

## <<Windows Workflow Fou>>

### 图书基本信息

书名：<<Windows Workflow Foundation高级编程>>

13位ISBN编号：9787302184430

10位ISBN编号：7302184437

出版时间：2008-9

出版时间：清华大学出版社

作者：基塔

页数：373

译者：陈宇寒

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 前言

WindowsWorkflowFoundation是最近刚刚发布的.NETFramework3.0中的4个主要基础部分之一。

虽然在微软的技术中曾经出现 workflow 技术，但是在那时它一般作为一次性的解决方案来实现，如专门针对像 BizTalk 或者 Microsoft Content Management Server 等产品的解决方案。

随着 WindowsWorkflowFoundation 的发布，微软为开发基于流程或基于工作流的软件解决方案提供了一种通用的、可扩展的应用编程接口（API）。

您可以将 WindowsWorkflowFoundation 和 .NETFramework 视为等同：把它作为通用的 API，从零开始构建软件应用程序。

开发 WindowsWorkflowFoundation 的目的在于弥补微软产品的不足。

虽然基于工作流的应用程序已经经历了多年的开发，但是绝大部分的工作流基础结构是由定制代码实现的。

当然，这样做需要编写大量的代码并进行长期的维护工作。

除此之外，这种纯粹的代码不能解决任何具体的业务问题。

因此，像 WindowsWorkflowFoundation 这样的架构就提供了一种通用的基础结构，它使开发人员无需考虑如何构造底层的工作流逻辑就能构建基于工作流的解决方案。

本书为软件开发人员和软件架构师提供了构建基于 WindowsWorkflowFoundation 的解决方案时所需要了解的一切知识。

书中的例子和示例代码详尽描述了该技术的每个主要方面。

本书旨在将 WindowsWorkflowFoundation 的相关介绍和运用该技术所构建的例子很好地结合起来。

由于 WindowsWorkflowFoundation 是一个全新的产品，有多少人将会选择使用它尚待分晓。

然而，根据微软对此产品的定位以及一些组织机构运用 workflow 技术的情况来看，人们对这一重大新技术的需求将与日俱增。

本书的目标在于实际指导如何开发基于工作流的软件以及如何将这个软件集成到现有的各种技术体系中。

本书读者对象本书适用于那些想更多地了解 WindowsWorkflowFoundation 的人。

对大多数人而言，只要对这种技术感兴趣，无论是底层的软件开发人员还是上层的架构设计人员都能在本书中找到所需要的内容。

当然，如果读者已经熟悉 .NETFramework，则会对理解本书有辅助作用，因为 WindowsWorkflowFoundation 是直接建立在 .NETFramework 技术之上的。

本书前几章介绍了 workflow 和 WindowsWorkflowFoundation 的基础知识，并使用了简单的代码演示如何进行与一般 workflow 应用程序相关的一些基础操作。

之后，每个章节深入讲解了 WindowsWorkflowFoundation 体系结构的各个方面。

本书还包括运用即插即用功能以及扩展基本 API 的方法开发解决方案的相关主题。

本书内容简介第 1 章包含了和 workflow 以及 .NETFramework3.0 相关的高级信息。

第 2 章面向那些需要直接进入编码阶段开始开发的读者。

第 3 章为那些想要大致了解 WindowsWorkflowFoundation 各个部分的读者提供了主要的资源。

第 5 章详细介绍了 WindowsWorkflowFoundation 架构的各个具体方面。

第 13、14 和 15 章分别介绍了同 WindowsWorkflowFoundation 相关的技术，同时还包括了其他技术，例如 Web 服务、ASP.NET 以及 SharePoint 技术。

您可以在对 workflow 架构了解的基础之上自由地浏览本书，但应该在阅读第 4 章之前读完前 3 章的内容。

使用本书的要求虽然不是必需的，但对于想要理解书中代码的读者，只需达到几个先决条件就可以了，其中最主要的是要有 VisualStudio2005。

虽然 VisualStudio 不是开发 C# 或其他 .NET 代码所必需的工具，但它是开发者开发此类代码经常使用的工具。

因此，本书中大多数代码的编写都是在假设读者使用 VisualStudio 的情况下进行的。

## <<Windows Workflow Fou>>

用WindowsWorkflowFoundation开发软件所需的其他软件都是免费的，并且可以从微软网站上获得。

首先需要下载的软件是.NETFramework2.0和3.0。

.NET2.0为.NET3.0提供基础类库和公共语言运行库。

而.NET3.0包括用于WindowsWorkflowFoundation的新代码以及其他基础部分（见第1章中的介绍）。

接下来需要下载的软件是.NETFramework3.0的VisualStudio2005扩展。

这个软件提供了一个可视化的 workflow 设计工具以及通用的 workflow 项目模板，以此来支持在VisualStudio中开发WindowsWorkflowFoundation。

