，云计算的发展正势不可挡。有35%的软件开发希望在一年内使用云计算服务，市场需求之大可见一斑。

第三个方面，中国用户对整个应用软件端到端的开发流程管理的了解程度正在逐渐深入。怎样综合利用现有的资源，使得流程更加系统化？传统的应用软件开发往往就是编码的开发，怎样将其过渡到更加理解客户的需求，然后再做测试、部署以及维护？这是一个端到端的系统工程，未来会有很好的发展趋势。

总而言之，我们可以看到Web应用、云计算的拓展，对ALM以及对整个研发流程管理专业化的需求提升是非常明显的。这种趋势下，回到产品层面，微软会有什么行动？在Web、RIA方面，如何通过我们的Silverlight把Web应用达到一个新境界？在云计算方面，如何有一个更好的开发工具，为下一代的在线应用软件服务？怎样为专业软件开发的发展提供好的应用？最新发布的Visual Studio 2010，将会给你一些答案，其背后体现的不仅是微软对技术发展趋势的回应，更是软件工业的未来潮流。

下面，我们从Visual Studio 2010所带来的新机遇、新特性和新动力三个维度，进行详细叙述。

## 新机遇：一键实现部署云计算开发

当前，微软最新的战略和愿景是“三屏一云”，即怎样通过互联网、以软件为主体，通过不同的设备为广大客户带来无缝体验。主旨就是研究怎样将互联网和数据中心结合在一起，为广大的在线服务提供更好支持，并在此基础上，为用户提供无缝体验。

RIA的概念，让互联网应用变得可以像桌面应用软件那样为用户带来丰富多彩的体验。在新发布的Visual Studio 2010中，我们用全新的WPF进行设计，开发者借助它就能更好地实现在线服务的应用软件开发以及丰富的互联网交互应用。

对于云计算，也可以理解为通过标准化IT，通过应用软件平台、基础设施和Internet支柱这三方面的约束软件，为用户提供有序的、能自主服务的



微软(中国)有限公司开发工具与平台事业部大中华区总经理谢恩伟对Visual Studio 2010在中国应用的市场前景充满信心

在线应用软件的体验。根据这个定义，如果这三个方面成为云计算的基础：软件、应用平台、基础设施，那微软具有相当大的优势。在应用软件层面，微软有Exchange、SharePoint、CRM等；在基础设施方面，微软同样有很好的产品，如Windows Server 2008，最新的System Center和虚拟化，而且对服务器的管理、系统管理以及数据中心的自动化，以及整个流程都有很大的支持。2009年9月，我们做了一个关于Windows Azure和SQL Azure在线软件开发平台的演示，这两个产品我们已经有了测试版本并会在短期内发布。

有了开发平台，开发工具就占据非常重要的地位。微软在现有的Visual Studio的基础上将在线应用软件开发无缝地进行实施。Visual Studio 2010不仅提供有云开发前端的指示，在后端对云计算也提供了一站式服务：只需要点击一个按钮，开发完成的应用就会快速地发布到Windows Azure这个计算平台。此外，从Azure的应用软件开发过程中，用Visual Studio 2010以及Silverlight 4的很多现成功能，就能快捷有效地搭建出这种平台。

### 新特性：Scrum敏捷实践

对于应用软件，我有15年的开发经验，我自己对Visual Studio的开发流程也非常熟悉。基于以往的经验，我的观点是：软件开发不只是一个技术问题，更是一个流程问题、一个质量保证的问题、一个针对客户需要的问题。2002年，在Visual Studio团队到上海工作之前，我便参与了整个Visual Studio 2010的设计工作。为了能真正把应用软件开发流程带动起来，我做了很多Visual Studio的具体调研。结果发现这样一个现象：很多开发人员花80%的时间用于软件的修改和更新，只有20%的时间用在用户体验和新功能的开发上。这显然造成了人才的巨大浪费。那怎样能够让一种程序非常简单地维护、修改，而让开发者能花更多时间去展示他们的创新能力呢？敏捷是一个解决方案。

敏捷开发到底是什么？我的理解是，就是怎么把项目的透明度提到更高的一个层次。比如说怎么能够把产品的生产流程，从以前的瀑布式增加，做一个阶梯式的调整再去更新？怎样能更好把用户的反馈纳入规划？在现有的开发能力下，如何做出更高效的工作？怎样更好地提高大家的工作积极性？回想早年间自己做开发的时候，漏洞修补就是一个非常头痛的事情，整个软件开发也是一个非常繁琐的流程。与此同时，更多的软件开发者希望用新的技术，在新的用户体验上有更新拓展。在敏捷开发模式下，这些情况会有很大改观。所以说，敏捷能够带来更多的益处，简单而言就是提升整个项目的透明度，提高个人项目开发的积极性。

另一方面，如果把敏捷作为一个理念，那Scrum就是

一种实践、一种应用方式。Scrum有一些比较核心的应用，此外，它还把项目管理作为一个核心，使得项目开发变得更加有序和可控，能在短时间利用迭代模式进行更新。Visual Studio 2010对敏捷开发会起很大作用，况且它自身的研发也是以敏捷开发的流程来进行。在Visual Studio 2010中，微软把Scrum作为基本的Agile开发模型，TFS2010中集成了MSF for Agile Software Development v5.0，在可操作性上融合了敏捷等软件开发流程的思想模型。

从2005年开始，微软就在进行敏捷实践。在应用了敏捷之后，我们发现产品的漏洞数量有显著地减少。就内部而言，微软本身也是自己最大的客户，Team Foundation Server 2010已经在微软内部做了布置，目前有三千多个非常活跃的用户在上面活动，这包括了从比较小的Web应用到比较大的企业级应用软件开发在内。

### 新动力：并行计算和C++

在多核并行运算方面，Visual Studio 2010做了非常大的改进。在2008年，微软就有并行运算的能力，内部专门组建了团队，针对多核并行运算做大量工作。新发布的Visual Studio 2010将集中体现我们对多核并行运算方面的全新支持。

Visual Studio 2010的另一个新动力就是C++。国内C++用户很多，但在过去的5年中，微软在C++方面的进展一直比较缓慢，但Visual Studio 2010却做了很多突破和尝试。我们在上海就有专门的团队在做这方面的深入研究，相信定会大大满足国内市场的应用。国内的现状是开发者从编译、嵌入式研发，到应用软件开发，都需要用到C++平台。在最近发布的Visual Studio 2010中，除了语言层面开始支持最新的C++标准外，在IDE以及MFC库等方面，我们也下了很大力气。Visual C++不仅改善了IDE的效率，同时也引入了很多新的特性，这些IDE增强能够切实地提高程序员们的开发效率。

### 结束语

随着信息应用需求的复杂性逐渐升高，广大开发者们在大量已有系统的兼容性、平台的演进、开发语言的升级等方面正面临着巨大的挑战。Visual Studio 2010 Beta 2版本截止2010年1月底，在中国的下下载已超过10万人次，居全球第二位，刷新了国内对非正式版软件产品下载的记录。这也说明目前国内超过150万的开发者们，正在积极寻找应对的方法。我相信4月12日Visual Studio 2010正式版的发布，一定会令大家欢呼雀跃。■

（编者按：本文根据谢思伟在Visual Studio 2010媒体技术预览会上的发言及本刊专访资料整理而成。）

# Visual Studio 2010中国团队的研发内幕

## ——微软Visual Studio商业软件部总经理潘正磊专访

■ 文 / 霍泰稳 常政

**随**着2010年4月12日微软官方正式颁布“Visual Studio 2010”的日期临近，无数中国开发者翘首以盼，津津乐道微软下一代软件开发平台的全新功能。但并不是很多人知道，有一群中国研发人员在Visual Studio 2010的诞生过程中，起着举足轻重的作用。这是一个什么样的团队？他们正以什么样的方式在工作？为此，我们采访了该团队的领导人——微软Visual Studio商业软件部总经理潘正磊。

### 研发历程：从“瀑布式”到“迭代式”

**记者：**请简单介绍您自己和正在从事的工作？

**潘正磊：**我1992年大学毕业后直接进入微软工作。十多年来，我的工作性质和职位发生了很多次变化。最早做开发程序员，参与Microsoft Access的开发，之后转做微软第一款针对网络Web的开发工具Visual Interdev；接着到Visual Basic团队担任开发经理，见证了微软从VB6到VB.NET的艰苦转型。之后在Visual Studio部门担任了一系列职务，包括开发总监、Visual Studio Team Architect全球产品总经理、Visual Basic全球产品总经理。目前我的职位是Visual Studio商业软件部总经理，主要负责面向企业的开发工具；同时兼任微软亚太研发集团服务器与开发工具事业部联席总经理，领导开发工具部门的中国研发团队。

**记者：**从您身上，几乎可以映射出微软研发团队的技术变迁史。请问这些年来，纵观微软产品的开发过程，主要经历了哪几个比较重大的阶段？

**潘正磊：**微软的产品研发，这些年来确实经历了很多重要变化。记得刚进微软的时候，包括Microsoft Access在内的很多微软产品还属于第一代。那时使用计算机的人数不多，相对来讲用户需求也较少，因此微软的产品已经基本满足了用户的需求，通常情况是只要产品发布，就有用户来使用。但如此一来，就造成了新版本产品的开发瓶颈。如何在满足基本用户需求的前提下，使自己的产品更上一层楼，这是我们当时考虑最多的问题。所以那个时期，我们团队的最大竞争者不是别人，而是上一版产品。这种情况下，我们采用了当时很流行的Waterfall（瀑布式）研发模式，即由我们预先定义用户的基本需求，然后销售给用户，在当时市场背景下，这种模式应用得十分成功。

但到了2000年左右，这里面隐藏的弊端慢慢凸现了。客户的满意度逐年降低，使得微软的研发一度陷入低潮。2000年以后，我们发现原来那种“做一款产品——收集用户反馈——再做一次升级”的模式，已经不合时宜了。我们开始改变策略，采用了“迭代式”的开发模式，即事先并不先入为主地确定用户的需求，而是先做一些prototype（原型）试验品，先给关键用户去体验，再根据用户的反馈意见不断做修改和调整。在整个研发流程中，我们会发现产品的最初设想和最终设计往往不同，这种对用户反馈需求的注重，逐渐成为我们在研发中所秉承的最基本理念。众所周知，微软的研发团队都很大，所以在贯彻这些全新理念的时候，我们也面临了许多挑战。

微软大约花了7、8年的时间成功完成了这样的转型。以Visual Studio



微软Visual Studio商业软件部总经理潘正磊拥有近20年的软件研发经验

产品为例，从2008年开始，我们大量推出了CTP（Community Technology Preview，社区技术预览版），与我们的客户、开发人员进行了非常透明的交流。我们会将产品定义和方案设想在网上公布，邀请很多MVP（微软最有价值专家）、客户进行点评。直到现在，我们也会花很大精力做这样的工作。

### Visual Studio 2010的中国制造

**记者：**Visual Studio 2010即将发布，您能否透露下这款产品包含哪些新功能？

**潘正磊：**从更宏观层面讲，Visual Studio 2010中有几个特点，值得大家多关注。最大的特点是Visual Studio 2010完善和强化了对开发平台的支持力度，全面支持Windows 7、Windows Azure、Office 14、SharePoint 14、SQL、SQL Azure、Silverlight、.NET 4.0等等所有微软的开发平台。

在平台支持方面，值得瞩目的亮点还有两个。一是C++产品功能在Visual Studio 2010里有大幅提高。如

果在Windows 7中做开发，不论并行运算、调试、还是IDE，都有很多新功能。第二是在SharePoint 14平台上增加了很多新功能。SharePoint是微软最近几年成长最快的一个服务器产品。我们Visual Studio 2010跟Office 14、SharePoint 14团队共同打造了一个非常强大的平台和开发工具，时间上基本同时发布。这个产品为开发人员在SharePoint平台上做二次开发提供了非常便利的功能。此外，.NET 4.0在整个.NET Framework中也增加了很多新功能，包括WPF、WCF、WF等领域，感兴趣的读者可以亲自体验。

除平台支持外，Visual Studio 2010的新特性还体现在团队开发管理和实施敏捷流程方面。通过Visual Studio 2010，可以有效地统筹整体项目成员，包括开发人员、测试人员、管理人员，甚至架构师等等，大大提高整个团队的集成度。从Team Foundation Server产品到更完善的架构师工具、测试工具，还有诸如面向开发人员的IntelliTrace（以前称为Historical Debugging）等等工具，其新增的功能和特性，都非常值得期待。

**记者：Visual Studio 2010中一系列靓丽的新功能，有哪些是我们中国研发团队贡献的？**

**潘正磊：**实际上，Visual Studio 2010是我们微软中国开发工具部研发团队所参加过的最大的开发项目，我们团队从中积累了许多经验，并主持了好几个重要模块的开发，这包括UML工具、Silverlight、WPF Designer里的一系列功能、对SQL Azure的支持、WCF和WF开发工具、以及Windows Form对.NET 3.5和.NET 4.0的支持等等。在并行开发工具的研发过程中，特别是调试和性能分析等领域，我们也起到了主导作用；对于使用C++工具的开发人员来说，在对项目进行编译时，会发现新版本的Link效率有显著提高——这也是由我们中国团队实现的。

## Visual Studio 2010 的敏捷精髓

**记者：听说Visual Studio 2010是用敏捷开发的模式完成的，对于其他想基于Visual Studio去实施敏捷的团队，能否谈谈你们的经验和建议？**

**潘正磊：**第一，我们受益于使用了的Team Foundation Server产品。我们充分利用Team Foundation Server的特性，管理源代码，做每日集成，整合和资源，自定义工作流程，管理项目进度。在每个milestone（里程碑）之后，根据TFS里汇总的实施情况，来优化下一个里程的计划。

对于企业用户来说，Team Foundation Server在2010版本中有一个很大的改进，就是对“跨平台”的支持。以前大家都认为它只适合用.NET或者C++的项目，其实Team Foundation Server上有很多Web Services和API。不久前微软收购了Teamprise公司，他们就曾经用这些API做了一套客户端，可以分别在Mac、Linux、Unix等平台上运行。开发者常常会遇到这样的项目：用户会用Java做服务器端，用.NET做客户端。此时如果把两个项目分开管理，会很繁琐。但通过Team Foundation Server，就可以把它们整合在同一个服务器；而且Team Foundation Server本身和Microsoft Project有非常好的集成，这会给管理者提供很大的便利。

第二，需要指出的是，有了好的管理工具，并不意味着它可以取代理管理者的的重要性。在一个团队，管理者始终要起主导作用。一个高效的团队，团队成员之间，需要有一种合作的精神，需要一个共同的目标。管理工具只能帮助促进这样的过程，而管理者本身工作的难度并不会完全减轻。因为你每设计一个目标，都需要团队的认可，需要大家能够全心全意地、目标一致地前进。这样的精神和力量，并不是靠工具就能简单实现的。

最后一点，在用Team Foundation Server进行敏捷开发的时候，要时刻记

住敏捷最重要的精髓，不要让过程化的东西妨碍了自己的思维，不能受条条框框的束缚。实施敏捷最重要的是，要根据不同的用户场景、需求，灵活地调整实施方案。这个项目的成功方案，并不意味着另外一个项目可以完全照搬，你还需要考虑不同的因素：如客户需求，工程师的素质和特长等，这些因素都会导致实施方案的不同。敏捷是用最好的方法帮你完成项目，它需要和用户有非常多的交流，能帮助你的团队迅速团结一致地朝一个明确的目标行进，这才是敏捷的精髓。

## Visual Studio 2010 的研发体验

**记者：你们团队曾参加了多个不同Visual Studio版本的开发和发布，您觉得完成Visual Studio 2010开发，同以往相比最大的不同是什么？**

**潘正磊：**跟以往不同的是，在开发Visual Studio 2010的时候，我们不只做了很多功能，而且我们和用户间的交流比以往更多更强。无论是刚开始的Beta 1还是Beta 2，或是RC的发布，我们和很多用户都有非常直接的沟通与交流，在产品的开发过程中及时解决了不少用户反馈出的问题。我自己感觉这是最大的不同。

其次，由于新款的Visual Studio 2010增加了很多新功能，不可避免地造成了软件本身的庞大，导致相应的运行速度有所降低，所以我们花了大量时间做Visual Studio 2010的性能调优，以确保产品的性能能够像Visual Studio 2008一样好。■



### 作者简介：

霍泰稳，InfoQ中文站总编辑，有多年的软件开发经验和媒体从业经历，以技术传播为己任，关注企业软件开发领域的变化与创新。曾先后参与《程序员》杂志、《MSDN开发精选》杂志、《开源大本营》图书和《开源技术选型手册》2008版图书的策划编辑工作。

