# 针对构架师的.NET 3.0介绍

作者 **Mohammad Akif**译者 **朱永光** 发布于 2007年8月6日 上午1时48分

社区 [.NET](http://www.infoq.com/cn/dotnet), [Architecture](http://www.infoq.com/cn/architecture) 主题 [.NET框架](http://www.infoq.com/cn/NETFramework) 标签 [WCF](http://www.infoq.com/cn/wcf), [CLR](http://www.infoq.com/cn/CLR), [WWF](http://www.infoq.com/cn/wf)

微软.NET的最新版本，.NET Framework 3.0，为开发下一代业务解决方案软件系统开启了新的可能。它的设计目标是：提高生产力，降低基础设施复杂性，提供一个一致的元系统，让企业级服务、工作流解决方案和用户体验软件系统的开发更容易。

相关**厂商**内容

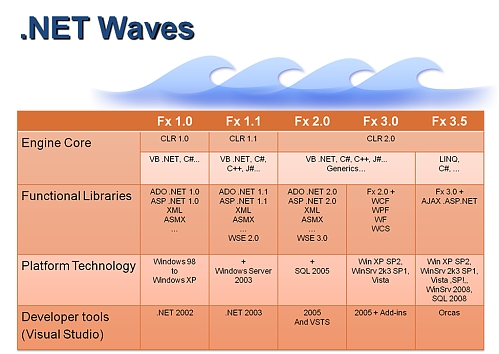
[SOY Framework：Java富客户端快速开发框架](http://www.infoq.com/cn/vendorcontent/show.action?vcr=303)

[QClub：OpenSocial规范、实现现状与展望（8.3 北京）](http://www.infoq.com/cn/vendorcontent/show.action?vcr=359)

在我和大量的构架师讨论的过程中，我听到解决方案构架师非常关心安全、开放的标准、互操作性、面向服务的构架、关键技术间的关系（例如Workflow Foundation和Biztalk）和生产力。在这篇文章当中，我将按照构架师社区最感兴趣的领域来逐一介绍.NET 3.0。

## .NET的相关版本

自从微软推出第一个版本的.NET Framework，已经过去差不多6年了。3.0 是第一个随操作系统发布的框架，它附带在每个Windows Vista中，也可以支持Windows XP SP2 和 Windows Server 2003。直到.NET 3.0，以前的每个版本的.NET Framework都会伴随着一个新的通用语言运行时（Common Language Runtime），后面将简称为CLR。但这次，微软没有修改.NET Framework 3.0 版本中的CLR（**译者注：**而是基于.NET 2.0的CLR），这一点需要重点注意。

由于.NET 2.0和3.0共享同样的CLR，在.NET 2.0中编写的任何程序都可以运行于.NET 3.0中，这是和以前版本的一个重要而显著的区别。以改变来看，对于那些喜欢代数方程的人来说，它们之间的关系可以总结为下面的公式：

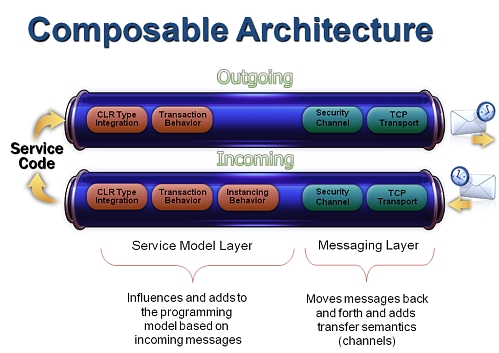
***.NET 3.0 = .NET 2.0 + WCF + WPF + WCS + WF***

我会为每个缩写提供一个定义，不过当你任何时候对.NET 2.0和3.0之间的关系感到困惑时，只要记住上面的这个方程式就好了。.NET 3.0其后的哲学之一就是提供一些功能让“基础设施结构”成为框架的一部分。它让你可以只关注你的关键的业务问题。

.NET Framework 3.0通过4个关键的、以标准为基础的支柱来对应我们客户要求和验证的领域，以达成上述目标。它也包含了一个重要的叫做XAML的新语言。XAML是一个基于XML的声明性语言，通过XML定义对象和它们的属性，允许客户声明性地开发工作流（WF）和身临其境的用户体验（WPF）。让我们来浏览一下.NET 3.0框架中关键支柱的更详细内容。