最后一个需要下载的软件是可选的软件，即MicrosoftWindowsSoftwareDevelopmentKit（SDK）forWindowsVistaand.NETFramework3.0RuntimeComponents。

不要被这个SDK的名称迷惑，它同样适用于其他的Windows操作系统，如Windows2003和WindowsXPSP2。

在这个SDK中提供了文档和示例，这些内容有助于您在下一代Windows开发平台即.NETFramework3.0上开始开发工作。

加入论坛后，您可以发布新的信息，可以给其他用户回信息。

任何时候您都可以浏览信息。

如果您需要某个论坛将新的信息用e-mail发送给您，可以在论坛列表中根据该论坛名称单击论坛的“订阅”图标。

若要了解更多如何使用WroxP2P的信息，请务必参看P2PFAQ，这里解答了论坛的软件是如何运作的，以及专门关于P2P和Wrox书籍的许多普遍的问题。

在任何P2P网页上单击FAQ链接即可阅读FAQ。

## <<Windows Workflow Fou>>

### 内容概要

如果您想掌握建立Windows Workflow Foundation解决方案的技巧，那么本书正是您所需要的。它为您提供了一个清晰、实用的指南，指导您开发基于工作流的软件并将它集成到现有的技术体系中。

在本书中，您还将看到很多现实世界中的例子以及示例代码，它们将帮助您快速起步。

书中对Windows Workflow Foundation中的每个主要部分以及与工作流应用程序相关的基本操作进行了深入研究。

详述了如何在Visual Studio中开发工作流，用定制代码扩展框架，以及使用微软的技术如SharePoint 2007和Windows Communication Foundation（简称WCF）来利用框架等内容。