# 简约而不简单的云开发平台

## ——浅尝Visual Studio 2010云开发

■ 文 / 王翔



### 作者简介:

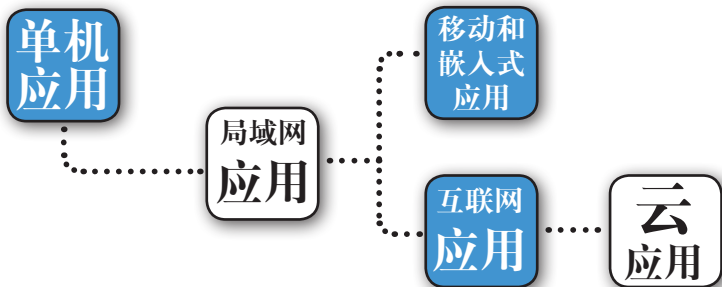
王翔, 全国海关信息中心开发部高级架构师, 从事海关主要广域分布式系统的设计和实施, 多次参与各业务系统的优化。工作之余喜欢旅游、写作和烹饪。

Visual Studio 2010无疑掀开了微软开发平台的全新一页。区别于Visual Studio 2003、2005、2008三个版本的渐进式完善, Visual Studio 2010配合.NET Framework 4.0有了脱胎换骨的变化。新的平台在更符合行业标准的基础上不仅提供对动态类型、并行计算、F#更好的支持, 开发云应用更是一个显著的亮点。

近两年, 云计算的热潮此起彼伏, 很多声音来自存储、服务器、简单的NoSQL访问, 并且误导大众云计算就像自来水、民用电的取用一样, 但这只是云计算的准备阶段或初级阶段, 就好比给您一台没有浏览器、没有Office和其他文档阅读、编辑的半裸机一样, 而真正为您定制的应用或您用于获利的应用还“飘”在地上。相对而言, Azure平台更贴近应用, 它提供的不是大众无法直接使用的“基础设施”, 而是各类实实在在的应用软件, 只不过“飘”在云上而已。

如果用软件运行机制划分, 云应用则是站在一个新的起点, 而Visual Studio 2010无疑为开发此类应用提供了快捷方式:

本质上, 云计算需要屏蔽掉物理区域、承载



节点、网络拓扑等因素向用户透明地提供服务, 而Azure恰恰提供了这些内容:

- 具备其他云服务商的基础环境能力。
- 整合了Windows Live、Dynamics和Windows Server产品, 其中很多功能非常适合商业环境, 类似功能对于ISV或个体开发者而言如果从头开始做, 技术上难度过大, 而且即便技术上能完成也错过了商业机会。

- 充分保护现有开发技能投资, 使用Azure的开发者可以继续把绝大部分本地WinForm应用、ASP.NET应用的经验运用到Azure中。

不过, 尽管Azure上述特点提供了一个较为可靠的运行环境和API体系, 但用户自己开发的云应用能否稳定可靠运行几乎是另一码事, 就好像同样一个基于SQL Server 2008类似功能的系统, 有的开发者可以支持200个并发用户, 有的只能支持10个并发。据笔者经验, 开发云应用在实现阶段的步骤上有些特别之处:

- 前期, 通过原型系统验证API。考虑云应用开发在业界还比较有限, 出于知识积累的考虑, 建议采用非抛弃型原型较好, 便于团队学习、理解、沉淀相关技术。不过, 现阶段也有些不利因素, 由于云应用开发尚不甚成熟, API的非兼容更新经常发生, 为此即便是原型也建议运用设计模型技巧, 在部分关键API处预留后手。

- 然后, 基于威胁建模分析安全性, 对原型进行必要的重构和简单的渗透测试, 无论采用CIA (保密性、完整性、可用性: C=Confidentiality, I=Integrity, A=Availability) 的防御模型还是SDL的STRIDE (Spoofing user identity, Tampering with data, Repudiation, Information disclosure, Denial of service, Elevation of privilege) 攻击模型, 开发云应用必须先过安全关。另外, 必须慎重权衡HTTPS的使用范围, 这里范围包括三个维度“功能”、“时间”、“数据量”, 做到“刚刚够用” (No more no less), 否则代价就是真金白银。

- 接着, 开发功能并通过必要的单元测试、集成测试确认功能有效。

- 下面是配置云运行环境, 根据不同资源情况下的运行表现在应用中留出“活扣”, 便于实际部署后可以根据吞吐率有先手布局。这步非常关键, 因为不同于我们以往的企业应用或互联网应用, Azure平台允许我们进行类似的模拟, 做类似的装载回归其成本很低。

- 最后, 根据预估的资源使用情况, 选择合理

的租赁套餐。相对以往的开发，在云平台部署也有不少优势，毕竟装载回归阶段很多租赁已基本量化。

（注：另外，云应用还需必要的“非云”运维监控机制保证，还要考虑相应的本地信息备份能力，否则“亡羊”都没处“补牢”。）

上述5个步骤结合Visual Studio 2010均可以较为便捷的完成。不过对于熟悉WinForm应用和ASP.NET应用的开发人员，如何使用Azure存储机制需要一个适应过程：

- **Blob Service**：虽然提供二进制信息的存储，但最好慎用，对于二进制的多媒体信息而言，采用该服务成本偏高，尽管官方有类似的标杆系统，但考虑到资费以及国内调用效率，建议租用或自建独立的流媒体服务器和文件服务器，调试阶段也须慎重安排调用次序和资源访问路径。

- **Table Service**：非常有用，但使用中建议考虑自己开发一些“土法”编码的JavaScript、XLS函数，在信息提交前将稀疏的用户数据进行处理，在展现时进行反向处理，例如：一条微博内容为“大胜印尼9个球！！！！！！！！！！”，不妨在提交前处理为“大胜印尼9个球/[112/]”，虽然有些烦琐但考虑到Azure的收费方式，能省还是给自己省点。之所以没有直接提压缩算法，同样因为Azure的收费方式，您可以根据应用的内容特点，权宜计算时间和存储空间。

- **Queue Service**：是个容易被忽略但其实更容易出彩的服务，不仅仅限于向Work Role发送消息。一方面通过他的异步处理能力，常常可以在相同Host Service使用的情况下支持多并发用户；另外，在处理结果（包括：查询结果）交付方面也有一定灵活性，减少因为多个处理流程争用资源产生的无谓支出，毕竟本地数据库死锁等待一段时间后Kill一方这个处理，资源消耗相对较小、成本低，类似问题出现在云库（毕竟现阶段SQL Azure的锁处理不如本地SQL Server的完善）消耗可是实实在在的费用。采用Queue Service可以通过分隔、分工处理流程的方法，将争用面缩小，让您的云应用可以“悠着劲儿”的完成处理。

上面关于三个存储服务开发、调试均可在Visual Studio中完成，而且区别于调试本地应用习惯看Task Manager，开发云应用不妨直接盯着计费“斤斤计较”。同时，云应用的调试能力是Visual Studio 2010的一大亮点，您之前的ASP.NET Web Service及WCF调试经验可以直接搬到Azure平台，而且调试信息可以直接显示在您Visual Studio 2010

的Console窗口中。此外，启动调试的过程和本地应用相差无几。Visual Studio 2010预置了C#和VB.NET的Azure项目模板，借助模板和向导您可以省去很多“八股”内容的编写，将注意力集中在业务功能上。

另外，对于致力于基于Azure平台从事大型或长期云应用项目的团队而言，Visual Studio 2010 IDE的扩展能力（VSX: Visual Studio Extension）很值得研究。

- 对于那些准备通过包装Azure相关服务，进而对外提供Open API（或者是商用API）的团队，为了便于用户使用您的API系统，不妨基于VSX提供额外的项目模板或者插件，尤其是提供定位较为准确的错误反馈和组织比较系统的调试信息。另外，为了便于编码方便，可以扩展QuickInfo Tooltips，便于用户及时、直观地了解API的内容。相信在第三方云应用Open API新秀还不算丰富的今天，如果您能先走一步提供一套开发人员友好的Open API,也能帮助您新的平台占据领先。

- 如果您想结合Azure发布商用服务，不妨在说明文档之余准备一些Code Snippet Library，一方面便于用户使用您的商用服务，也便于向下游开发者提供“不出格”的示例。不仅如此，考虑到云应用部署方面相对本地应用过程上烦琐些，建议扩展MSBuild，便于您的下游集成商打包调试、部署他们的系统。

如果您直接向用户提供基于Azure的前端应用（WinForm或ASP.NET、甚至是类似JSF、PHP的其他平台），那么不妨用Visual Studio方便的调试功能先开发“胖服务端”，为前端提供更为丰富Facade Interface的同时，借助缓存、压缩、并行处理等技术尽可能的节省资源使用，进而降低运营费用。

整体而言，尽管微软通过Azure的开发包尽最大可能降低云应用开发门槛，但毕竟这个平台还很年轻，虽然理论上您可以借助其他IDE环境完成类似开发工作，但相对Visual Studio 2010还有一定差距。不过，Visual Studio 2010的云计算功能也存在对其他云服务供应商支持不足的问题，云应用开发环境整体还处在诸侯割据的战国时代。RESTful虽然通用但毕竟成本较高，使用上相对IDE环境有较大改善的Visual Studio 2010就好像蜀道。

云计算是不会总停留在基础环境服务这个层面，在主力厂商和大批中小规模用户的推动下，云应用（或称为基于云的软件）预期会更具附加值，从灵感到产品的周期也更短。使用得当，Visual Studio 2010则是实现该目标的利器。■

[编者按] 本文结合现有一些项目的实际体验，简单地介绍了Windows 7，Visual Studio 2010，Silverlight 4三位一体在打造酷炫软世界方面令人振奋的一些新功能。

# 三位一体打造高效酷炫软世界

■ 文/卢开所

## 梦想的软世界

2009年11月18日，在洛杉矶进行的微软PDC开发者大会上，Steven Sinofsky（微软Windows及Windows Live 部门总裁）宣布IE9浏览器将使用Direct2D和DirectWrite引擎，通过GPU（图形处理器）加速网页渲染，这将极大地提高浏览器的图形及文本的显示质量、渲染速度和增强用户体验。在展示会上，Bing地图在IE8中每秒大约可渲染14帧，而经过硬件加速的IE9则可达到每秒60帧。这一振奋人心的好消息，预示我们期盼已久的酷炫软世界即将来临。

那么，什么是我们所梦想的酷炫软世界呢？第一当然是功能，满足功能需求永远是第一位的。第二是效率，这里效率包含两方面的含义——速度加智能化。速度要快到不要让用户有等待的感觉；智能化是要让用户有享受“傻瓜办大事”的乐趣。第三是UI，要酷要炫，要让用户感觉到“养眼”，让用户来享受这个世界而流连忘返。

总之一句话，我们梦想的软世界是“既要跑得快，又要很精彩；便我很傻，但办事顶呱呱”。

## 追逐梦想的艰难

在软件世界里，Windows操作系统经历了Win3x，Win9x，Windows NT4/5，Windows XP，Windows Server系列，Windows Vista到今天的Windows 7（这里我只列出了具有代表性的部分Windows操作系统）。从这一发展历程我们可以看到，其发展更新的速度是越来越快，功能是越来越丰富、完善、友好和高效。从追求简单的图形化到实现了多任务，多线程，并行化，网络化，并引领的外围硬件来支持Windows的风潮，使Windows 7成为了软件发展的龙头。

Windows展示技术的发展大概分为如下四个阶段：1985~1991年，C搭配Windows API；1992~2001年，C++配合MFC动态链接库（这一时段，Visual Basic也得到了广泛地应用）；2002~2006年C#配对Windows Form（包含ASP.Net）；2007年以后，C#搭配WPF开始得到了广泛地使用，其中经过改装了用于Web的WPF就成为了我们今天说的Silverlight。从这一发展历程看，网络化的酷炫趋势也是有目共睹的。

同样地，微软的开发系统也是一样，从C7，VC1.X，VC++5/6，VS200x，到现在的VS2010。流行的应用系统从

单机到C/S系统，再到B/S系统，再到云计算，等等。此外，它还从单一语言发展到了集成多语言，从简易功能发展到了涵盖设计、建模、工作流、协作开发、测试、维护等高度集成一站式开发系统。

软件业经过几十年的井喷式发展，按其他行业的发展历程来讲，应该从起步阶段发展到了平稳行驶阶段了，到了收获的季节了。但理想和现实总是有差距的，让我们来看看现实的软世界吧！

越来越多的人抱怨软件不够友好、界面单一，网络速度太慢、反应不够及时，海量的数字信息中无从选择、找不到需要的信息，娱乐成分不够或是娱乐没有面向不同层次的用户群体等等；同时，软件开发的需求是越来越复杂，软件维护体系是越来越庞大、越来越艰难。鉴于此，甚至有人大声地喊出了“软件危机”四个字。

面对无限膨胀的需求，下面来看看我们是如何用Windows 7、Visual Studio 2010和Silverlight 4来推动软件事业的发展，来打造我们所梦想的酷炫软世界的。

## 什么是我们所梦想的酷炫软世界？第一是功能，第二是效率，第三是UI。

## 追逐梦想的行动

让我们来看看，我们的软件能不能“既让马儿吃得少，又让马儿快速跑”。为什么要选择Windows 7、VS2010和Silverlight 4来打造我们理想的酷炫软世界呢？这还得从这三位“贵客”所具有的绚丽特性说起。

Windows 7新特性琳琅满目，其中与开发相关的新特性也很多，这里我们就简单介绍几个和我们打造酷炫软世界相关的新特性。

## 增强的图形功能

Windows 7推出后给大家最大冲击是其震撼的用户体验。大家是否有这样的感觉，Windows 7显示和切换对话框又快又炫。我们不会像Windows XP那样有一点点单调的感觉，我们也不会像Windows Vista那样炫而缓慢，Windows 7在用户体验方面做了巨大的改善，这主要是Windows 7



改变了Vista的DWM显示方式，取而代之，采用了全新的API、Direct2D、DirectWrite以及Direct3D 11新技术，支持图形处理器（GPU），总结了图形图像及文本显示的经验，优化了性能，改善了结构，这些都大大地降低了内存的占用，极大地提高了渲染的速度和效果。我们有一个图像测量系统，在其他操作系统下运行时，因为要适时捕捉、跟踪、处理动态图像，为了达到适时的效果，我们对图像进行了预处理，先转化为灰度图来提高测量速度，但在Windows 7下，我们就没有必要再转换了，其本身的速度已经满足了适时的要求，这使得我们在测量窗口看到的就是物体真实的彩色图像，相比较以前的黑白灰度图，真实的彩色图像极大地提高了用户的测量体验和使用感受。

## ClearType技术

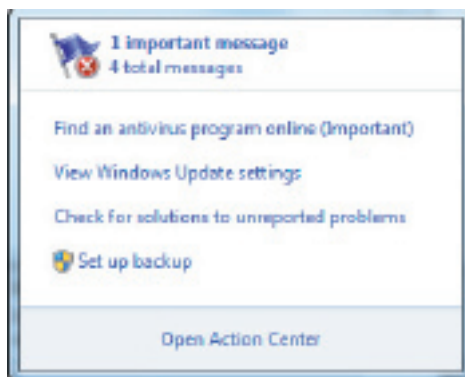
ClearType是微软推出的提高显示和阅读效果的一种新技术，其采用亚像素(Sub-Pixel)技术来平滑显示，提高文字的清晰度。在工业生成中，亚像素技术已得到了广泛地应用。如在我们开发的图像测量系统中，为了提高测量精度，我们采用了亚像素技术，使得我们的测量精度提高了一个等级。Windows 7采用亚像素显示技术后，其显示效果和用户感受也同样提高了一个数量级。举例说，当显示一条倾斜的直线时，如果不采用亚像素技术，其显示效果就像台阶式电梯，一个台阶一个台阶的；如果采用了亚像素技术，其显示效果就像滚梯一样，既平滑，又没有跳跃。据统计，一个计算机用户花在计算机上的80%的时间是阅读文本，所以在这方面的提高对计算机用户来说是一个很大的福音，同时这是硬件软件化的一种趋势，在不增加硬件成本的基础上用软件来提高显示品质。

## 自我诊断与修复系统

操作系统在安装初期和后期维护过程中，会有各式各样的硬件错误，对一个普通的用户来说，要诊断和修复这些问题是强人所难。但在Windows 7中，却为我们提供了这种可能。在Windows 7新增了“自我诊断与修复系统”的新功能。能够自动分析、修复并验证客户使用电脑遇到的问题。其涵盖了从视频、音频、网络连接、打印任务到性能优化等十多个问题。一个普通用户可以通过点击鼠标和键盘就可以解决很多以前需要专业人员才能解决的问题。而且除了内置的解决方案外，用户还可以联网接收最新的疑难解答方案。