## Windows Communication Foundation (WCF)

WCF允许你通过提供一个基于标准的框架和一个组合的架构来构建服务。WCF的3个关键设计理念是互操作性、生产力和面向服务的开发。

微软提供了若干个可以轻易添加和删除的消息层通道和服务模型层行为。你也可以定义自己的自定义实例，比如你能编写或者购买一个自定义ASCII编码器，把它作为可重用通道插入到消息层中，让多个系统都能使用。WCF可以和现有的投资互操作，并结合和扩展现存的微软分布式系统技术，如：Enterprise Services、System.Messaging、Microsoft .NET Remoting、ASMX和Web Services Extensions (WSE)。这样的改变预示着，你能使用单一的模型来处理不同类型的应用程序行为，这将显著降低应用程序开发的复杂性。通过支持WS-I 基本概要（WS-I Basic Profile）和大量的额外WS-\*标准，WCF也提供和非微软应用程序的互操作能力。

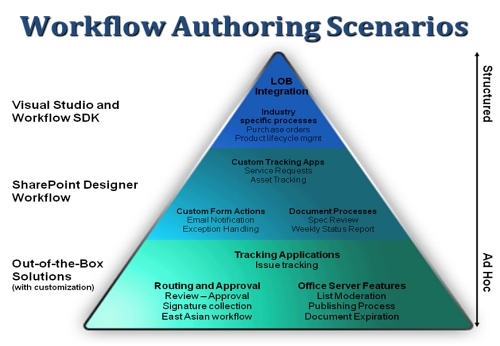
最后，从生产力来看，使用WCF来开发安全的事务性网络服务，你将获得生产力在级别上的显著提高。想想看，要实现类似WCF的功能你需要开发、生成和维护上万行代码，而现在WCF却已经作为基础框架的一部分提供给你了。WCF已经为你提供了一个首要的核心编程框架，以应付逐步增长的面向服务的开发。

## Windows Workflow (WF)

Workflow Foundation是一个企业级工作流开发框架和引擎，它首次把声明性工作流引向主流。WF支持有人参与的（Human）、系统的（System）、连续的（Sequential ）和状态机（State-Machine）工作流。它提供了运行时基础、灵活的工作流控制机制、长时间运行和状态化工作流、对用户而言运行时和设计时的透明性，以及用于规则遵从和记录管理过程的审核能力。

Workflow Foundation允许你把一个工作流定义为一系列的活动。活动即是执行的单元，并被允许轻易地进行重用和组合。基本活动是在一个工作流里面的一些步骤，而组合活动可以包含其他活动。你甚至可以在工作流已经处于运行过程中时添加和删除活动，这将使你在面对改变的时候具有巨大的灵活性。Workflow Foundation提供了一个开箱即用的基础活动库，以及一个让合作伙伴和客户容易创建自定义活动的框架。

在创建方式的选择方面，你可以用纯XAML标记、标记加代码或者纯代码。Visual Studio 2005 Designer for Workflow Foundation作为一个插件程序已经可以下载了，它提供了一个拖拽方式（drag-and-drop ）的设计界面、直观的图形工具、并集成了属性（Properties）窗口、调试和图形注释功能。

许多构架师曾向我询问关于Workflow Foundation、Biztalk、Microsoft Office SharePoint Server 2007(MOSS 2007)和Windows SharePoint Services (WSS)之间关系的问题。

Workflow Foundation(WF)，是由微软开发Biztalk工作流引擎的同一个团队开发的，它倾向于被未来版本的Biztalk Server使用。

WF提供了一个基础，用于实现在一个应用程序和在特定的情况下多个应用程序间的大部分工作流场景。Biztalk允许你自动化你的业务过程，通过适配器来编排由不同技术实现的系统所混合而成的过程，并提供了高级的业务活动监控能力。

对于MOSS 2007和WSS，MOSS 2007构建于WF之上并使用WF作为基础功能提供了额外的功能和特性。Windows SharePoint Services作为Windows Server的添加项提供了一个MOSS 2007的功能子集。简言之，WSS提供了简单的文档管理和工作流能力。

## Windows Presentation Foundation (WPF)

Windows Presentation Foundation试图弥合在游戏和娱乐产业中常见的身临其境的用户体验和在业务软件世界中静态又难用的界面之间的差距。WPF利用XAML让你无需成为图形设计师就能尽情地开发下一代界面。