从而帮助读者学会使用Windows Workflow Foundation来为公司创建极具价值的新型业务解决方案。

作者简介

作者：(美国)基塔 (Kitta.T.) 译者：陈宇寒

## &lt;&lt;Windows Workflow Fou&gt;&gt;

## 书籍目录

第1章  workflow和Windows Workflow Foundation导论	1.1 背景简介	1.2  workflow概念	1.2.1  业务流程管理	
1.2.2  workflow原则	1.2.3  workflow的类型	1.2.4  workflow场景和实例	1.2.5  workflow的实现	
1.3  Windows Workflow Foundation	1.3.1  .NET和.NET Framework 3.0简介	1.3.2  为何选择Windows Workflow Foundation	1.3.3  谁应该关注Windows Workflow Foundation ?	
1.4  小结	第2章 Hello , Workflow !			
2.1  Hello World	2.1.1  需要的工具	2.1.2  练习目标	2.1.3  开始练习	
2.1.4  深入学习	2.2  小结	workflow的组成		
3.1  平台概览	3.1.1  这是一个Foundation	3.1.2  适用对象	3.2  Windows Workflow Foundation的组件	
3.2.1  workflow	3.2.2  活动	3.2.3  宿主	3.2.4  运行时服务	
3.2.5  规则	3.2.6  V Studio	3.2.7  Windows Workflow工具	3.3  Windows Workflow Foundation特性及原理	
3.3.1  动态更新	3.3.2  设计器宿主	3.3.3  扩展性	3.3.4  配置	
3.3.5  监视workflow	3.4  命名空间	3.5  Windows Workflow Foundation及其他技术	3.5.1  SQL Server	
3.5.2  BizTalk Server	3.5.3  Office和SharePoint	3.5.4  Window Communication Foundation	3.5.5  ASP.NET	
3.6  小结	第4章 workflow开发及Visual Studio			
4.1  workflow的组成	4.2  workflow开发风格	4.2.1  Markup-only	4.2.2  Code-and-Markup	
4.2.3  Code-Only	4.3  workflow序列化	4.3.1  序列化类	4.3.2  定制序列化	
4.4  workflow编译	4.4.1  wfc.exe	4.4.2  WorkflowCompiler	4.4.3  编译步骤	
4.5  Visual Studio开发环境	4.5.1  解决方案和项目	4.5.2  菜单	4.5.3  workflow工具条	
4.5.4  工具箱	4.5.5  项目项的类型	4.5.6  Properties窗口	4.5.7  设计器	
4.5.8  设计时验证	4.5.9  主题	4.5.10  调试	4.5.11  类型浏览器	
4.6  小结	第5章 workflow宿主、执行及通信			
5.1  workflow宿主	5.1.1  workflow运行时	5.1.2  WorkflowInstance类	5.1.3  workflow执行	
5.1.4  WorkflowEnvironment类	5.2  workflow通信	5.2.1  参数	5.2.2  本地通信服务	
5.3  小结	第6章 活动			
6.1  活动简介	6.2  内置活动	6.2.1  CallExternalMethod活动	6.2.2  Code活动	
6.2.3  CompensatableSequence活动、Compensate活动以及CompensationHandler活动	6.2.4  CompensatableTransaction Scope活动和TransactionScope活动	6.2.5  ConditionedActivityGroup活动	6.2.6  Delay活动	
6.2.7  EventDriven活动	6.2.8  EventHandlingScope活动和EventHandlers活动	6.2.9  FaultHandler活动和FaultHandlers活动	6.2.10  HandleExternalEvent活动	
6.2.11  IfElse活动和IfElseBranch活动	6.2.12  InvokeWebService活动	6.2.13  InvokeWorkflow活动	6.2.14  Listen活动	
6.2.15  Parallel活动	6.2.16  Policy活动	6.2.17  Replicator活动	6.2.18  Sequence活动	
6.2.19  SetState活动	6.2.20  State活动	6.2.21  StateFinalization活动和StateInitialization活动	6.2.22  Suspend活动	
6.2.23  SynchronizationScope活动	6.2.24  Terminate活动	6.2.25  Throw活动	6.2.26  WebServiceFault活动	
6.2.27  WebServiceInput活动	6.2.28  WebServiceOutput活动	6.2.29  While活动	6.3  开发活动	
6.3.1  Activity类	6.3.2  CompositeActivity类	6.3.3  活动属性	6.3.4  活动组件	
6.3.5  示例：开发WriteFile活动	6.4  小结	第7章 运行时服务		
7.1  内置服务	7.1.1  调度服务	7.1.2  批处理服务	7.1.3  持久化服务	
7.1.4  追踪服务	7.1.5  workflow加载器服务	7.1.6  数据交换服务	7.2  开发运行时服务	
7.2.1  开发调度服务	7.2.2  开发批处理服务	7.2.3  开发持久化服务	7.2.4  开发追踪服务	
7.2.5  开发workflow加载器服务	7.2.6  开发其他服务类型	7.3  小结	第8章 开发高级活动	
8.1  活动执行模型	8.2  活动执行上下文	8.3  开发一个迭代活动	8.4  事件活动	
8.4.1  IEventActivity接口	8.4.2  workflow队列	8.4.3  开发一个Event活动	8.5  小结	
第9章 规则				
9.1  业务规则和Windows Workflow Foundation	9.1.1  代码条件	9.1.2  规则	9.1.3  Windows Workflow规则语言	
9.2  相关活动	9.2.1  IfElse 活动、While 活动、Replicator 活动和ConditionedActivityGroup 活动	9.2.2  Policy	9.3  规则的求值	
9.3.1  链接选项	9.3.2  规则重新求值	9.4  创建规则	9.4.1  规则	
9.4.2  以编程方式创建规则	9.4.3  XML中的规则	9.5  使用workflow规则	9.6  动态升级规则	
9.7  小结	第10章 状态机workflow			
10.1  状态机解析	10.2  何时使用状态机	10.3  状态机活动	10.3.1  StateMachineWorkflow Activity类	
10.3.2  State活动	10.3.3  EventDriven 活动	10.3.4  StateInitialization活动	10.3.5  StateFinalization活动	
10.3.6  SetState活动	10.4  递归状态组合	10.5  状态机设计器	10.6  操作和查询状态机workflow	
10.6.1  StateMachineWorkflowInstance类	10.6.2  查询一个状态机实例	10.6.3  省略步骤	10.7  购物车示例	
10.8  小结	第11章 动态更新和设计器重新载入			
11.1  动态更新	11.1.1  动态更新API	11.1.2  在workflow内部动态更新	11.1.3  更新还是不更新？	
11.1.4  规则的动态更新	11.2  设计器重新载入	11.2.1  设计器API	11.2.2  状态机设计器	
11.3  使用				