Windows 7的新功能还很多，这里就不逐一介绍，有兴趣的用户可参照Windows 7 相关文档。

从前面Windows展现发展的历程



自我诊断与修复系统

看，Silverlight 4肩负着微软拓展Web市场的重担。下面我们来介绍一下Silverlight 4与我们打造酷炫软世界密切相关的一些新特性吧。

## Silverlight 4部分新特性

### COM操作

Silverlight 4提供了对COM访问的支持，这对开发来说是一个大大的福音，我们不能孤独地生活着，我们要和现有的组件交流对话，我们要访问别人开发好的组件。Silverlight 4为这种交流提供了很好地支持，如我们要客户端的Excel应用程序，只要写上如下一句话（需要信任授权），就可以操作桌面程序了。

```
dynamic excel = ComAutomationFactory.  
CreateObject("Excel.Application");
```

### 支持客户端摄像头、麦克风、打印机

在当今的媒体世界里，图像和语音的交流也是必不可少的，Silverlight 4也为这种提供了强有力地支持，同时编程也是很方便的。对打印输出，就像Silverlight 4是客户端桌面系统一样（其实Silverlight是Web程序），连编程看上去都一样，同样地使用PrintDocument来完成打印任务。

### 支持Net.Tcp协议

支持Net.Tcp协议是一个巨大的好消息，它极大地改进了吞吐量（对于UI线程来说提升了5.5倍，对于worker线程来说提升了870倍。）和连接数（服务器可支持的连接客户端数量是之前的5~6倍）。对以前版本的Silverlight,大家对其酷炫的用户体验是深有感触的，但是其速度总是有那么一点点慢。Silverlight 4在速度方面有了很大的改善。我们有一个社交网站项目，其主要目的是公司内部对同一主题具有相同兴趣爱好的员工的一个自动交流平台。采用Silverlight+WCF 开发，因为主题采用自动化捕捉，所以需要适时查询相关主题感兴趣的群体，并希望相关用户具有友好的用户体验。以前的感觉总是速度有点慢，特别是高峰时间，对一些热门主题，用户能感到比较明显的等待，但在采用Silverlight 4升级后，我们采用了Net.tcp协议和双工模式后，速度和用户体验上都有了明显的提高，用户的操作非常流畅，没有任何等待的感觉。

### 丰富了数据绑定

如果没有数据绑定，开发中将会增加很多的重复代码。Silverlight 4丰富了数据绑定，简化了用户更新维护数据的操作，提高了开发效率。

讲了这些Silverlight 4的新特性，下面让我们转投Visual Studio 2010，也来简单介绍几项对我们打造酷炫软世界紧密相关的新特性吧。

## Visual Studio 2010部分新特性

### 即用即生成

即用即生成，就是需要用时再去生成相应的类、方法、属性等，这比较符合人们的思维习惯。大家对极限编程XP，敏捷开发Agile，及Scrum、TDD等当前软件开发的热门话题都耳熟能详吧，甚至有人号称这些是解决软件危机的良方妙药。可是从我们周围的项目情况看，这些很好的开发模式并没有得到很好的应用，究其原因，其主要是没有很好的工具支持。不过现在VS2010对其提供了很完善地支持，即用即生成对TDD开发提供了强有力地支持。

### HTML Snippets超越智能感知

智能感知Intellisense省去了我们记忆的痛苦，但HTML Snippets（智能感知到后，按2个Tab键）进一步帮助我们自动添加对象的一个默认整体，进一步提高了书写代码的速度。

### 动态类型Dynamic和动态编程

随着Web应用的风行，动态语言也开始展露锋芒，如PHP、Python、Perl、JavaScript。为什么会出现这样的风潮呢？因为任何时候都难于预测未来，对于程序来说，其同样难于预测下一个对象，难于预测下一个类型，为了适应这种动态的需要，动态语言应运而生，VS2010也顺势在C#4中加入了动态类型和动态编程。

```
dynamic word = ComAutomationFactory;  
CreateObject("Word.Application");  
word.Visible = true;
```

大家可以看到，在程序运行后，Word才知道自己是Word对象，同时我们就可以顺利地访问其相关的属性、方法了，真是很神奇。

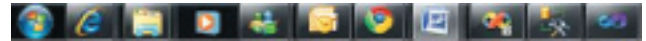
### Quick Search

现在的软件项目，不论大小，其结构都分得十分清晰，代码量都不会很小，如果这些都需要开发人员记住其中的内容，那对开发人员来说绝对是一种摧残，Quick Search（Ctrl +显示其搜索对话框）为开发人员提供了快捷而模糊的查询方法，帮助开发人员很快地定位到需要查找的内容。

从我们上面讲述的各项新特性来看，有一个共同的感受，就是为了提高开发效率和增强软件质量。VS2010是我用过的开发器中最完美的一款了，可以说VS2010就是为Windows 7和Silverlight 4而生的（有点儿夸张，但反映了其间紧密的关系）。VS2010拥有大量的与WPF相关的新功能，包括新的可视化设计器，拖放式数据绑定，在XAML中

智能感知标记扩展，这些功能结合WPF新的任务栏类，使用最新版本Visual Studio 2010中的强大功能，让你可以简单、容易地构建最新的、酷炫的Windows 7应用程序。

Windows 7+VS2010+Silverlight 4三位一体，使我们的开发变得如鱼得水。大家对Windows 7的酷炫工具栏体会颇深吧，使用Windows 7+VS2010+Silverlight 4开发这一绚丽的功能真是变成了小case，如下几行代码就搞定，真是享受开发的乐趣吧。



```
<Window.TaskbarItemInfo>  
<TaskbarItemInfo x:Name="TaskbarItemInfoExperience"  
Description="Customer Browser: Using WPF 4 on Windo  
ws 7" >  
</TaskbarItemInfo>  
</Window.TaskbarItemInfo>
```

## 梦想成真的快乐

我们并没有列举Windows 7、Silverlight 4和Visual Studio 2010的所有增强特性，这里只是和大家分享了一些文本、图形显示渲染等相关的新特性；从上述体验看，Windows 7、VS2010和Silverlight 4三位一体的开发让我们体验了酷炫的软世界，让我们的部分梦想变成了真。

Windows 7为我们提供了一个得力的“管家”，并且是一个能够自我管理、自我维护、自我学习的管家；Silverlight 4为我们提供了一个绚丽的舞台；VS2010为我们提供了一个世界级的编程器，使得我们每个开发人员都具有导演一部《阿凡达》那样真实、震撼、三位一体软世界的的能力。

## 梦想没有尽头

同时，我们也体会到“路漫漫兮其修远”，我们对梦的追求是无止境的。我们在兴奋快乐的同时，继续追逐着更酷更炫更快更智能。从智能感知到智能化，甚至智慧化；从矢量字体，3D翻转到真正的数字3D模型……

如果有一天，我们一想到，软件就能帮我们自动实现，甚至我们没有想到，软件也能引导我们并实现我们所需要的软世界，那才是我们梦想的“Dream Land”。■



### 作者简介：

卢开所，宇思信德科技(北京)有限公司C++、.NET架构师，对桌面和Web应用软件有深刻了解。在数据拟合、数字图像、USB&1394、自动化开发和Windows Mobile开发方面有深入的研究，热衷于前沿技术的探索。

# 在Visual Studio 2010中实践Scrum

■ 文 / 段君毅

2001年2月由17位世界轻量级方法学家提出了一份敏捷联盟宣言，这个宣言只是简单的四句话，但却是敏捷方法的精髓，也是对敏捷的高度抽象，这便是敏捷之道的最高境界：

- 个体与交互胜过过程与工具
- 可以工作的软件胜过面面俱到的文档
- 客户协作胜过合同谈判
- 响应变化胜过遵循计划

我们知道敏捷开发并不是本身的方法，而只是一个统称。在敏捷宣言签署后出现了很多敏捷方法论，这些方法包括Scrum、XP、Crystal、FDD和DSDM等等。综观这些方法论我们发现敏捷的主要思想就是简单。如太极，简单中透着智慧。实行起来如行云流水般，任意伸缩，如手中无剑，心中有剑的境界，其中蕴藏着一种“至简至易”的意境和思想。敏捷团队注重简易，这样做可以消除那些没必要的复杂。现如今，敏捷开发已经在很多方面取得大的进展，在很多公司的项目中已经得到了比较好的验证，微软也将敏捷开发引入到MSF解决方案中，形成了MSF for Agile Software Development 过程框架，来指导开发团队使用敏捷方法进行项目开发。

MSF for Agile Software Development 过程框架从4.2升级到5.0，并且是以著名的Scrum模型为基础导向扩展，并且结合了VS2010和TFS2010工具的众多特性，从而成为微软.NET相关技术人员手中不可多得的利器。

Scrum最初的含义是英式橄榄球队，是敏捷软件开发模型中的一种。Scrum 将软件开发团队比拟成橄榄球队，有明确的最高目标，熟悉开发流程中所需具备的最佳技术，具有高度自主权，紧密地沟通合作，以高度弹性解决

各种挑战，确保每天、每个阶段都明确的朝向目标推进。Scrum是一个敏捷开发框架，是一个增量迭代的开发过程。在这个框架整个开发周期有若干个小的迭代周期，每个小的的迭代周期称为一个Sprint，每个Sprint的长度2~4周。在每个Sprint中，Scrum的开发团队拿到一个排列好优先级的需求列表，我们称它为为用户故事或者叫Sprint Backlog，所以我们先开发的是对客户具有较高价值的需求。在每个迭代结束后，都会完成可交付的产品。

在Scrum中的Product Backlog通过很多个冲刺来实现，放入MSF以后则改变为通过很多个迭代实现，在每个迭代中包含很多个User Story，每个User Story又由很多个相关Task类型的WorkItem来实现。如图1所示。



图1 MSF中的Scrum

下面我们就从Visual Studio 2010和TFS2010中对于Scrum方法实现所支持的特性做下分析：

## 产品清单（Product Backlog）

Scrum开始时，产品负责人（或者客户）通常需要创建一个产品清单用来描述客户需求，在产品清单中我们要写清楚每个需求的特点以及在完成后如何去演示。需要注意的是，我们客户可能使用任何工具来创建产品清单，如Word，Excel等。对于团队来讲就是要分析清楚这些内容。为以后的冲刺计划和开发作准备。通常来讲创建产品清单

是非常重要的，它直接影响到团队人员对于客户需求的理解，在此基础上才能够进行工作任务的细化和编写。

在VS2010中，给我们提供了一个很好的Product Backlog的Excel模板来方便地管理我们的Product Backlog。在模板中有的下三个工作表组成：

- **Product Backlog:** 使用此表来筛选职级和优先需要管理的User Story。可以指定的Story Points 和为迭代分配User Story。
- **Iteration Planning:** 使用此表安排安排的迭代，审查每个迭代的工作量，并确定如何平衡整个迭代的工作量。
- **Interruptions:** 此表来指定小组将不执行任何工作的节假日或其他日期。

我们在项目开始阶段，就可以使用上面的模板来填写Backlog，进行迭代计划，工作量平衡，在这个模板中，Visual Studio还提供给我们可以与Visual Studio 2010同步的功能。下面我们来看一个实例：

我们建立一个产品Product Backlog（示例），如表1所示。

ID	名称	重要性	初始估算	如何演示	注解
1	用户创建新帐户	30	4	打开注册页面，填写注册信息，提交注册，返回注册成功	需要检查用户邮箱是否存在
2	用户登录网站	10	2	输入用户名密码点击登录，返回用户信息页面	需要记录用户信息
3	用户添加商品到购物车	20	5	登录，点击商品，点击商品下方添加到购物车	

表1 产品 BACKLOG（示例）

我们可以把Backlog存放在TFS2010服务器上，或者共享在TFS2010的Excel或者Project文档里面，这是为了多个用户可以同时编辑它。一般来说项目成员习惯采

用Project和Excel来编写Product Backlog, 项目成员可以通过Office提供的团队协作功能, 将文件中的内容同步到TFS中然后通过VS2010显示出来。虽然正规意义上这个文档应该归产品负责人所有, 但是我们并不想把其他用户排斥在外, 开发人员常常要打开这个文档, 弄清一些事情, 或者修改估算值。

通常情况下我们编写Product Backlog都是通过文档类编写工具来实现, 实际上在VS2010中除了给我们提供了一个Excel模板以外, 它本身也支持Scrum的Product Backlog的编写, 并且可以进行Backlog的迭代, 我们在Team Explorer中点击右边的Product Backlog后, VS2010会给我们列出当前我们已经定义的Product Backlog。

我们需要对这些Backlog进行冲刺计划, 也就是我们要分阶段对其进行迭代开发。通过VS2010中提供的迭代编辑对于User Story进行分块, 进行计划迭代。

### 一种新的WorkItem: User Story

使用VS和TFS的开发人员知道, VS中把工作分成每个单独的工作项, 形成一个工作项的数据库保存在TFS2010中, 相应的TFS2010提供了对报表服务的操作, 工作项和报表都会被保存到一个数据仓库中。在项目工程中, 这些东西都可以被抽取出来进行展示, 而在MSF For Agile 5.0中总共定义了6种类型的WorkItem 如下:

- **User Story:** 定义用户角度的产品特性。
- **Tasks:** 工作中工作任务, 如设计、开发、测试。
- **Test Case:** 定义测试用例。
- **Bugs:** 当测试人员完成测试后, 发现该功能有BUG, 则提供。
- **Issues:** 定义对项目有影响的风险。

VS2010与TFS结合通过WorkItem的编写形式来作为编写User Story的模

板, 我们可以在项目的WorkItem中方便地创建它。在User Story模板填写过程中有很多内容需要填写, 其中最主要的内容如下:

- **Title:** User Story的标题, 尽可能的表达出User story的要点。
- **Stack Rank** 设置User Story的优先级。
- **Story Points** 实现这个Story需要多大工作量, 即需要多少个工作任务来实现, 如: 实现一个登录网站的User Story, 需要做设计界面, 开发程序, 测试等工作, 那么就要将这些工作任务作为Story Points来填充进来, 便于以后跟踪。
- **Iteration:** 这个Story属于第几个迭代。我们可以事先做计划好几个迭代来完成整个项目, 如可以假定通过四个迭代完成项目, 那么在编写User Story时候就要指定这个User Story需要在哪个迭代中完成的。

在创建新的User Story时候, 除了要填写上面的一些主要内容以外, 在implementation栏目中需要填写实现这个User Story的任务, 如开发任务和测试任务等。

我们在上面的示例中计划进行4个迭代, 打开第一个迭代计划后VS2010就会列出包含在这个迭代下的所有User Story和工作任务。

类似于编辑Product Backlog一样, 对于迭代计划中我们同样可以通过Excel表来同步, 而且, VS2010在Excel表中还给我们提供了对于当前迭代中工作任务状态的报表展示, 用来帮助我们更好的了解当前迭代的计划状况。

### 丰富的报表展示图

在项目执行过程中, 通常我们通过图表展示项目目前状态来对项目进行跟踪。TFS2010在项目创建的时候就为我们创建了一个SharePoint的项目站点, 打开项目站点后, 就会看到很多由Web Part编写的Dashboard被展现在项目站点中, 如图2。

我们注意到在这个项目网站中



图2 Visual Studio中报表展示

有很多的图表显示, 其中最重要是任务燃尽图。从上图我们可以看到在TFS2010中通过Excel Web Services展示的燃尽图。它非常直观的体现了当前工作任务的状态, 通过可视化的展示方式来显示团队所有工作任务需要的总时间, 以及过去四周的工作成果。除此之外在TFS2010中还提供了强大的报表显示系统, 提供了Crystal 和Excel 两种工具, 为项目团队进行丰富的报表显示, 如Bug状态报表、User Story状态报表、Build状态报表等。

### 总结

在新版的VS2010和TFS2010中, 通过MSF for Agile Software Development V5.0实现Scrum思想以及实施方法, 以便于.NET管理和开发人员能随心所欲的应用在自己的项目中, 从而构建出高效敏捷的软件开发团队。需要注意的是, 在Scrum下的一些线下活动, 如每日站立会议、每个冲刺的会议等, 对于项目来讲, 我们要清晰的了解客户的需求, 正确的定义Product Owner, 书写清晰的User Story, 这样才能更好地发挥MSF指导模板的作用, 加大项目成功的几率。■



### 作者简介:

段君毅, 现任惠普公司高级软件架构师。微软最有价值专家(MVP)。具有多年的软件开发和项目管理经验。多次担任微软TechEd大会, MSDN Webcast、微软.NET俱乐部讲师。曾为多家大型企业提供基于微软技术的解决方案顾问, 曾带领过团队进行多个大型跨国项目的研发。

# Visual Studio 2010 架构建模新特性

■ 文 / 高阳

**随**着软件项目规模越来越大，为了支持更大规模的企业级、甚至世界级软件开发应用，同时对相应规模的软件设计应用进行建模协作集成管理，作为微软旗舰开发产品，Visual Studio 2010增加了一系列建模设计新特性，架构分析模式、建模设计模式、模型集成扩展等方面都有突出的进步。

