第一图书网, tushu007.com

<<Windows 7高级编程>>

图书基本信息

书名：<<Windows 7高级编程>>

13位ISBN编号：9787302295952

10位ISBN编号：7302295956

出版时间：2012-9

出版时间：清华大学出版社

作者：米勒

页数：469

译者：贺新征

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

第一图书网, tushu007.com

<<Windows 7高级编程>>

内容概要

《Windows 7高级编程》深入研究了微软自Vista以来修改的各种元素，以及微软根据用户需求添加的新功能。
作者John Paul Mueller具有丰富的实践经验，他在书中深入讲解了如何使用安全策略，同时还从不同的角度讨论了Windows Firewall的管理。
此外，他还以一种简单直接的、过程化的方法讲解了如何使用触发器启动服务，深入讨论了如何利用应用程序的用户账户控制（UAC）功能，细致地分析了通过组策略对象来使用Windows Firewall的方法。

第一图书网, tushu007.com

作者简介

作者：（美国）米勒（John Paul Mueller） 译者：贺新征 米勒（John Paul Mueller），是一位自由职业作家和科技文章编辑。
他的写作天赋是与生俱来的，时至今日，他已经出版了87本专著，并发表了300多篇文章。
从网络技术到人工智能、从数据库管理到自顶向下的程序设计，涉猎范围极广。
最近出版的一些书涉及了Windows命令行参考、VBA和Visio 2007的设计、C#的设计和开发手册以及IronPython编程指导。
他运用自己娴熟的科技文章撰写经验至少帮助过52位作者修订他们的手稿。
John还为DataBased Advisor和Coast Compute杂志提供科技文章编辑服务。
他还积极为其他杂志撰写文章，如DevSource、InformIT、SQL Server Professional、Vsual C++ Developer、Hard Core VisualBasic、asp.netPRO、Software Test & Performance 和Visual Basic Developer。
当John不使用计算机时，他一定是在自己的工作室中。
他是一位狂热的木制工艺和蜡质工艺爱好者。
每个午后，他或是坐在车床旁，或是对他的书籍进行加工。
他同时还喜欢制作甘油肥皂和蜡烛，这些物件迟早会在礼篮中派上用场。