我建议你去看一个WPF应用程序的演示，来理解我指的下一代用户界面是什么意思。例如，你可以查看一下收藏于[不列颠图书馆](http://www.bl.uk/ttp2/ttp1.html)里面的15大最珍贵图书，其中包括了莫扎特和达芬奇的手稿。这个阅读器是一个基于WPF的应用程序，运行在Internet Explorer浏览器中，就是指所谓的XBAP（可扩展浏览器应用程序）——这是一个用于代替浏览器中的ActiveX功能的技术。WPF的重要不同之处是，它不是一个最终产品，或者一个美妙的丰富界面，而是一种开发和维护应用程序代码的方式。

从构架的角度看，WPF通过分离图形元素和业务逻辑来保持一个非常清晰的划分。一个设计师可以使用Expression产品线和XAML来创建视图，而开发人员可以使用Visual Studio和VB.NET或C#来编写代码。

近来另外一个需要更多关注的技术是WPF Everywhere（WPF/E），它现在的官方名字叫SilverLight。请记住，SilverLight不是.NET 3.0框架的一部分。SilverLight是一个具有自己运行时的跨浏览器、跨平台的插件，它用于开发下一代微软基于.NET的多媒体程序和丰富交互的Web应用程序。你能在<http://www.microsoft.com/silverlight>中找到更多信息并观看一些演示。

## Windows Card Spaces (WCS)

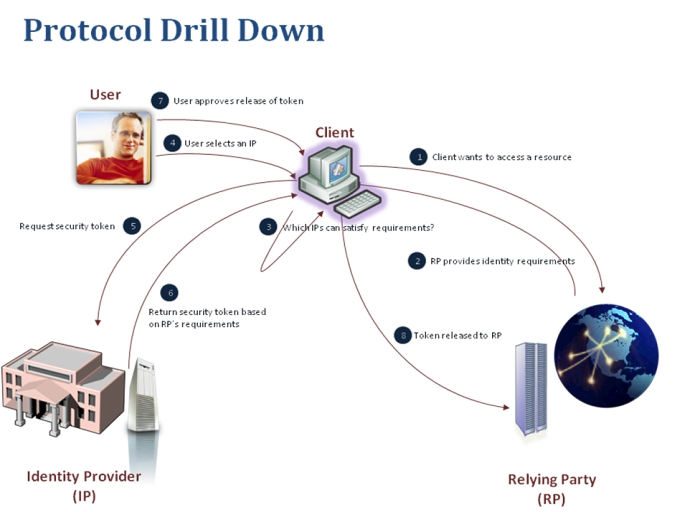
在今天的世界中，每个人都携带着大量的自我声明和第三方颁发的身份标识。身份标识的例子包括驾驶证、信用卡、电影卡和其他类似的卡。

我们把这些由自己控制的信息提供给请求方来证明我们的身份。Windows Card Spaces把用户控制的这个概念扩展到了数字世界。WCS创建了一个身份标识元系统，能显著改善在组织内部和组织之间的企业身份管理的方式。为了理解它的潜力，一位微软著名的评论家指出“这是自加密技术出现以来对计算机安全最重要的贡献之一”。

在数字世界，身份标识被表述为对象（谁）、身份要求和安全令牌（对象和要求的数字表示）。WCS使用自我声明和托管两种概念的身份标识，一个自我声明的数字身份标识卡可以用于登陆类似Hotmail这样的服务，而托管的身份标识可能是一个由银行颁发的信用卡。

下面的图片描述了被用于在不同的实体当中交换信息的协议。请记住在这个例子当中，身份提供者能使用Kerberos、X509或一个自定义的机制。类似的，中转方可以用SAML或者使用HTTPS post来发送安全令牌。

WCS为不同的身份标识管理技术实现提供了一个总体的框架，以让它们共同工作。在Java One（世界上最大的Java会议）上，Sun和微软做了一个联合主题，演示了基于WS-\*标准的互操作机制。我会把这个演示的链接和工具包贴到我在此文结束时提到的博客中。



## 结论

.NET 3.0 Framework为构架师和开发人员开启了一个具有无限可能的新世界。它旨在让你开发、集成和维护应用程序更加容易。微软计划在.NET Framework未来的版本中继续推行这样的理念，来降低基础设施结构复杂性，并同时提高互操作性和标准支持。关于支持标准的完整列表和本文中描述的各个主题的详细资源，可以访问<http://blogs.msdn.com/mohammadakif>和点击.NET 3.0分类来获取。