微软在最初的时候，项目中的技术人员使用一些单独的开发工具进行软件开发，例如，软件架构师使用Rose或者Visio进行架构建模设计，开发人员则使用Visual Studio进行编码；项目中的非技术人员则使用一些通用的项目管理工具进行项目管理，例如，项目经理使用Project，Excel，Word等制订项目计划任务分配等。Visual Studio经过十年左右的发展真正主角已经不再是作为编码工具的VS IDE，而是ALM的核心TFS（Team Foundation Server），它使各个功能之间，互通能力大大增强。Visual Studio 2010可以利用各种工具辅助每个关键环节进行管理（ALM）是其重要特性，架构建模设计是其重要的环节。

## 对特定领域语言的增强

Visual Studio 2010对特定领域语言(DSL, Domain Specific Language)的进一步支持，架构设计人员在VS2010中可以进行架构与建模设计，从而摆脱了传统使用非VS架构设计工具，使得设计师可以更有效率地进行.NET架构设计工作。

特定领域语言(DSL)是一种特别用来描述某一专业领域内涵的描述语言，也是架构方面的一个重点。DSL是站在领域专家的高度，而非软件开发专家来解决问题，与UML不同的是用来解决问题而不是描述问题。它是软件开发中出于某特定考虑而设计的小的、目标明确的语言，作用是使用某特定领域里惯用的符号或者表示法对该领域的实体或者流程进行建模，这也是区别于那些提供统一符号表示的通用建模语言。另外一个区别是UML经常被用于文档分析或者设计交付，而领域特定语言既可以用作建模工具，还可以用作代码生成器。

微软的DSL工具是微软针对特定领域开发而专门设计，并提供了一种图形化的操作方式。它包含在Visual Studio 2010 SDK（单独安装）中，允许开发人员自行设计专属的图形化工具，内置了模型的相关支持，以及模型与图形之间的支持，还包括对模型的验证、规则、事务的支持，同时还允许开发人员在结合VS.NET的一些扩展VSIX（Visual Studio Integration Extension）一同使用。VS.NET现在的类设计器，分布式系统设计器(Distributed System Designer)，LinqToSql设计器，Entity Framework设计器都是基于VS.NET DSL开发。VS2010新增了UML Modeling Project，提供了对类图，时序图，用例图等的支持，这也是基于VS.NET DSL来实现。我们在开始创建一个领域特定语言（DSL）时，是围绕DSL设计器进行的，设计器的功能是设计和产生DSL或者设计依附于可生成代码DSL的模型。如图1所示。



图1 DSL设计模型

完整的DSL设计包含两个阶段：第一阶段是为DSL语法设计相应的图形表示。简单地说，必须为语言设计图形、连接器和底层的元模型（包括文字性属性）；第二阶段是写代码模板，即使用存储在DSL中的实例（图表）信息产生对应的代码。

## 引进图形化建模设计工具

在系统需求确定之后，我们便进入基于UML的面向对象的设计（Object-Oriented Design, OOD）系统设计阶段，在这个阶段中，将通过以下UML图来呈现VS2010在设计阶段的支持和应用。

Visual Studio 2010在架构设计方面新增了对统一建模语言(UML)的支持，例如类图、用例



## 作者简介：

高阳，微软ASP.NET MVP，复旦大学软件硕士，10多年管理与程序设计经验，7年.NET应用程序开发经验，对于微软.NET企业应用开发与架构设计有较为深入的理论基础与实践经验。著有《我也能做CTO之程序员职业规划》、《.NET软件设计新思维——像搭积木一样搭建软件》和《深入体验Moss实践开发》等。

图、活动图和时序图等；以及架构层图(Architecture Layer Diagram)，并用图形化的方式描述系统架构，从而使得项目中的技术人员或非技术人员都能以模型通过图形化的方式进行协作，以及定义企业与系统功能关系等。通过VS2010的Architecture工具模板可以创建UML图（Architecture—>New Diagram—>UML Activity Diagram, UML Class Diagram, UML Component Diagram, UML Sequence Diagram, UML Use Case Diagram），图设计器是全新的默认行为的DSL工具。

VS2010的Architecture工具包括以下UML图模板：

- 活动图（Activity Diagram）：活动图是状态图的一个变体，用来描述执行算法的工作流程中涉及的活动。活动状态代表了一个活动，即一个工作流步骤或一个操作的执行。活动图由多个动作状态组成，当一个动作完成后，动作状态将会改变，转换为一个新的状态。
- 类图（Class Diagram）：最基本的元素是类或者接口，描述的对象是系统的类结构，包括每个类属性和方法(成员函数)。它显示出类、接口以及它们之间的静态结构和关系，以及各个类之间可能存在的各种关系，比如继承关系等，并用于描述系统的结构化设计。其实施的关系在许多方面，如数据库表，XML节点，或软件对象组成。
- 组件图（Component Diagram）：是一种用来设计系统的静态实现视图的图形，通过组件图可以表现出一组组件间的组织和其之间的关系。组件图也是UML的建模图形中用来设计面向对象系统实体角度的图形之一。它描述了软件组件之间的关系，包括组件间的依赖、通信、位置以及其他条件，比类图更加抽象。
- 顺序图（Sequence Diagram）：又叫时序图显示多个对象间的动作协作，重点是显示对象之间发送的消息的时间顺序（强调的是消息的时间顺序）。它主要用途为设计系统的动态视图，是UML里两种交互图的一种，在UML中的交互图主要为了显示交互。
- 用例图（Use Case Diagram）：是参与者的外部用户所能观察到的系统功能的模型图，呈现了一些参与者和一些用例，以及它们之间的关系，主要用于对系统、子系统或类的功能行为进行建模。它描述了从一个外部的观察者的视角对系统的印象，强调这个系统是什么而不是这个系统怎么工作。

此外，Visual Studio Ultimate 2010新增层图（Layer Diagram）可视化系统的逻辑结构设计功能。层图的目的是指定的各个架构层次，使系统具有更清晰的体系结构，并可以沟通现有的系统或预定的逻辑结构；可以解决现有代码和预期的架构探索之间的冲突；当重构、更新或者发展系统

时，可以进行规划架构可视化结构设计，以增强维护代码架构能力，包括探索验证和创建业务逻辑等。

在Layer Diagram中，让架构设计和开发结果之间的验证更加简单。当我们在做架构设计时，一开始最常碰到的就是决定系统的分层状况。例如，三层、MVC和MVP等体系架构中的用户接口、商业逻辑规则和数据存取。从开发的角度来看，用户接口应该只能透过商务逻辑规则层去存取数据层。不过在实际开发时，我们不经意地绕道而行是常见的事。如何让开发出来的东西真的能遵循我们原本设计的架构，不至于产生违背设计（以致后续难以维护）编写代码的情况，Visual Studio 2010新增层验证（Layer Validation）功能可以方便的解决这种问题。如图2所示。

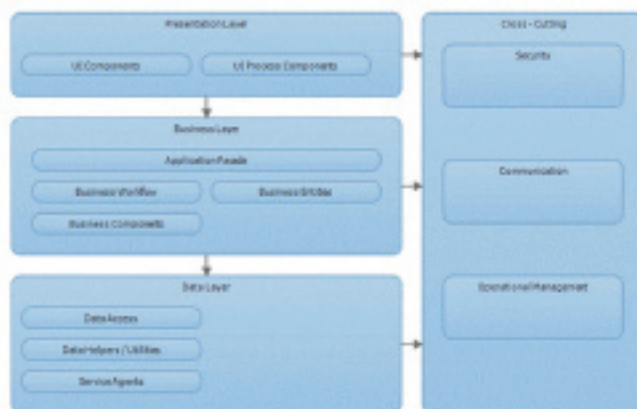


图2 VS2010的层的可视化设计器绘制层图

通过使用Visual Studio 2010层图，可以很方便地通过创建自定义工具项来复用现有的图（UML或Layer），还可以创建一个图层，将代码映射到图上，并以手动的方法使用在图上创建的约束来对代码进行验证和约束。

UML 2.1.2规范定义了13种图，Visual Studio 2010支持其中最常用的5种图（类图、用例图、活动图、组件图和顺序图）。它们的底层的Meta模型是完全符合UML2.1.2规范定义的关系。

## 强大的架构代码分析

Visual Studio 2010 Ultimate加入了一种全新的建模功能，称为依赖图（Dependency Graph），是用来辅助用户更好的理解和分析已有的解决方案（从代码到模型）。

依赖图是采用DGML来表示的。DGML（Directed Graph Markup Language）是微软在Visual Studio 2010中开始引入的一种完全符合XML格式的语言，但并不是Visual Studio所特有的。DGML主要是用来描述循环（cyclical）和非循环（acyclic）的有向图，有向图是由一系列的由边（link/edge）进行互联的节点（node）组成。用户可以根据自己的需求，在DGML的基本语法基础上进行扩展，也

可以开发展示工具来可视化DGML图。依赖图仅是DGML的一个应用，Visual Studio也只是为DGML提供了一个展示的工具，在VS2010中已经新提供了“Directed Graph Document”文件模版。

在Visual Studio 2010中使用依赖图，可以帮助我们快速了解的组织 and 现有代码的关系。这些图表作为显现节点之间关系结构的箭头，可表示为节点和Links之间的关系，也可以使用图形表示的结构节点(Nodes)和链接(Links)的关系，还可以显示图表上的节点和解决方案的关系。DGML的语法非常简洁，最简单的DGML文件只需要有DirectedGraph和Links这两个元素即可。如图3所示。

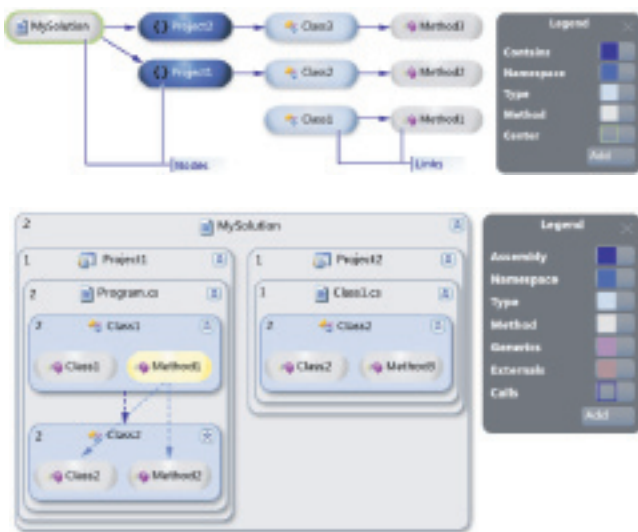


图3 结构节点链接和图表结构包含图

如果想得到解决方案的程序集、命名空间、类更详细的反射信息，可以在一个有向图标记语言文件(DGML)中使用依赖图生成功能自定义分析解决方案。架构工具可以让我们了解应用程序和功能设计，并帮助验证设计与执行过程中不偏离开发的目的。

另外，Visual Studio 2010新提供了两个重要的建模工具：UML模型浏览器(UML Model Explorer)和架构浏览器(Architecture Explorer)，并以图形化的方式描述系统架构，从而使得项目中的技术人员或非技术人员都能以模型通过图形化的方式进行协作。模型浏览器用来帮助用户理解和使用所创建的UML模型。它是一个树状结构的WPF组件，用来展示UML模型中的层次结构。构架浏览器(Architecture Explorer)是用来辅助开发人员分析和理解现有代码和.NET Assembly。

## 模型和图表扩展

Visual Studio 2010的重点工作之一不仅是API进行

改进，而且对整个扩展开发的生态系统进行提升，它的目标很简单就是要使从创建—>发布—>使用的整个过程都更加容易。

安装Visual Studio 2010 SDK可以进行模型、图表和功能扩展。SDK提供了一组项目模板，可帮助我们构建从编辑器视觉效果到菜单命令和工具窗口等各种不同的扩展。WPF是扩展模板中的比较重要的部分。我们可以使用XAML来设计，也可以很轻松的将一个自定义的用户界面添加到VS2010编辑器。以前，尝试做一些简单操作如获取一个符号的位置，也要花费几行难看的互操作代码，使用新的编辑器API你只需一行代码，而无须任何操作。扩展的功能很强大，我们可以将团队的共享网络工具集成到编辑器中，从而便于直接在VS 2010中共享代码。我们也可以自己进行模型扩展，能够将架构图或TFS历史记录也嵌入代码编辑器中，并且用几行代码就可以改变关键字着色的默认样式等等。使用扩展，可以将Visual Studio定制成我们喜欢的强大的编程工具。

安装VS 2010 SDK后，Visual Studio 2010 Ultimate可以进行定义扩展模型图。通过这种方式，可以适应图表和所需要的模型。例如，可以定义菜单命令、UML概要文件，验证限制和工具箱的项目。还可以定义在一个扩展几个部分，也可以分发这些扩展到其他Visual Studio Ultimate用户。在一个Visual Studio集成扩展(VSIX)的形式，还可以创建一个VSIX来使用在Visual Studio VSIX项目。

VS2010 SDK使部署扩展插件变得非常简单，无须再构建MSI来分享扩展。所有的扩展项目模板都会生成VSIX文件。VSIX是新的部署单元，能够利用开放包装公约(Open Packaging Convention) Zip格式，并利用了XCOPY部署功能来部署扩展库。另外，在使用新扩展管理器(Extension Manager)时，用户可以直接在IDE中找到并下载扩展。这种体验得益于新增的及改进的Visual Studio库(Visual Studio Gallery)，只需上传VSIX文件就能与世界共享自己的创作。

## 总结

事实上，Visual Studio 2010在架构建模设计上的支持不仅仅是笔者提到的这些点。Visual Studio 2010架构设计上更强大的是其集成了众多ALM功能，而其核心实际上是后端的Team Foundation Server。Visual Studio 2010的建模设计、开发和测试等都与TFS2010有着紧密的集成。UML图及其图上的元素也同样与TFS的工作项(WorkItem)进行了集成，即由图上的元素可以关联到TFS中的任意类型的工作项。笔者相信，如果您了解了Visual Studio 2010建模方面提供的强大支持之后，在今后工作中一定可以事半功倍。■

# Microsoft在进化？

## ——外包供应商对Visual Studio 2010测试架构的观感

■ 文 / Mody Sheng



### 作者简介：

Mody Sheng (盛曼冰), 20年软件业工作经验, 曾担任开发、设计、测试、过程改进和管理各项职务。参与、主管过多个大型软件项目, 并获得国外项目质量奖项。现服务于博朗公司, 担任测试总监职务。

**印**象中Microsoft的开发平台从诞生之日起就被打上了深刻的“开发专用”印记, 似乎它关心的永远是如何更快速, 更方便地开发和调试。测试, 特别是被严格管理的系统测试向来不是它的关注重点。

然而近日即将发布的新一代开发工具Microsoft Visual Studio 2010却出乎意料地提供了一套包含测试在内的比较完善的软件开发生命周期管理的解决方案。这个华丽转身背后说明了什么? Microsoft在进化? Microsoft不再仅仅爱开发, 同样将爱的光辉洒在测试的身上?