<<Windows Workflow Fou>>

态更新和设计器重新载入 11.4 小结 第12章 异常、错误以及调试 12.1 异常 12.2 workflow错误 12.2.1 处理错误 12.2.2 取消句柄 12.2.3 抛出错误 12.3 调试workflow 12.3.1 在Visual Studio中调试 12.3.2 运用WorkflowRuntime事件 12.3.3 追踪 12.3.4 性能计数器 12.4 小结 第13章 Windows Workflow Foundation和ASP.NET 13.1 ASP.NET中的应用程序场景 13.2 技术考虑 13.3 管理工作流运行时 13.4 作为页面流引擎的workflow 13.4.1 Model-View-Controller 13.4.2 建立一个页面流应用程序 13.5 小结 第14章 Windows Workflow Foundation和互连系统 14.1 互连系统和面向服务的架构 14.2 workflow和Web服务 14.2.1 Web服务活动 14.2.2 在一个workflow中调用Web服务 14.2.3 将workflow显示为Web服务 14.3 workflow和Windows Communication Foundation 14.3.1 了解WCF 14.3.2 使用WCF将一个workflow作为一个Windows服务 14.4 小结 第15章 Windows Workflow Foundation和微软Office系统 15.1 微软Office系统 15.1.1 通常的疑虑 15.1.2 已经上市的新产品 15.1.3 SharePoint 15.2 workflow和Office系统 15.2.1 将SharePoint作为一个宿主 15.2.2 Office应用程序作为前端 15.2.3 自带的工作流 15.2.4 SharePoint中的workflow特性 15.2.5 workflow联合 15.2.6 运行workflow 15.2.7 workflow表单 15.3 Office中workflow开发 15.3.1 在SharePoint设计器中进行开发 15.3.2 在Visual Studio中进行开发 15.4 小结

## 章节摘录

插图：第1章 workflow和Windows workflow Foundation导论本章概括了开发商业应用程序以前和现在常用的方式，并介绍了 workflow基础知识以及Windows Worknow Foundation平台。

1.1 背景简介起初，计算机的作用是解决校园里的复杂的数学问题。

在人们认识到可以用计算机解决商业问题之前，计算机的计算能力在一段时间内还仅限于学术范围。商业应用程序的时代就此开始。

作为本书的读者，您可能已经对一些商业应用程序有所了解。

您也许使用.NET代码、C++、Java、SQL或者其他语言编写程序来帮助公司实现商业战略或减少成本。

若是这样，这就意味着您在现代商业的成功中扮演了重要的角色。

传统上，一个业务决定一项含有信息技术的工程值得一做，因为它将给组织带来竞争优势，减少营业成本，或者将一个复杂的手工流程自动化。

一般情况下，项目的软件开发人员先从业务中收集需求，然后进行系统设计和软件设计，最后创建源代码。

当然，任何有价值的软件开发过程都比这种情况更加复杂，但这是一个大体思路。

软件开发过程在过去的50多年里发展迅速，公司已经使用计算机来辅助解决业务流程问题。

在不久以前，软件解决方案的实现方式事实上仍然是面向过程的代码。

在过去10到15年的时间里，面向对象的代码为开发人员建立与现实世界对象相对应的可重用代码块提供了很好的开发方法。

如果使用正确的话，这种方法可以帮助开发人员更加有效而快速地实现软件解决方案。

订单的处理流程、新员工的处理流程、保险的索赔流程只是可以进行自动化业务处理的几个例子。

对这些业务流程进行建模和归档，在适当的时候开发人员通过编写代码来解释对业务过程的描述。

在真正开始编码阶段之前，业务分析员通常要先捕捉过程的每个步骤，并用图形或者任务列表的形式表示业务过程中必须完成的任务以及完成这些任务的顺序。

在这一阶段，完整无缺的任务列表或者可视的图形表示被转换成源代码，这样就可以由机器执行该过程。



#### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>