第一图书网, tushu007.com

<<Windows 7高级编程>>

书籍目录

第1部分 Windows 7简介 第1章 WindOWS 7的改进之处 1.1 WindOWS 7是否真的只是增强版的Vista 1.2 从开发人员的角度分析主要的改进 1.2.1 了解用户界面的改进之处 1.2.2 考虑安全的改进之处 1.2.3 开发可扩展的应用程序 1.3 考虑Wlldows xP模式 1.4 开发Windows PowerlsheU 2 1.5 将应用程序移植到windows 7 第2章 开发Windows 7的策略 2.1 确定用户WindOWS 7的舒适级别 2.1.1 明确使用需求 2.1.2 考虑培训需求 2.1.3 取悦于用户 2.2 考虑是否移植 2.2.1 测试应用程序中的问题 2.2.2 明确向Windows 7移植的优势 2.2.3 采用Windows XP模式 2.3 将应用程序移植到willdtows 7 第3章 了解NET 4.0 3.1 了解为什么需要.NET 4.0 3.1.1 定义应用程序的兼容性和部署 3.1.2 研究新核心功能及其改良之处 3.1.3 使用Managed Extensibility Framework 3.1.4 实现并行计算 3.1.5 考虑联网问题 3.1.6 了解数据的改进之处 3.2 获取和安装.NET 4.0 3.3 考虑Windows 7的扩展功能 3.4 将应用程序移植到Willdows 7 第 II 部分 使用Windows7用户界面 第4章 使用任务栏 4.1 简单了解Windows 7的新任务栏 4.1.1 深入了解各项新功能 4.1.2 了解应用程序设置的重要性 4.2 考虑任务栏的升级 4.2.1 任务栏是应用程序的一种交互工具 4.2.2 任务栏新功能简介 4.3 创建基本的任务栏应用程序 4.3.1 获取Microsoft.NET Framework的Windows API.Code Pack 4.3.2 创建解决方案 4.3.3 添加跳转列表代码 4.3.4 测试Code Pack结果 4.4 使用.NET 4.0的方法 4.4.1 建立解决方案 4.4.2 添加代码 4.5 避免滥用任务栏 4.6 将应用程序移植到Windows 7 第5章 高级任务栏技术 5.1 使用跳转列表 5.1.1 添加通用类别 5.1.2 添加自定义类别 5.1.3 执行自定义任务 5.1.4 使用跳转列表导航到网页 5.2 使用进度条 5.2.1 配置进度条应用程序 5.2.2 管理进度条 5.2.3 更改状态 5.3 使用缩略图工具栏 5.3.1 定义缩略图工具栏应用程序 5.3.2 绘制按钮图标 5.3.3 创建工具栏和事件处理程序 5.4 使用覆盖图标 5.5 组合使用任务栏的各种控件 5.6 创建完备的接口 5.6.1 应用程序用跳转列表和缩略图工具栏交互 5.6.2 使用覆盖图标和进度条显示状态 5.7 将应用程序移植到Windows 7 第6章 使用Ribbon界面 6.1 把Ribbori作为一个整体考虑 6.1.1 Ribbon到底是什么 6.1.2 了解Ribbon如何帮助初学用户 6.1.3 在Windows中查看Ribbon 6.2 查看Office中的Ribbon 6.2.1 了解Office的文档链接 6.2.2 考虑如何将Office的技术 扩展到Windows 7 6.3 定义Windows 7中的Ribbon功能 6.3.1 介绍Windows 7中的Ribbon控件 6.3.2 了解控件属性 6.4 将应用程序移植到Windows 7 第7章 创建自定义Ribbon界面应用程序 7.1 开始学习Ribbon 7.1.1 获取Windows 7 SDK 7.1.2 获取RibbonLib 7.1.3 配置应用程序 7.1.4 定义一个Ribbon界面 7.1.5 创建应用程序 7.2 WPF下使用Ribbon 7.2.1 为WPF获取微软的Ribbon 7.2.2 配置WPF应用程序 7.2.3 定义WPF应用程序的Ribbon界面 7.2.4 创建WPF应用程序 7.3 将应用程序移植到Windows 7 第8章 Aero Glass编程 8.1 设计Aero Glass需考虑的问题 8.2 使用Windows 7通用文件对话框 8.2.1 Common File Dialog控件 8.2.2 配置Common File Dialogs示例程序 8.2.3 定义File Open对话框 8.2.4 定义File Save对话框 8.3 使用Windows 7的Task对话框 8.3.1 有效地使用Task对话框 8.3.2 配置Task对话框示例程序 8.3.3 定义Task对话框 8.3.4 编写自动选择代码 8.4 提供扩展语言服务 8.4.1 了解扩展语言服务的角色 8.4.2 配置Extended Linguistic Services示例程序 8.4.3 向应用程序添加扩展语言服务 8.5 将应用程序移植到Windows 7 第9章 使用多点触摸用户界面 9.1 多数公司对多点触摸的需求分析 9.2 明确多点触摸的用户需求 9.3 向应用程序添加多点触摸功能 9.3.1 获取多点触摸平台互操作库 9.3.2 配置应用程序 9.3.3 添加多点触摸界面的功能 9.4 将应用程序移植到Windows 7 第III部分 开发安全的应用程序 第10章 使用Windows 7的标准NT安全功能 10.1 了解基础NT安全功能的变化 10.2 了解基础NT安全 10.2.1 使用ACL 10.2.2 了解安全描述符 10.2.3 了解ACE 10.3 直接使用Windows NT安全 10.3.1 检查用户权限 10.3.2 修改用户权限 10.3.3 审核用户动作 10.3.4 检测文件和目录的权限 10.3.5 修改文件和目录的权限 10.3.6 审核文件和目录 10.4 将应用程序移植到Windows 7 第11章 了解用户账户控制 11.1 了解UAC 11.1.1 分析UAC的需求 11.1.2 覆盖UAC不是什么好主意 11.1.3 用较少权限开发应用程序 11.2 使用UAC 11.3 为应用程序添加UAC支持 11.3.1 创建清单文件 11.3.2 作为一个独立进程执行 11.4 将应用程序移植到Windows 7 第12章 开发增强安全性的应用程序 12.1 现代应用程序的安全需求 12.1.1 使用传统的Windows NT安全 12.1.2 使用新的NT安全功能 12.1.3 使用区域 12.1.4 添加安全角色 12.1.5 添加权限 12.1.6 使用安全策略 12.2 定义应用程序的安全需求 12.3 创建增强安全的应用程序 12.3.1 开发区域 12.3.2 开发安全角色 12.3.3 开发权限 12.4 设计并实现安全策略 12.4.1 配置ClickOnce Intranet示例程序 12.4.2 调试和安装ClickOnce Intranet示例程序 12.4.3 配置ClickOnce Custom示例程序 12.5 避免过多的安全机制 12.6 将应用程序移植到Windows 7 第13章 使用内置安全功能 13.1 使用防火墙 13.1.1 使用