从外包行业的角度分析, 快速的开发的确是客户的一个需求, 但是保证质量更被用户所关注。没有有效受控的测试及测试管理工具, 产品开发只能被认为原始。Visual Studio 2010在这方面堪称巨大的改进确实也可以被称为进化了。从软件测试这方面来讲, Visual Studio2010在新一代的产品中表现抢眼。至此我将结合本公司的一些实际情况及项目经验来谈谈Visual Studio是如何帮助我们团队更好地提高质量以交付给客户满意的产品。

首先先关注下在测试方面VS的新版本和旧版本所涵盖的内容上的差异:

	VS2008	VS2010
Unit Testing	✓	✓
Code Coverage	✓	✓
Test Impact Analysis	✓	✓
Coded UI Test		✓
Web Testing	✓	✓

Load Testing	✓	✓
Test Manager		✓
Test Case Management		✓
Manual Test Execution		✓
Fast Forward for Manual Testing		✓
Lab Management Configuration		✓
Extensible WPF-Based Environment		✓
Virtual environment setup & teardown		✓
Provision environment template		✓
Checkpoint environment		✓

因为软件测试的结果直接影响到产品是否能够准时而有效的交付, 以及项目的进度等等重要因素。由此看来, 微软对于软件测试这个领域也越来越重视。

博朗 (Bleum) 作为一家世界级的软件外包服务公司 (CMMI5) 一直致力于使用微软的优质解决方案来提高团队的运行效率以提高客户满意度。外包项目追求的是高效有序, 控制严格, 在交付高质量产品的同时, 对项目的成本的考虑也异常敏感。随着Visual Studio 2010正式版本即将发布之际, 借此介绍一下我们对于该解决方案的使用情况。我们的客户群体主要集中在欧美国家, 来自于: 医疗、信息技术、慈善、物流、零售业等领域, 因此对于产品的质量有着相当严格的要求。VS2010除了帮助开发人员在开发过程中创造出功能各异的产品, 也同时为软件产品的测试工作带来了福音。随着敏捷开发模式的兴起, 我们的团队也将其精髓付诸于实践, 并且借助VS2010获得了不小的成功, 其中包含了敏捷开发、测试及测试驱动、配置和变更管理、敏



捷项目管理等领域。

对于一些测试项目我们公司的解决方案主要分为商业以及一些开源的工具。在项目实践过程中我们发现商业测试工具虽然可以满足大部分的应用程序而且有着非常完善的测试机制，但其较高的价格令不少客户望而却步，有些尽管是性价比比较高的产品，但诸如对整个测试项目的跟踪报告机制缺乏等问题，对于我们在实际项目中的使用也有不少限制。开源自动化测试前期的投入可能很少，但后期由于缺乏必要的商业支持，在真正部署项目前我们更是会慎之又慎。微软的Visual Studio 2010帮我们跨过了这些障碍，给我们的测试工作带来了便捷。

测试是一个过程也是一种方法，以一个测试项目为例，通常客户会将一些新的需求提交到TFS上，我们的需求人员会对此进行分析，这些宝贵的资料对以后的测试工作都起着重要的作用。但需求不仅仅是表面看起来那么简单，是需要变通的，需要站在客户的角度去考虑如何达到客户的要求。无数软件项目的失败案例中，对于需求的调研不足，或者忽略了客户的需求重点都为以后项目的管理留下隐患。我们以TFS作为项目管理的一个起点，通过各种综合模型或迭代模式，根据不同行业领域、客户特征、软件用途、项目规模等因素考虑。只有符合客户需求的软件才是有价值的，就像每一件艺术作品，都有它特定的受众群。只有明确受众的兴趣所在，才能够打动对方，为更多人所欣赏。当测试人员在拿到需求说明后，测试用例的工作就这样慢慢展开了，VS2010提供了一些撰写测试用例的模板，待进行手动测试时可以随时对测试的结果进行追踪和管理，以评估新的版本能否按时发布，此外VS2010本身还提供了Web Test的组件，根据录制，回放再修改的原则可以方便我们的测试人员进行自动化的测试，而作为进一步的提高产品品质的手段。Web Test能针对.NET平台的Web 应用程序进行有效的自动化功能测试，如果结合VS2010的Loading Test组件配合使用，更可以进行一些比较复杂的性能测试以确保被测系统的抗压性能。自动化测试的结果会以电子表格的结果呈现给测试人员进

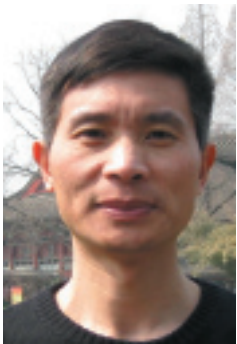
行分析比较，亦可选择将其递交到TFS进行归档处理。对于一些开发项目来说，开发人员在编写完代码之后，可以借助VS2010内建的单元测试框架来检测代码的质量以及一些基于代码规范的工具检测代码中的错误和潜在威胁。

然而在以往的项目经历中遇到的一些问题VS2010依然没有能够完全的覆盖到。比如，从需求到测试用例以及测试结果之间的一系列过程，旧版本的Visual Studio没有很好的将其一起整合起来，中间会出现断档，对于测试项目的整体情况所提供的一些报表信息颗粒度较粗，没有进一步深入。为此我们特意自行开发了一套代号为ACE的套件，用来弥补这些不足及更深层次地完善流程，强调项目的整体性以及互相之间的协作关系，提供各种丰富的报表功能实时监控目前项目发展和进行状况。

软件工程从来都被认为是一个需要绝对理性和超强逻辑的领域，其从头到尾都是一门艺术，像一首完整的交响乐，每个阶段有各自的乐章，但通篇统一完整；每个角色要演好自己的部分，但也要配合好其他乐器的节拍；再加上把握全局的指挥，才能构成一曲和谐优美的音乐。此次新版的Visual Studio 2010在测试方面最大的变化就是完美地和Team Foundation Server结合在了一起，使得整个软件测试项目变得井然有序而又不失高效。从需求，编写用例，测试计划，测试执行，测试结果追踪，在新版的Visual Studio 2010中Microsoft Test Manager作为一个新的组件，通过它的测试计划功能能够帮助定义和管理测试工作和添加任何需要的测试套件，测试案例，或所需要的配置。针对每一项指定的配置执行测试。测试计划还能度量测试进度和报告测试状态。测试结果将自动和测试计划建立起联系，同时测试结果被保存在数据库，便于生成测试趋势和测试历史报告，并能用于比较不同类型的数据。这样一整套流程Visual Studio 2010无缝地将其结合在了一起，使得整个测试项目的乐曲听起来和谐动听，也进一步加强了团队之间的协作，使得目标一致，节奏统一。■

# Visual C++ 2010 新体验

■ 文 / 荣耀



## 作者简介:

荣耀，IT培训师和著译者，南京师范大学中北学院副教授，通过咨询、培训、讲座、著译以及项目合作研究等，帮助企业用户、IT公司、科研院所、大学以及个人理解和应用企业级应用构建语言、工具、思想和方法。长期研究C++编程技术，对于模板编程和面向对象编程具有透彻的理解，对于与C++编程相关的软件开发思想和方法学，包括设计模式、重构、测试等，具有独到的认知。

在1998年推出具有里程碑意义的Visual Studio 6.0之后，好景不长，微软在企业应用开发领域便面临尴尬，其C/S模式解决方案“Visual Basic + Visual C++”遭受“功能全内置”的Borland Delphi的严重冲击，其B/S解决方案“ASP + ActiveX”与Java网页开发技术相比难登大雅之堂。后一点尤其让微软头疼，稍有头脑的人都能看出企业应用互联网化（网页化）已经势不可挡。在这种紧急情况下，微软启动了.NET战略。比尔·盖茨从Borland挖走了Delphi架构师Anders Hejlsberg（即后来的C#语言设计师），在狙击了Java的同时也重创了Delphi，可谓一箭双雕。

.NET成败事关微软生死，微软全力以赴，无暇他顾。鉴于在Windows平台上Visual C++ 的优势地位一时无人能敌，且C++新标准一再难产，因此微软没有动力同时也很难对C++做些什么，结果，从Visual Studio .NET一直到Visual Studio 2008，微软对Visual C++的本地应用（非托管应用）开发能力的升级基本是虚应故事。其间，微软在Visual C++上投入的精力主要是让C++访问.NET设施以及与其他.NET语言交互。在最初推出的Managed C++溃败后，Herb Sutter入主Visual C++团队，担任C++/CLI架构师，目的仍然是为C++访问.NET提供通道，而非对C++本身的改进。随着.NET平台不断发展壮大，逐渐占据企业应用开发半壁江山，微软已经度过了.NET发展的焦躁期，终于能腾出手来重振Visual C++。

2010年2月10日，微软发布了Visual Studio 2010发布候选版，预示着Visual Studio 2010正式版的发布已进入秒读阶段。在发布候选版中，从丰乳肥臀般的启动画面（设计者也许从Naughty America的成人电影Logo中得到了灵感），到体贴入微的源代码编辑器，都表明微软对软件用户体验的理解又上了一个档次。Visual Studio 2010毫无疑问具备

秒杀微软阵营开发人员的魅力。无论是Windows、Office、Web、移动智能设备应用开发，还是并行计算、云计算应用开发，都得到了增强甚至崭新的支持。对于C++的新增支持则是微软营销Visual Studio 2010时有意无意都要提及的一点。

在Visual Studio 2010中，微软从语言、库、编译器以及IDE等多方面增强了C++应用开发能力，篇幅所限，在此仅罗列一些主要的新功能和特性。首先，在语言方面，Visual Studio 2010提前支持如下“C++11”标准（即此前所谓的C++0x标准。据可靠情报，下一代C++标准将于2011年发布……）特性：

- **static\_assert**关键字：编译期断言。若断言失败，则编译失败并产生编译期错误诊断消息。
- **auto**关键字：该关键字被赋予新含义，默认用于自动类型推导，即从变量声明表达式（或初始化表达式）中推导其类型。在模板编程中，auto可以有效地简化编码且避免出错。程序员在Visual Studio 2010中可以使用/Zc:auto编译选项在auto关键字的新旧含义之间进行切换。
- **decltype**关键字：用于返回指定表达式的类型。在与auto关键字结合使用时，可以为模板编程提供极大的方便。程序员可以声明只有编译器才知道的类型，例如，可以声明返回类型依赖于其模板实参类型的函数模板。
- **nullptr**关键字：一个专门的空指针值，表示一个指针变量或对象句柄不指向任何对象。
- **右值引用**：利用右值引用声明符（&&）可以将一个引用声明为右值。右值引用能够解决C++中因不必要的临时对象创建所导致的效率问题，允许程序员使用移动（move）语义和完美转发（perfect forwarding）编写更加高效的代码。
- **Lambda**表达式：类似于匿名函数，兼具函数指针和函数对象之优点同时消除了两者的缺

点。它比函数指针灵活且可以维持状态，又不像函数对象那样需要进行明确的类定义。程序员可以使用lambda函数取代函数对象作为函数模板的实参，还可以结合使用auto关键字将变量类型声明为lambda。

在库方面，Visual Studio 2010除了对STL、ATL、MFC等库进行修补和完善外，还新增了一些库，比较重要的特性如下：

- PPL (Parallel Pattern Library, 并行模式库)：在已经到来的多核CPU时代，PPL为C++程序员编写并行计算代码提供了方便。PPL由并行算法、并行容器和任务并行机制组成。并行算法是一组并行处理数据集合的泛型算法；并行容器是一种泛型容器，提供了并行访问元素的能力；任务并行机制则提供了并行执行多任务的机制。

- CTaskDialog类：MFC的一个新增特性。利用CTaskDialog而不是AfxMessageBox创建的消息框可以向用户显示更丰富的信息。

- 重启管理器 (Restart Manager)：可以为软件提供灾难恢复功能。当应用程序意外关闭或重启时，重启管理器可以用于自动保存文档并重启应用，当应用程序被自动更新时，也可以使用重启管理器启动应用。在创建MFC应用时，程序员可以利用向导也可以手工为应用程序添加该项功能。

- SafeInt库：用于各种类型的整数算术运算，可以自动侦测运算过程中可能发生的整数溢出或删除错误。

对于大多数C++程序员而言，Visual Studio 2010对IDE的增强可能比对语言和库的增强更有吸引力。强大、友好的IDE能够极大地提高程序员的开发效率，缩短软件产品的开发周期。采用了WPF技术开发的Visual C++ 2010 IDE新增了大量的新特性，例如：

- 头文件包含指令自动完成功能：当键入#include时，IDE会创建一个头文件下拉列表以供选择，该列表可以随程序员的键入不断筛选列表项。这样，即使程序员记不清头文件的全名也可以迅速地包含所需的头文件（例如对编程新手而言难于记忆的标准头“algorithm”）。

- 改良的智能感知错误信息：IDE可以更好地侦测可能导致丧失智能感知功能的错误，并在出错代码下面显示红色的波浪线，同时将智能感知错误信息显示在错误列表窗口中。程序员双击错误列表窗口的错误信息行就可以定位到出错代码。

- MFC类向导重装上阵：对于MFC程序员来说

绝对是一个好消息，该向导可以方便地创建类、消息和变量等，而无须手工修改各个代码文件。

以下是笔者个人喜欢的两个小功能，由Visual Studio 2010 IDE提供，并非仅针对Visual C++：

- 文本编辑区缩放：既支持Ctrl+鼠标滚轮无级缩放，也可利用左下角的下拉框缩放。该功能简单、实用。不仅便于演示和交流，也便于开发者迅速定位到指定代码。

- 调用层次窗口：通过该窗口可以方便地查看函数和成员函数的调用关系。

另外，Visual Studio 2010支持多显示器，即可将IDE中的多个窗口浮动显示在不同的显示器上。虽然不知道究竟会有多少程序员使用这个功能，不过显然这个功能很有趣，至少可以让程序员干活时显得更有范儿。

还有一些其他方面的改进。例如，在项目构建方面，Visual C++ 2010解决方案和项目改用MSBuild（而不是VCBuild）进行构建。MSBuild是一种基于XML的构建工具，具有更大的灵活性和更好的扩展性，至此，MSBuild真正成了Visual Studio所有语言类型的项目的通用构建工具。此外，Visual C++ 2010在编译器、连接器以及辅助工具等方面也做了一些完善。

不过，令人遗憾的是，Visual Studio 2010仍未像对C#那样直接提供对C++代码的重构支持，仍然需要辅以Visual Assist X这样的第三方插件。按照道理讲，以微软的资源提供对C++代码的重构功能并非难事，收购Visual Assist X不能说易如反掌，至少不差钱，但微软并没有这么做，或者说没有把事情做绝，虽然有它自己的考虑，但仍然让人感到一丝不爽。

Visual C++ 2010能有今天，Herb Sutter功不可没，其人智商和情商都高人一筹（在程序员这个群体，技术能力强的情商往往比较低，反之亦然），十几年来一直身居ISO C++标准化委员会要职。作为Visual C++小组（C++/CLI）架构师，Herb Sutter不但竭力消除Managed C++一败涂地给Visual C++造成的负面影响，而且确保了Visual C++ 2010不犯方向性错误。不过，从公开的文献看，Visual C++ 2010只是小心翼翼地加入了一小部分必然会进入“C++11”标准的新特性，虽然此举无可厚非，但在新标准正式出台之前，微软其实可以做更多的事。可以想象，随着“C++11”标准的发布，在下一个版本的Visual Studio发布时，微软必然会大肆渲染对C++的新增支持。■

# 免费午餐之后的选择

## ——体验 Visual Studio 2010 中的并行开发

■ 文 / 曹阳

**若**若干年前我在学校里进行图形图像处理方面的实验项目。当时我们面临的一大挑战，是项目的实时性要求。同学们聚集在一起讨论可行的解决办法，其中最简单直接的，则是向导师申请经费，购买更高端的硬件系统。的确，对于若干年前的程序员来说，即使程序的运行速度不够快，那也是暂时的，随着客户使用的计算机更新换代，性能方面的问题就自行解决了，这不啻为免费的午餐。

### 免费的午餐已经结束

CPU的计算能力仍旧在不断提升中，与其说要感谢摩尔定律，不如说要感谢一代代CPU设计师们不懈的努力。那为什么说免费的午餐已经结束了呢？简单来说就是，由于散热和功耗等物理条件的限制，处理器主频的提升遭遇了瓶颈。业界已经不能提供指数级增长的更快的处理器，而只能选择提供指数级增长的更多的处理器。更加详尽的论述可以参考《程序员》2006年11月号的译文《免费午餐已经结束——软件历史性地向并发靠拢》（<http://blog.csdn.net/hstutter/archive/2006/08/29/1136281.aspx>），在此不再赘述。最近笔者在国内某家在线电脑商城查询主流笔记本电脑的CPU配置，配置单核CPU的笔记本只有6台在售，而双核笔记本的数量是245台。这进一步证明了多核已经成为毋庸置疑的主流。

在这样的发展趋势下，如果把今天在单核下实现的应用程序拿到一台64核的机器上运行，你会看到任务管理器显示如下的画面——只有1/64的计算能力得到了利用。

既然传统的应用程序再也无法顺

其自然地在更高端的硬件设备上获得更高的性能回报，我们要问的是，程序员们面临的选择是什么？Herb Sutter在他的原文中非常明确地给出了答案：并行计算。如果在四年前说“并发将是软件开发史上的又一个重大变革”是一个预测，那么今天，并行计算已经成为软件开发的核心趋势之一（援引自Soma的博客中文版<http://blog.joycode.com/soma/archive/2010/03/04/115899.joy>）。

是到了亲自下厨的时候了。

### 自己烹饪并不容易

我们都知道，应用程序的开发并不仅仅是写代码这么简单。从需求分析，程序的设计编写，到调试和性能优化，以及测试、发布和后期维护，这一系列的行为都有其复杂性。而并行计算在程序原有的复杂性上，更平添了一个维度。在这个过程中，程序员会遇到各种各样的问题：

- 我的性能目标是什么？应用程序性能瓶颈在哪里？
  - 这个任务可以并发执行么？哪种形式的并行最适合我，任务并行，数据并行，还是管道（Pipeline）并行？
  - 如果可以利用并行，性能究竟能提升多少？
  - 该怎么测试？如何确保一定的覆盖率？
- 在实际编程过程中，更加细节的问题也会涌现：
- 并发执行的线程之间如何同步？共享的内存应该如何保护？怎样避免死锁？
  - 为什么我并发执行的任务反倒使得性能下降了？或者为什么性能提升不如我预想的那么多？
  - 资源分配是否合理？任务可以在

处理器之间平均地分配么？

● 代码可读性和可维护性如何？我的程序扩展性怎么样？如果在更多核的机算上运行会如何？

● 为什么同样一段程序运行多次的结果不一样？如何重现执行程序中的错误？用什么方法调试我的应用程序？

毋庸置疑，开发并行应用程序要比开发传统的串行应用程序难。那么一些准备好的食材，一套可供参考的菜谱和一系列顺手的厨具，是不是能让自己烹饪的过程稍微容易一点呢？

### 食材，菜谱和厨具很重要

2007年，微软宣布成立并行计算平台组，致力于简化并行应用程序的开发。随着Visual Studio 2010和.NET Framework 4的发布在即，我们看到这一努力的初步结果。

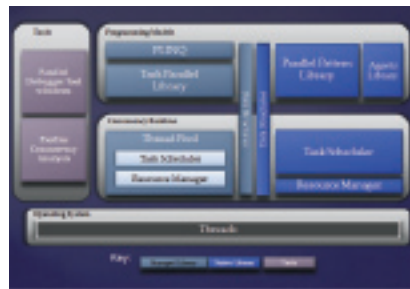


图1 Visual Studio 2010和.NET Framework 4对并行开发的支持

### 食材——并发运行时

#### （Concurrency Runtime）

不论是任务并行、数据并行，还是管道并行，都意味着要把一个耗时的任务或数据分割成更小的单位。分割的颗粒度往往是由任务的性质决定的。细颗粒度并行的好处主要体现在可扩展性和负载均衡上。假设一个耗时的任务只被粗略分割成四个子任务并发执行，那么它对于多于四核的机

器的扩展性就不够好；哪怕是在四核的机器上运行，也无法做到实时动态的负载平衡，可能发生三个子任务早早完成，而另一个任务还在一个核上苦苦等待的悲剧性事件。

为了支持细粒度并行，Visual Studio 2010中为C++实现了一套并发编程框架，支持常用的协同任务调度和硬件资源（CPU和内存）管理。 .NET Framework 4也在已有的线程池（Thread Pool）基础上，实现了协同任务调度和Work-stealing算法。Work-stealing算法充分利用了细颗粒度并行的优势，保证空闲的线程依照一定的顺序，从本地、全局，甚至是其他线程的任务队列中“偷取”任务执行。当然，默认的任务调度器可以被扩展或配置以支持特殊的调度策略。

### 菜谱——编程模型

高抽象层次的、统一的编程模式是简化并行程序开发的一个重要方向。因此你会在Visual Studio 2010和.NET Framework 4中看到许多新的语言和库功能，以及一系列面向任务的并行结构和算法。

在C++用户方面，并行模式库（Parallel Pattern Library）引入了支持任务并行的任务（Tasks）和任务组（Task Groups）概念。Concurrency::task\_handle类包含执行细颗粒度任务的代码，用来代表一个任务。而Concurrency::task\_group类用来组织、调度、等待或者取消某个或某些任务。你可以直接写如下的代码来并发执行若干个任务：

```
// 使用tg对象来并发执行若干任务
task_group tg;

tg.run([] { do_something(1);
});
tg.run([] { do_something(2);
});
tg.run([] { do_something(3);
});

// 等待所有的任务结束执行
tg.Wait();
```

为了降低将串行应用程序改写为

并行应用程序的难度，并行模式库提供了parallel\_for和parallel\_for\_each，你可以很容易地将自己的for和for\_each循环改写为并发执行，而丝毫不会降低代码的可读性；同时对于管道并行，Agents Library提供了基于数据流编程模型的C++模板库，利用进程内消息传递，隔离了对共享资源的访问。

与此相对应，在托管代码方面，任务并行库（Task Parallel Library）引入System.Threading.Tasks.Task类，以及Parallel.For和Parallel.ForEach来支持任务或数据并行。

```
// 串行for循环
for (int i = 0; i < 100; i++) {
    a[i] = a[i]*a[i];
}
// 并行for循环
Parallel.For(0, 100,
delegate(int i) {
    a[i] = a[i]*a[i];
});
```

此外值得一提的是PLINQ（Parallel Language Integrated Query）。PLINQ作为LINQ的并行实现，对内存内的IEnumerable数据源进行分区，随后利用系统内的多核，并行在子数据源上操作。

### 厨具——开发工具

好的开发工具是成功的一半。Visual Studio 2010里包含了两个新的调试器窗口和一个新的性能可视化剖析器（Profiler）。

在MSDN你能找到详尽的关于调试基于任务的并行应用程序的文档（<http://msdn.microsoft.com/en-us/magazine/ee410778.aspx>），在此不再详述。有了这两个调试器窗口，你可以在代码同等的抽象层面上——也就是任务——进行调试，你可以看到任务的状态，彼此之间的关系，调用堆栈等等。当然如果你感兴趣，也可以看到任务对应的线程，并利用一个全局的统一视图来查看所有线程的调用堆栈，以及彼此之间的关系。

性能优化是程序员明知山有虎偏向虎山行的重要原因，那么一个全面的

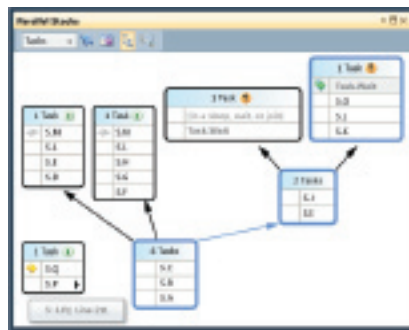


图2 并行堆栈窗口（Parallel Stacks Window）

性能探查器自然是必不可少的。Visual Studio 2010里的并行性能可视化探查器让你看到应用程序对资源的利用情况，程序的哪个部分是受计算量限制的；你也可以看到线程的执行情况，阻断的原因，线程在不同的核之间切换的情况，等等。你也可以在线程执行的不同时间点上，跳转到相应的调用堆栈，去研究造成线程阻断的根本原因。

### 总结

我相信Visual Studio 2010和.NET Framework 4的推出有助于简化并推动并行应用程序的开发，前文提到的几个在并行开发过程中会遇到的问题也会得到部分的解答。

在微软研究院，还有一些正在进行的研究项目，比如Axum（一种针对域的并发编程语言），DryadLINQ（在大型PC集群上进行大规模数据并行操作的编程环境），CHESS（用于查找和重现Bug的并发测试工具）等等。随着研究的深入和更多并行编程模式的推出和实践，我们面对并行计算的挑战已经做好了准备，你呢？■



### 作者简介：

曹阳，现任微软亚太研发集团开发工具中国团队的项目经理，主要负责Visual Studio并行调试器的开发和设计。此前曾参与Visual Studio 2008及Visual Studio 2010中类设计器，数据同步设计器，WPF数据绑定等工具的开发设计工作。

# Windows Phone 开发工具初体验

■ 文 / 马宁

Windows Phone 开发工具在 MIX 2010 上火热登场了。Windows Mobile 开发者们压抑许久的热情终于爆发出来，对于 Windows Phone 的华丽转身，开发者们褒贬不一，有人对 Silverlight、XNA 等新技术充满期待，有人对放弃以前版本兼容性、不支持 C++ 开发满腹怨言。

无论如何，Windows Phone 为我们带来了新的用户体验、新的开发工具，总比让开发者守着 Windows Mobile 的老面孔，一起慢慢变老，要好吧？下面，就让我们看看 Windows Phone 开发工具会为开发者们带来哪些惊喜，呃，或者惊讶。

Windows Phone 开发工具的社区预览版（CTP）可以在 MSDN 上下载。其中包括 Visual Studio 2010 Express for Windows Phone, Windows Phone 模拟器, Silverlight for Windows Phone 和 XNA 4.0 Game Studio。目前，Windows Phone 开发工具只能安装在 Windows 7 和 Vista SP2 上，还需要显卡支持 DirectX 10。

打开 VS 2010 后，我的感觉喜忧参半，喜的是 Windows Phone 提供了两种不同的开发平台：Silverlight for Windows Phone 和 XNA 4.0，忧的是，以前 Windows Mobile 的 C++ 和 .NET Compact Framework 都用不上了。所以，以前的 Windows Mobile 程序员也要经历一个痛苦的转型过程。

接下来，我们就看看 Windows Phone 带来的双重惊喜吧！

## Silverlight for Windows Phone

Silverlight for Windows Phone 险些沦为一个笑话，Windows Mobile 程序员在抱怨，Symbian 上的 Silverlight 都出来了，Windows Mobile 上的 Silverlight 在哪里？这次，Silverlight for Windows Phone 来了，而且所有 Windows Phone 上的应用必须用 Silverlight 来写。

Silverlight 最大的优势在于与 Expression Blend 的无缝整合，可以快速设计出非常漂亮的用户界面，将程序员从烦琐的 UI 开发中解放出来。而且 Silverlight for Windows Phone 不再是 Silverlight 的子集，而是支持 Silverlight 的全部功能。这样的好处是，已有的 Silverlight 程序很容易移植到 Windows Phone 手机上。

在程序菜单的 XNA Game Studio 4.0 下打开 VS2010，创建一个新工程，在 Visual C# 下找到三个模板，选

择“Windows Phone Application”。

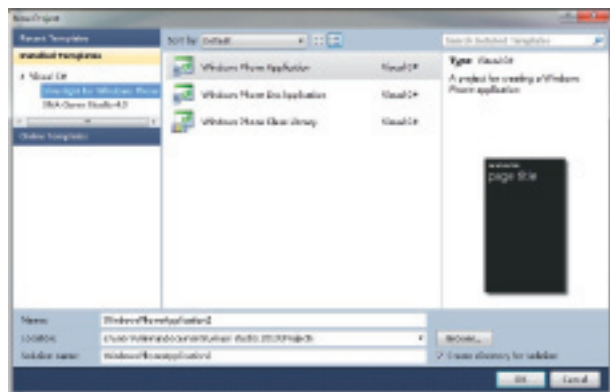


图1 Windows Phone Application 创建界面

不知道 Phone List Application 是啥，看起来 Windows Phone 的界面会以 List 为主了。点击 OK 后，会直接进入 IDE 的主界面：

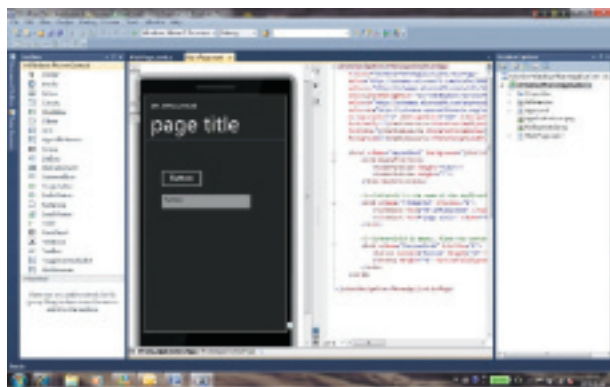


图2 Windows Phone Application IDE 主界面

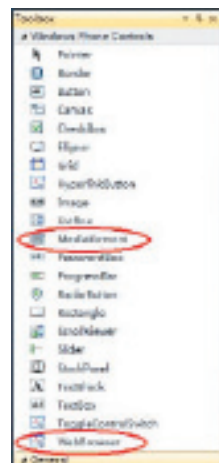


图3 Windows Phone Application IDE 工具栏

似曾相识，左边是工具栏，可以拖拽控件到 UI 设计器上，与 XAML 界面对应。根据示例程序，我准备编写一个浏览器和媒体播放器，首先是从 Toolbox 中将 MediaElement 和 WebBrowser 控件拖拽到 UI 设计器上。

为了播放多媒体文件，我们还要在 Solution Explorer 里将 WMV 文件添加到工程里。做法和 Windows Mobile 工程一致，选择 Content 类型，WMV 文件会和工程一起发布。

接下来就是编写代码了。

MediaElement组件需要设置Source属性，然后调用Play方法。而WebBrowser选择Navigate方法来访问指定网站。

```
private void button1_Click(object sender,
RoutedEventArgs e)
{
    textBlockListTitle.Text = textBox1.Text;
    if (Convert.ToBoolean(checkBox1.IsChecked))
    {
        webBrowser1.Visibility = Visibility.Collapsed;
        mediaElement1.Visibility = Visibility.Visible;
        // play Video
        mediaElement1.Source = new Uri("Wildlife.wmv",
UriKind.Relative);
        mediaElement1.Position = TimeSpan.
FromMilliseconds(0);
        mediaElement1.Play();
    }
    else
    {
        mediaElement1.Visibility = Visibility.Collapsed;
        webBrowser1.Visibility = Visibility.Visible;
        // Browser
        webBrowser1.Navigate(new Uri(textBox1.Text,
UriKind.Absolute));
    }
}
```

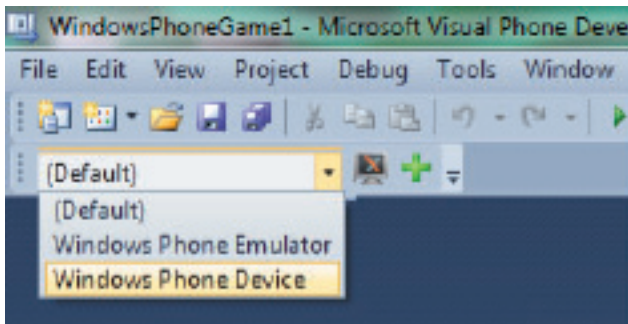


图4 Windows Phone Emulator菜单

代码编辑完成后，选择Windows Phone Emulator作为目标设备。直接按F5，Windows Phone Emulator会被启动。

Silverlight for Windows Phone类库中增加了很多Phone独有的功能。Microsoft.Phone的命名空间下，包括Phone、电话本、电子邮件、Camera、Bing地图、Marketplace、浏览器和搜索等功能。Microsoft.Device.Sensors中包括各种各样的传感器，其中包括重力感应。Silverlight中还包括了Location和Push Notification等独有功能。

我的程序运行起来就是下面这个样子。程序带有强烈的Zune风格，不但有标题，而且还有明显的Block List Title。另外就是输入法，带有明显iPhone的痕迹，对电容屏的支持看来不是问题了。

我还没有仔细看Windows Phone Emulator的实现机制，不过从启动速度来看，绝对不是把整个Windows Phone放在模拟器里执行了。原来Windows Device Emulator那个基于三星2410的小身板估计也跑不起来。

反正也是Managed Code，估计又是JIT编译那套。恍惚间，仿佛回到了Pocket PC 2002的年代。

## Windows Phone 新界面

既然把Windows

Phone Emulator起来了，我们就顺便参观下Windows Phone的新界面吧。点左边的Back键，会将应用程序的Debugger断掉，不知道程序是否退出了。点中间的Win Button，会回到主界面。

不过主界面上只有IE一个图标。Windows Phone模拟器中还没有包括其他的系统功能，比如电话、电子邮件、搜索等。模拟器也没有办法模拟重力感应、Location定位服务。希望到Windows Phone开发工具正式发布时，这些系统功能都可以被支持。

IE的用户体验非常好，一开始新手发蒙，没找到地址栏，后边就非常顺利了。无论是页面缩放时的平滑动画，还是输入时切换到适当大小的设计，都非常顺手。特别是Multi-



图6 Google Map

Tab的设计，感觉非常舒服。特意试了Google Map，显示一切正常。智能手机时代，有一个好的浏览器，其实就成功了一半。

## XNA 4.0

哦，还有一点，我在Windows Phone Emulator里没有做任何配置，就可以上网了。

在移动开发界，XNA还不是一个响亮的名字，但是在游戏开发界，Xbox 360上的XNA则是泰山北斗级的开发技术。XNA不仅仅是一个.NET的游戏开发平台，还包括了很多支持Xbox Live服务的功能，比如，Avatars技术支持在游戏中显示用户设计的形象。希望这项技术能够和同名电影一样，改变移动游戏开发的历史。

Windows Phone支持的是XNA 4.0，与Zune HD上的XNA 3.1有啥区别？支持3D游戏！目前我们看到的这个XNA 4.0预览版只支持Windows Phone开发，不支持Windows和Xbox 360的游戏开发。