第一图书网, tushu007.com

防火墙 13.1.2 检验防火墙的状态 13.1.3 修改设置 13.1.4 添加和删除端口 13.1.5 添加应用程序 13.1.6 使用GPO技术 13.2 使用自动更新 13.2.1 配置Automatic Update示例程序 13.2.2 编写设置代码 13.2.3 编写更新代码 13.3 访问AppLocker 13.3.1 在注册表中查看AppLocker项 13.3.2 配置AppLocker Demo示例程序 13.3.3 读取AppLocker项 13.3.4 创建AppLocker项 13.4 将应用程序移植到Windows 7 第IV部分 Windows 7高级编程 第14章 后台运行 14.1 后台运行的优势 14.2 开发触发器—启动服务 14.2.1 触发服务 14.2.2 获取ServiceNative.CS文件 14.2.3 配置TriggerStartService示例程序 14.2.4 编写TriggerStartService示例程序代码 14.2.5 测试TriggerStartService 14.3 提供电源管理 14.3.1 配置Power Management示例程序 14.3.2 获取电源管理状态 14.3.3 检测显示器的状态变化 14.4 实现应用程序的重新启动和恢复 14.4.1 配置Application Restart示例 14.4.2 编写Application Restart示例程序代码 14.5 使用网络列表管理器 14.5.1 配置Network List Manager示例程序 14.5.2 编写Network List Manager示例程序代码 14.6 将应用程序移植到Windows 7 第15章 使用Windows 7库 15.1 使用已知文件夹 15.1.1 配置Known Folders示例程序 15.1.2 编写Known Folders示例程序代码 15.2 使用非文件系统容器 15.2.1 配置Non—Filesystem示例程序 15.2.2 编写Non—Filesystem示例程序代码 15.3 考虑用户自定义集合 15.3.1 配置User—Defined Collection示例程序 15.3.2 列举库 15.3.3 添加库 15.4 使用Explorer Browser控件 15.4.1 向Toolbox中添加 Explorer Browser 15.4.2 配置Explorer Browser示例程序 15.4.3 编写Explorer Browser 示例程序代码 15.5 将应用程序移植到 Windows 7 第16章 编写基于Windows 7的64位应用程序 16.1 分析64位应用程序的优点 16.2 了解64位应用程序的相关需求 16.3 解决64位应用程序的开发问题 16.3.1 处理编程问题 16.3.2 访问Windows 7的注册表 16.3.3 承载陈旧的DLL文件 16.4 编写64位应用程序 16.4.1 配置Large—Number示例程序 16.4.2 使用Configuration Manager 16.4.3 编写Large—Number示例程序代码 16.4.4 运行Large—Number测试 16.5 将应用程序移植到 Windows 7 第17章 在Windows 7系统下使用并行编程 17.1 分析并行处理机制的优点 17.2 了解并行处理机制的一些需求 17.2.1 评估任务长度 17.2.2 评估任务类型 17.2.3 有关调试的问题 17.2.4 获取所需求的资源 17.2.5 团队技能 17.3 编写运用并行处理技术的应用程序 17.3.1 了解Parallel类 17.3.2 配置Parallel Process示例程序 17.3.3 编写Parallel Process示例程序代码 17.3.4 调试Parallel Process示例程序代码 17.4 将应用程序移植到Windows 7 第18章 使用传感器和位置平台 18.1 定义传感器和位置设备 18.1.1 传感器种类概述 18.1.2 软件设备 18.2 获取传感器列表 18.2.1 配置Get Sensors示例程序 18.2.2 编写Get Sensors代码 18.3 获取特定传感器信息 18.3.1 了解Geosense for Windows的数据 18.3.2 配置Get Location示例程序 18.3.3 初始化传感器 18.3.4 创建和处理传感器事件 18.3.5 配置Geosense for Windows安全 18.3.6 查看位置传感器的活动 18.4 开发其他传感器和位置设备 18.5 将应用程序移植到Windows 7 第19章 有效使用Windows XP模式 19.1 分析Windows XP模式的问题 19.1.1 为更新进行检测 19.1.2 寻求第三方支持 19.1.3 使用Compatibility Troubleshooter 19.1.4 直接更改应用程序兼容性设置 19.1.5 使用应用程序兼容性工具包 19.1.6 调整用户账户控制 19.2 在Windows XP模式下测试应用程序 19.2.1 获取并安装Windows XP模式 19.2.2 配置Windows XP模式 19.2.3 在虚拟环境中使用应用程序 19.3 Windows XP模式常见问题解决方法 19.3.1 资源权限问题 19.3.2 应用程序拒绝使用资源 19.3.3 运行缓慢的虚拟环境 19.4 将应用程序移植到Windows 7 …… 第V部分 在命令行方式下工作