Silverlight for Windows Phone是典型的事件驱动型应用程序。而XNA是由时间驱动的应用程序，这也是游戏开发技



图5 Windows Phone 界面图

术的典型特点。每隔固定时间，系统会触发Update事件，使程序得以更新。

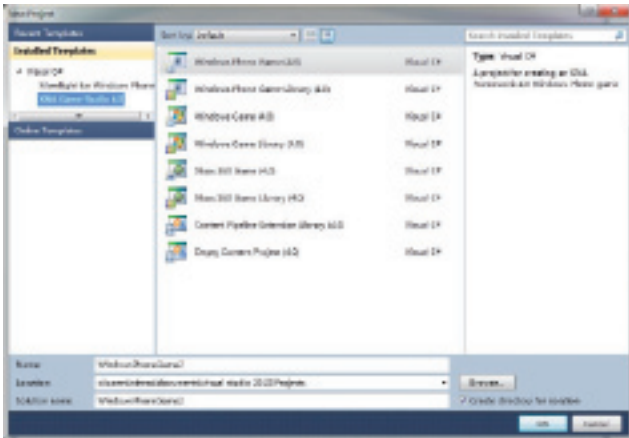


图7 XNA4.0新工程界面

不那么多废话了，创建个XNA 4.0的新工程吧！

选择Windows Phone Game就好了。IDE环境下没有界面编辑器，需要程序员通过代码来实现UI。XNA程序的结构非常简单，程序员只需要实现几个简单的方法，就可以实现一个简单的XNA程序了。这些方法包括Initialize初始化方法、LoadContent加载文件、Update更新内容和Draw绘制等方法。下面是示例代码片段：

```
protected override void LoadContent()
{
    // Create a new SpriteBatch, which can be
    // used to draw textures.
    spriteBatch = new
    SpriteBatch(GraphicsDevice);
    texture1 = Content.
    Load<Texture2D>( "winphone" );
    texture2 = Content.
    Load<Texture2D>( "winphone" );

    soundEffect = Content.
    Load<SoundEffect>( "Windows Ding" );

    spritePosition1.X = 0;
    spritePosition1.Y = 0;

    spritePosition2.X = graphics.
    GraphicsDevice.Viewport.Width - texture1.Width;
    spritePosition2.Y = graphics.
    GraphicsDevice.Viewport.Height - texture1.Height;

    sprite1Height = texture1.Bounds.Height;
    sprite1Width = texture1.Bounds.Width;

    sprite2Height = texture2.Bounds.Height;
    sprite2Width = texture2.Bounds.Width;
}
```

LoadContent方法中使用Content.Load方法来加载资源文件，这些文件也是作为资源被加载到XNA程序中的。

```
protected override void Update(GameTime
gameTime)
{
    // Allows the game to exit
    if (GamePad.GetState(PlayerIndex.One).
    Buttons.Back == ButtonState.Pressed)
```

```
this.Exit();

    // Move the sprite around.
    UpdateSprite(gameTime, ref
spritePosition1, ref spriteSpeed1);
    UpdateSprite(gameTime, ref
spritePosition2, ref spriteSpeed2);
    CheckForCollision();

    base.Update(gameTime);
}
```

Update方法负责更新元素的位置，进行碰撞检测。如果后退键被按下，则退出程序。

```
protected override void Draw(GameTime
gameTime)
{
    GraphicsDevice.Clear(Color.
    CornflowerBlue);

    // TODO: Add your drawing code here
    // Draw the sprite.
    spriteBatch.Begin(SpriteSortMode.
    BackToFront, BlendState.AlphaBlend);
    spriteBatch.Draw(texture1,
    spritePosition1, Color.White);
    spriteBatch.End();

    spriteBatch.Begin(SpriteSortMode.
    BackToFront, BlendState.Opaque);
    spriteBatch.Draw(texture2,
    spritePosition2, Color.AliceBlue);
    spriteBatch.End();

    base.Draw(gameTime);
}
```

Draw方法对更新后的元素进行绘制。

我们的第一个XNA程序会显示两张图片，这两张图片会在屏幕中运行，运行到边缘时，会反弹回来。

## 写在最后

Windows Phone开发工具并不是完美无缺的，比如Visual Studio启动调试程序的时间比较长，希望这是模拟器的问题，在Windows Phone设备上不是这样。

写完这篇文章，心中最多的还是感慨，我们不应该将Windows Phone视作Windows Mobile的升级了，这是一个全新的手机操作系统。Windows Phone的功能可能没有Windows Mobile丰富，但是它赢得喝彩，靠的只有四个字——用户体验。

凤凰涅槃，Windows Phone的路还很长，但是我们这些紧紧追随的开发者们，要的不就是对未来的一丝憧憬吗？■



## 作者简介：

马宁，微软最有价值专家，Windows Mobile 开发者。



# .NET 4和VS2010新技术亮点选析

■ 文 / 金旭亮

**微** 软公司宣布，2010年4月12日，在全球同步发布 .NET 4和Visual Studio 2010。

作为微软平台最重要的产品，.NET 4和Visual Studio 2010所带来的诸多新技术成为了众人关注的焦点，本文将选取其中几个重要的技术创新点，介绍其来龙去脉，点出其设计思路，剖析其技术关键，展示其应用前景，从而帮助读者在这一轮新技术浪潮中占据先机。

## 保证代码质量的强有力工具：Code Contract

防卫型编程风格已为广大程序员所熟悉，最典型的做法是在函数开头书写代码检测传入参数的有效性，发现无效参数时抛出一个异常。

“防卫式编码”简单而有效，因而在实际开发中得到了广泛应用。

但是，任何事物都不是完美无缺的，“防卫式编码”也一样，因为它要求在执行真正的功能代码前进行一些检测工作，因此必然会引入一些额外的开销，如果某些检测工作还比较复杂，对程序运行效率会有负面影响。

为了避免这种情况的出现，人们就想出一个“两全其美”的方法——使用“条件编译”。通过定义条件编译符号控制编译器是否编译特定的代码。

从.NET 4开始，软件工程师们多了一个选择，这就是“代码协定（Code Contract）”。

代码协定（Code Contract）的核心类是“Contract”，以下是一个示例：

```
static double Divide(double divisor, double dividend)
{
    Contract.Requires(divisor >= 0 && dividend > 0,
        "除数与被除数必须大于0,并且除数不能为0");
    return divisor / dividend;
}
```

如果只是重复已有的功能，那么“代码协定”就没有太多存在的意义，事实上，它的最强大之处在于提供了一系列的静态方法实现“协定式编程”。

“协定式编程”要求软件工程师在设计和编写代码时必须考虑：

- 方法调用入口所必须满足的条件，称为“前条件（precondition）”
- 方法调用出口所必须满足的条件，称为“后条件（postcondition）”。
- 方法调用前后，必须始终满足的条件是什么，称为“不变量（invariant）”。

由此可见，“协定式编程”能清晰地传递“原始”开发者的设计意图，使得代码调用者在编写新代码时有所参照，无须另外查找文档。

.NET 4将代码协定相关的类型移入到.NET核心程序集mscorlib.dll中，成为CLR功能集的组成部分，对所有.NET编程语言“一视同仁”地给予了代码协定的支持。

## 使用MEF开发拥抱变化的软件系统

.NET 4引入了一个“Managed Extensibility Framework（MEF）”，开发支持插件的软件系统变得前所未有的简单。

### MEF的基本原理

将一个MEF应用程序可看成是多个“可组合部件（Composable Part）”的集合，通过简单地给部件附加“[Import]”和“[Export]”标记，可以清晰地表明部件之间的“服务消费”与“服务提供”关系（图1）。

“导入（Import）”描述了一个部件需要什么。“导出（Export）”描述了一个部件能提供什么。

从面向对象的角度来看，MEF其实就是将类之间的组合关系转换为“导入（需要外界提供的服务）”和“导出（向外界提供服务）”罢了。

导入与导出之间必须相互匹配，匹配的标准由“协定（contract）”进行描述。协定由“协定名（ContractName）”和“元数据（Metadata）”组成。

协定名通常就是导出所关联的字段或属性类型的完整名称。

元数据其实是一个“名称/值”的集合，在实际开发中，程序员可以给部件附加特定的元数据，然后在部件宿主程序中查询这些元数据，完成特定的功能。

装配部件为一个完备的软件系统的过程称为“组合（Compose）”，由部件“组合容器（Composition Container）”完成。每个需要装配部件的.NET部件宿主程序都至少创建一个部件组合容器对象。

部件组合容器为了完成“为部件宿主程序创建合适的部件对象”的工作，必须知道以下信息：

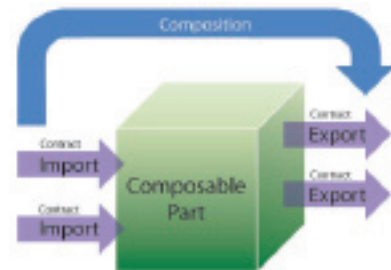


图1 可组合部件示意图

- 有哪些部件可用。
- 这些部件之间的依赖关系。

MEF使用“部件编目(Catalog)”来封装上述信息。

部件组合容器通过查询Catalog,就能获得它所需要的信息,然后,在底层使用反射动态地完成组件识别、装配工作。

## MEF的具体应用

在实际项目中,我们可以将系统中需要动态扩展或升级的部分封装为部件,将其编译为独立的程序集(一个程序集中可以包含多个部件),通过MEF完成部件的实时装配。由此可见,MEF可用于实现系统的可扩充点,多用于应用系统的中间层。

需要特别指出的是, Silverlight 4支持MEF。

可以给一个Silverlight页面附加[Import]标记,指明它可以装配部件,类似地,通过给Silverlight“用户控件(User Control)”附加[Export]标记,每个用户控件就可以成为一个可组合的部件,我们可以将这些用户控件编译为独立的Silverlight程序集,放置在网站的特定文件夹下。

当用户访问Silverlight应用程序时,开始可以只显示一个“初始的简单的”页面,当用户需要时,再动态从Web网站上下下载新的Silverlight程序集(这利用了Silverlight所支持的“按需下载(download on demand)”功能),然后, Silverlight客户端应用程序再使用MEF将程序集中所包含的页面组件“组装”为一个新的功能增强的Silverlight页面。

应用这种技术,可以减少Silverlight客户端与Web服务器间的数据流量,提升Silverlight应用程序的响应性能,同时,也使得Silverlight应用程序的动态扩展变得简单。

## 迈入多核时代

多核化是CPU确定不移的发展方向,双核CPU早已普及,装备有四核CPU的个人电脑正在热卖,所有一切都告诉我们,硬件技术的发展推动着软件开发进入了多核时代。

.NET 4为程序员提供了强大的工具以迎接这一技术挑战。

### 线程取消模型

.NET 4在多线程开发中新推出了一个引人注目的线程“统一取消模型(Unified Model for Cancellation)”。

这个模型的核心是两个类: CancellationToken和CancellationTokenSource。

CancellationToken称为“取消令牌”,它用于代表一个外界的“取消操作”请求。

.NET4所提供的线程统一取消模型可以简述如下。

- 创建CancellationTokenSource对象,它包含了一个取消令牌对象,外界可以通过其Token属性获取“取消令牌”。
- 设计可以被“中断”的线程函数,在此函数中检测取消令牌的IsCancellationRequested属性,从而知道是否应该

中止执行。

- 需要取消线程执行时,调用CancellationTokenSource类的Cancel()方法即可。

.NET 4所引入的线程统一取消模式,规范了多线程程序中“提前结束”线程的方式,在.NET 4新增加的类和组件中得到了较为广泛的应用。

## 并行任务库

“任务并行库(Task Parallel Library : TPL)”是.NET 4为帮助软件工程师开发并行程序而提供的一组类,位于System.Threading和System.Threading.Tasks这两个命名空间中,驻留在3个.NET核心程序集mscorlib.dll、System.dll和System.Core.dll里。使用这些类,可以让软件工程师在开发并行程序时,更关注于问题本身,而不是诸如线程的创建、取消和同步等烦琐的技术细节。

使用TPL开发并行程序,考虑的着眼点是“任务(task)”而非“线程”。

一个任务是一个Task类的实例,它代表某个需要计算机执行的数据处理工作,其特殊之处在于:

在TPL中,任务通常代表一个可以被计算机并行执行的工作。

任务可以由任何一个线程执行,特定的任务与特定的线程之间没有绑定关系。在.NET 4中,TPL使用.NET线程池中的线程来执行任务。

负责将任务“分派”到线程的工作由“任务调度器(Task Scheduler)”负责。任务调度器集成于线程池中。

### 1 Parallel类

在TPL中,最容易使用的类是Parallel,此类提供了三个方法“群”用于实现三种常用的并行程序执行结构。

(1) 使用Parallel.Invoke()方法可以让多条语句并行执行:

```
Parallel.Invoke(
    () => StatementA(),
    () => StatementB(),
    () => StatementC());
```

(2) 使用Parallel.For()方法启动并行循环

```
Parallel.For(0, 100, (i) => DoWork(i));
```

(3) 如果你有一大堆“单个”的数据,并且每个数据都需要进行同样的处理,并且这些处理可以并行,那么使用Parallel.ForEach()方法可以实现这些处理工作的并行执行。

```
Parallel.ForEach(sourceCollection, item =>
    Process(item));
```

### 2 Task类

TPL提供了多种手段实现多任务间的同步。例如,以下代码在task1完成之后自动运行task2:

```
Task task1=new Task(()=>
{
    DoStep1();
});
Task task2 = task1.ContinueWith((PrevTask) =>
{
    DoStep2();
});
task1.Start();
```

使用Task开发,比直接使用Thread要方便许多。

### 3 并行LINQ

PLINQ主要用于并行执行数据查询,而它本身又是.NET 4所引入的并行扩展的有机组成部分,因此,它与LINQ和TPL都有着密切的联系。

在目前的版本中,PLINQ只实现了LINQ to Object的并行执行。

将LINQ查询转换为PLINQ非常简单,在许多情况下只需简单地添加一个AsParallel子句就行了,例如,以下代码把整数集合中的偶数挑出来:

```
//创建一个100个元素的整数集合,保存从1到100的整数.
var source = Enumerable.Range(1, 100);
var evenNums = from num in source.AsParallel()
               where num % 2==0
               select num;
```

与LINQ类似,PLINQ也具有“延迟执行”的特性,只有对查询集合调用foreach迭代、或者调用ToList()之类方法时,PLINQ查询才会真正执行。

另外,PLINQ在底层使用TPL所提供的基础架构完成所有工作,因此,PLINQ是比“Task”抽象层次更高的编程手段。

总之,在设计并程序时,推荐按照以下顺序来设计技术方案:

基于PLINQ的声明式编程方式→使用Task的直接基于TPL的“任务并行”编程方式→使用线程的基于CLR的“多线程”编程方式。

## 使用Silverlight 4开发多层商业应用

Visual Studio 2010的正式版本中集成了Silverlight 4,与以前的版本相比,Silverlight 4带来了众多的新变化,最大的亮点是Silverlight 4已经成为了一个强大的商业应用开发平台。

图2所示为一个典型的Silverlight多层应用程序架构。



图2 Silverlight多层应用程序架构

如图2所示,在Web服务端我们使用ASP.NET应用程序作为WCF RIA Service的宿主,使用ADO.NET实体框架封装数据。

WCF RIA Service在服务端将业务逻辑功能封装为“领域服务(Domain Service)”,Visual Studio 2010在编译项目时,会同时生成此领域服务的客户端代理类

(派生自DomainContext),Silverlight 4提供了一个DomainDataSource组件作为数据绑定源,诸如TextBlock和DataGrid之类客户端数据绑定控件可以直接绑定到DomainDataSource组件。

可以这么说,Silverlight直到4.0版本才算是成为了一个真正的商业应用开发平台,我想这将大大地促进其推广和应用。

## .NET编程语言新成员: F#

函数式编程语言拥有很长的历史,但大多局限于学术界使用,直到Visual Studio 2010中新添加了F#,才使函数式编程开始走向真正的商业应用。

Visual Studio 2010为F#提供了方便和强大的编辑手段,只需在代码编辑器中选中要执行的F#语句,按Alt+Enter组合键,就可以直接在Visual Studio 2010集成的F# Interactive窗口显示结果。

与其他的函数式编程语言Lisp、OCaml等不一样,F#是一个“混血儿”,它同时支持函数式和面向对象两种编程风范,并且可以直接调用.NET基类库中的各种组件,它所开发出来的组件也可以被其他.NET编程语言所调用。

由于F#中的数据结构都是不可改(immutable)的,因此如果使用它来开发多线程程序会非常简单,因为它无须耗用计算资源来进行解决多线程开发中的“竞合条件(race condition)”共享资源的问题。另外,函数式编程的风格,使得F#在实现某些计算机算法时显得简洁而高效,因此,预计F#会在计算密集型的应用系统中有很好的表现。

## 新时代的曙光——云计算

微软首次将对云计算的支持直接融入到Visual Studio 2010中,微软的云计算应用依托于Azure这个云计算平台。

如果希望在云计算应用中访问传统的关系型数据,微软提供了一个SQL Azure服务,云计算应用访问此服务,非常类似于传统.NET应用程序为了方便用户开发云计算应用,微软在Visual Studio 2010中提供了一个本地的Azure开发环境(称为“Windows Azure Simulation Environment”),它使用本地的SQL Server Express作为数据存储。毫无疑问,云计算将是下一个十年中软件技术竞争的主战场,从微软在Visual Studio 2010中首次集成云计算应用开发功能这一现象上,我们似乎已经嗅到了一场技术大战的硝烟:一个新时代正在向我们走来! ■



### 作者简介:

金旭亮,任教于北京理工大学计算机学院,15年开发经验,8年从教生涯,已著三本.NET技术书籍和大学教材。中国高校计算机教育改革的实践者,致力于软件技术的应用与推广,为计算机专业学生和程序员个人提供技术学习指导,为企业提供员工.NET技术系统培训服务。

# 梦想创造可能

## ——盘点微软 .NET 技术八年发展历程

■ 文 / 刘如鸿



### 作者简介:

刘如鸿，擅长大型系统的架构与互联网产品设计，目前为支付宝上海研发中心架构师，专注于系统架构设计。

2000年对于微软是颇有意思的一年，一方面终于迈入了21世纪，担心许久的千年虫问题也没有预想中的那样大面积爆发，通过Windows 95和Windows 98的成功，微软在桌面电脑市场取得了绝对垄断的地位。虽然官司不断，但通过IE捆绑策略也终于彻底打败了傲慢的Netscape，搭上了互联网班车。而Windows 2000的发布也结束了Windows 98和Windows NT两个平台互不兼容、互相掐架的问题，在全新的NT 5.0内核上，服务器和客户端操作系统终于得到了整合。至于办公软件领域，从Office 97之后已经睨视宇内，Lotus Note已经渐渐退出市场，再也没有了重量级的竞争对手。

可在开发工具领域，自从1998年发布Visual Studio 6.0之后，在语言的角度似乎已经止步不前，鼓吹多年的COM技术依旧令人称垢，极高的开发门槛加上本身的复杂性，使自家的VB始终不得其入。而在另一方面，1995年Sun公司推出的Java语言风生水起，因为其倡导“一次编程，到处运行”，加上面向对象、垃圾回收、健壮性等诸多优点，使得这个原本用来开发嵌入式设备和桌面应用的语言反倒服务器开发领域异军突起。

反观微软本身，虽然Visual Studio 6.0号称做到了同一开发平台可以编写多种语言，可实质上只是不同语言、不同开发工具的大杂烩，不同语言之间是彼此孤立的，如果说和Java一样，只是同样做到了可视化开发，或许这就是Visual Studio的本意。

不管是出于进军中高端服务器市场的需要，还是对竞争对手战略防御的考虑，微软都需要一个全新的开发平台来锁定开发人员，使其能够在微软的平台上进行开发和提供服务，这不是令人晦涩的COM或者COM+，也不是Anders初到微软的作品

WFC（可有多少人还记得当年的Visual J++？），也不会是VC++，因为令人绝望的MFC让大多数人望而怯步，当然也不会是VB，它无法胜任相对底层的应用开发，当然，更加没有理由是Java平台，这不符合微软一直以来的风格。

Windows 2000推出之际，微软提出了一个NGWG（下一代的Windows服务）的愿景，除了作为市场宣传的噱头，一个模糊定位的概念并没有做到让更多人明白，对于当时已经全球领先的微软而言，需要一个全新的、充满想象力的平台来延续他们的成功。

2000年，在那年的6月22号，比尔·盖茨向全球宣布其下一代软件和服务，即Microsoft .NET平台的构想和实施步骤。新一代的Microsoft .NET家族产品和技术替代了此前“下一代Windows服务(NGWS)”的提法，它涵盖了帮助软件开发商构建下一代互联网服务和给予新一代智能互联网设备强大功能的软件。此外，微软还宣布了基于.NET平台的新产品计划，其中包括新一代的微软Windows操作系统、Windows DNA服务器、微软Office、MSN互联网网络服务、Visual Studio开发系统。

当时，许多人怀疑这是否是微软一次市场宣传的Buzz Word，

微软是否会在未来的十年乃至二十年坚定地推进他们构筑的.NET战略。2000年微软的白皮书这样定义.NET: Microsoft® .NET 是Microsoft XML Web Services平台。XML Web Services允许应用程序通过Internet进行通信和共享数据，而不管所采用的是哪种操作系统、设备或编程语言。Microsoft .NET 平台提供创建XML Web Services并将这些服务集成在一起之所需。对个人用户的好处是无缝



的、吸引人的体验。

显然，微软把.NET战略真正押宝在了XML Web Services之上，时隔十年之后，我们可能会说当初微软的决策是极具远见和充满想象力的，可在当时，孰是孰非谁人定论。在平台正式推出之前，为了推进.NET，微软开展了一场声势浩大的Dotnetination（.NET化）的运动，几乎所有传统的、创新的、虚构的产品统统打上了“.NET”标签，最典型的莫过于2000年底推出的.NET Enterprise Server，本质上和.NET技术毫无关系，仅仅是因为这场声势浩大的宣传运动中，需要这样的铺垫罢了。

不可否认，微软在当时过分“夸大”了.NET战略，加之2001年推出的Windows XP并没有.NET的影子，于是渐渐有人怀疑微软用心不专，是否在逐步地放弃当初的信誓旦旦，所谓的.NET战略只是水花镜月。

2002年，一段时间的步履阑珊之后，2002年2月13号微软正式发布了Visual Studio .NET 2002，其中包含了.NET Framework 1.0，除了引入一门全新的语言C#之外，同时提供了对于Java的支持。也有人怀疑过微软当时推出J#的意图，现在看来不过是为了吸引更多的Java开发人员转投.NET怀抱罢了，而不是对于Java市场的有心经营，这个在后来的Visual Studio版本也得到了证明。

版本	版本号	发布时间	Visual Studio	Window内置
1.0	1.0.3705.0	2002-02-13	Visual Studio.NET	
1.1	1.1.4322.573	2003-04-24	Visual Studio.NET 2003	Windows Server 2003
2.0	2.0.50727.42	2005-11-07	Visual Studio 2005	
3.0	3.0.4506.30	2006-11-06		Windows Vista, Windows Server 2008
3.5	3.5.21022.8	2007-11-19	Visual Studio 2008	Windows 7, Windows Server 2008 R2
4.0		2010-04-12	Visual Studio 2010	

作为一个全新的开发平台，在语言方面，C#大量借鉴了Java的语法，同时保留了VB方面的诸多便利性。作为平台的重头戏ASP.NET，也传承了微软一直以来的可视化设计风格，允许开发人员以拖放方式开发Web应用，对于许多人来说是极具吸引力的，于是逐步有开发人员转投.NET怀抱。尤其是随着Borland的没落，越来越多的Delphi开发人员投向了.NET或者Java阵营。而在开发工具的支持下，.NET无疑是有巨大优势的。

可是.NET 1.0毕竟是全新的平台，许多类库是不成熟的。尤其是安全方面，如果反射看过当时的.NET Framework源代码，可以看到许多仓促的痕

迹。另外也继承Windows一直以来的毛病，在7x24运行的场景下，还是“不靠谱”了一点儿，虽然相对于COM，已经好上许多，但依旧会出现一些“诡异”的场景。

2003年4月25日，曾被命名为Windows .NET Server的操作系统Windows Server 2003正式发布，同日发布了Visual Studio .NET 2003，并将.NET Framework的版本升级到了1.1.4322。这个四度易名的Windows操作系统内置了.NET Framework 1.1，同时IIS的版本升级到了6.0。相对于当初.NET战略的喧嚣，此时的微软对于.NET已经低调务实许多。除开发工具本身，对于微软而言，Windows Server 2003是一个非常重要的里程碑：一方面证实了Windows操作系统在企业级应用的能力，一方面.NET终于完成了和Windows操作系统的无缝集成，也真正意义上为开发人员提供了一套完整的.NET解决方案。

在这个版本的.NET Framework中，修正了大量1.0的Bug，性能和稳定性也得到了比较大的提高，与此同时引入了手机应用开发，包含了.NET Compact Framework。不管开发桌面应用，还是Web应用，或者是手机设备的应用，Visual Studio .NET提供了统一的开发语言和开发界面，使得开发人员能够在不同应用开发中自由切换。

随着.NET Framework的稳定，微软内部越来越多的产品采用.NET重新开发，或者提供了和.NET的无缝对接，这个也就是微软一直强调的“Connected”。所有的软件和产品都是可以互联的，其中莫过于2003年发布的Exchange 2003、Office 2003以及2004年发布的Biztalk Server 2004，允许开发人员使用.NET开发应用，并且做到了无缝集成。

在2003年拉斯维加斯的微软开发者大会上，微软首次展示了开发代号为“Whidbey”的下一代开发工具。与此同时展示了开发代号为“Yukon”的下一代数据库服务器，也就是后来的SQL Server 2005。令人惊艳的远不止这些，还有开发代号为“Longhorn”的下一代操作系统，在这个全新的操作系统中，包含非常重要的几个组件：

- Avalon：图形渲染引擎，全新的XML定义语言XAML，使其可以用类似HTML的文本语言来定义界面，并提供了华丽的界面效果。

- Indigo：通信子系统，这是微软为了解决异构通信协议而设计的，旨在希望通过它帮助开发人员

以不同的方式提供服务和通信。

● WinFS: 文件子系统, 现在看来, 这是微软一个美丽的谎言, 也是微软对于文件系统伟大的梦想。据说关于WinFS的概念设计早在2000年就被提出来, 并希望应用于新一代的SQL Server。2003年虽然实现了技术原型, 但终究因为无法克服的性能问题, 微软不得不将计划搁浅。

与此同时, 微软承诺提供一个.NET编程接口, 叫着WinFX。可也正是从这次大会开始, 微软不断跳票成了一种习惯。就如前面提到的, 2004年之前微软发布的大多产品, 都是部分甚至全部采用.NET重写, 至少提供了开发方面的无缝支持, 也就意味着开发人员可以使用C#或者VB.NET在微软平台上开发所有的应用。这里又不得不提Longhorn了, 这个“长角”的犀牛似乎有些吹牛不用本钱, 一次又一次地跳票, 发布日始终无期, 有消息说Longhorn考虑过全部用

方法学, Visual Studio为开发人员提供了全套的解决方案, 使开发人员可以在不离开IDE的情况下完成团队协作。而在类库方面, 引入了Web Parts、Membership、Master Pages、Theme等多个能够加快Web应用开发的技术。只是在这, 微软犯了一个不大不小的错误。一直以来微软开发工具的设计原则是“拖放操作”, 而ASP.NET 2.0的许多技术也正是契合这样的设计思想, 只是微软依旧按照桌面开发的设计思路去引导Web开发。世界已经改变, 在浏览器市场, 由于微软多年的不思进取, Firefox被逐渐接受, 同时随着Web 2.0崛起, 越来越多的网站设计师倾向于更加简洁的界面和更加丰富的用户体验, 于是DIV+CSS和Ajax变得越来越流行。不管Web Parts也好, Membership也罢, 虽然可以通过拖放很简单地完成一个业务功能, 但是过于冗余的HTML生成加上自身模型的复杂, 除了一些对性能

和用户体验要求不是很高的站点, 在Web开发上, 开发人员更加愿意坚持手写HTML和使用简单的“控件绑定”。因为他们深信唯有如此, 方可写出符合Web

## 微软用一年的时间给自己所有的产品打上了虚无缥缈的“.NET标签”, 然后用三年的时间完成了艰难蜕变。时至2004年, .NET终于成为微软技术平台的灵魂。

.NET重写, 但这显然是疯狂、不切实际的想法。毕竟Windows的历史包袱太过沉重, 另外在没有硬件支持的情况下, CLR运行效率在针对系统级方面的开发是否有性能问题还是要打个问号。

**2005**年, 一切风调雨顺, 两年前PDC大会给众多开发人员吊足了胃口的Whidbey和Yukon, 终于在2005年11月7号姗姗来迟。一直以.NET命名的Visual Studio换成了Visual Studio 2005, 同时将.NET Framework的版本升级到2.0, 在CLR底层引入了泛型, 同时为了方便开发人员, 内置了一个用于开发调试的Web服务器, 使得开发人员在开发过程可以更加方便地测试与部署。一般来说微软的产品到第三个版本才会稳定下来, .NET Framework也逃不过这个宿命。CLR并不是简单的类库方面的升级, 而是全新的CLR, 相对于1.1版本, 运行效率和稳定性都有了大幅度的提高。

在开发方面, 微软第一次兜售自己的软件开发

2.0风格的Web应用, 而Visual Studio 2005能够帮上的并不多, 加上第一次集成了Visual Studio, 开发工具的性能并没有得到很好的解决。不管如何, 这是一个成功的Visual Studio版本。

同日发布的SQL Server 2005完全架构在.NET之上, 并允许开发人员使用.NET编写存储过程和函数, 并且因此了XML的数据类型, 同时支持.NET编写的用户自定义类型(UDT), 在这个版本的数据库上, .NET从之前的“Connected”变成了“Embed”。同时SQL Server 2005引入了激动人心的SOAP支持, 这就意味着可以直接将数据库暴露成Web服务, 从而提供给应用系统直接访问, 为了方便开发人员进一步基于SQL Server进行开发和管理, SQL Server 2005引入了全新的管理界面——SQL Server Management Studio, 并且和Visual Studio 2005的界面一致。

**2006**年, Longhorn操作系统的一次又一次跳票, 几乎让所有人都失去

了耐心，为了抚平抱怨不断的开发者社区，也是为了挽回些颜面，微软将WPF（图形渲染）、WCF（通信）、WF（工作流）和Windows CardSpace整合成代号为“WinFX”的.NET Framework 3.0，并于2006年11月6日发布。不过此WinFX并不是三年前提到的WinFX。不管如何，从某种程度来说，微软在三年前的开发者大会的承诺总算兑现了一部分，3.0并不是一个全新的.NET Framework，而是2.0的补充，之前提到过2.0在CLR层面脱胎换骨，但是在基础类库上依旧比较薄弱，随同2.0发布的许多新类库华而不实，并无法对开发者社区产生重大影响。而3.0的发布是对2.0框架的一个重要补充，它弥补了微软在企业级开发的软肋。

同年Biztalk Server 2006发布，如同SQL Server 2005的做法，这是一个完全用.NET编写的企业级平台，也是完全支持.NET的开发平台。

**2007**年1月30日，在一次又一次的跳票之后，微软终于全球发布了全新的操作系统——Windows Vista，或许是因为等待了太久，所以期望很高，失望也就更大。我想更多的是因为不断跳票，包括之前的过度许诺，加上对于硬件要求苛刻和驱动的不兼容，从上市的第一天开始Vista就恶评如潮。这个全新的操作系统理所当然地没有全部用.NET编写，因为这本就是天方夜谭。因为诸多的原因，大多企业并没有选择从XP迁移到Vista，而是持续观望，等待下一个版本的Windows。本希望如Windows XP那般成功的Vista注定是一个过渡的操作系统。

2007年11月19号，时隔.NET 3.0发布一年之后，微软发布了Visual Studio 2008，随同发布了.NET Framework 3.5。虽然在CLR上没有改变，依旧是作为2.0的一个扩展，但是在编程模型上，有了重大突破。如果说.NET之前都是处在追赶和模仿其他平台的阶段，那么3.5将是一个华丽转身，Linq和XLinq为开发人员带来了激动人心的编程体验。开发人员可以混合对象与数据，然后用同样的查询方式进行数据处理，更重要的是允许开发人员在任意环节进行扩展，许多Web 2.0网站或者开源框架都提供了Linq for XXX的Provider，从而帮助开发人员以一致的方式进行数据处理。

同年Exchange 2007和Office 2007发布，在企业级办公领域，微软拥有越来越强的竞争力，贯彻微软一致以来的作风，.NET编程在服务器端和客户端都得到了淋漓尽致的支持。

**2008**年2月，Windows Server 2008发布，随同一起的有IIS 7.0。之前的IIS访问.NET，都通过aspnet\_isapi来转发的，也就是说并不是IIS天然支持ASP.NET，而直到IIS 7.0全新引入的管道模型，.NET终于成为Windows操作系统密不可分的一部分。

同年8月6号发布的SQL Server 2008，配套Windows Server 2008和Visual Studio 2008，立足于.NET技术，构成了完整的Windows应用解决方案。八年时间，从一个模糊不清的概念到如今的无处不在，这家全球最伟大的软件公司一直以来不遗余力地推进它的.NET战略。

## 翹动未来的可能

经过近十年的耕耘，微软创造了一个全新的软件开发平台，其中有浮躁、挫折，更有锐意进取。当年的小伙子Scott早就掌管了微软的开发工具事业部，或许他个人的成长历程犹如.NET的成长，喧嚣之后，云烟散尽，当年的.NET梦想已经照进现实，可是否如微软当初预期的成功，外界很难明了。

不过有一点可以肯定，微软通过.NET战略改变了软件开发格局，同样也改变了微软。这家以封闭保守著称的公司，一直以来对开源世界嗤之以鼻，这个状况从2005年之后逐步改观，毕竟开放的微软远比固执的微软更受欢迎。2006年5月启动的codeplex.com，是一个微软拥抱开放的重要标志。除了社区上的开源代码，微软也逐步地将一些内部的产品开源，著名的莫过于Enterprise Library和ASP.NET MVC，与此同时越来越多的开发经理和产品经理开始通过Blog和世界保持沟通。

临即发布的Visual Studio 2010，也带给了我们更多的期待，能够坚持十年去做一个事情已经很难得，未来的路很长，对于微软如此，对于开源世界亦如此。而对于我们每一个人，我想用《程序员》杂志的“技术改变世界，创新驱动中国”来作最后的总结。

有梦想，就有可能。 ■