第一图书网, tushu007.com

章节摘录

版权页： 插图： 有趣的是，数据结构PortData也位于托管内存中，所以代码必须将其封送到本机内存中，以便Win32 API可以访问。
与字符串不同，指针PortDataPtr指向的本机内存结构的大小不容易确定。
因此，代码调用MarshaI.AllocHGlobal（）从全局堆中分配本机内存结构要求的内存，但它必须告诉该方法要分配多少内存。
所以代码调用Marshal.SizeOf（）来确定SERVICE_TRIGGER_SPECIFIC_DATA_ITEM所需本机内存的大小。
但您决不可调用标准的sizeof（）方法来确定数据结构的大小，因为sizeof（）返回的是托管内存的大小。
分配PortData需要的内存仅是第一步。
所分配内存尚未包含任何数据。
Marshal.StructureToPtr（）方法将PortData中的数据转移到本机内存中，同时用PortDataPtr存放指向该内存地址的指针。
如果您不想删除原先的内容，就将第三个参数设置为false。
上述步骤为触发器创建了一个数据元素，并将其封送到本机内存中。
现在需要创建实际的触发器。
第一步是将FIREWALL_PORT_OPEN_GUID所描述的触发器子类型封送到本机内存中。
因为Guid是一个结构，所以采用与处理PortData相同的方法——调用Marshal.AllocHGlobal（）为其分配内存，然后调用Marshal.StructureToPtr（）将数据移动到该内存中。
通过定义SERVICE_TRIGGER对象创建一个触发器。
此例中，代码先创建StartTrigger然后填充数据。
一个触发器由4种元素组成：触发器类型、触发器子类型、动作和数据。
触发器StartTrigger.dwTriggerType是一个简单的枚举
值SERVICE_TRIGGER_TYPE_FIREWALL_PORT_EVENT。
动作StartTrigger.dwAction是一个简单的枚举值SERVICE_TRIGGER_ACTION_SERVICE_START。
触发器子类型StartTrigger.pTriggerSubtype是到FirewallPortOpen所指向的先前已封送的数据的指针。
同样，数据StartTrigger.pDataItems是到PortDataPtr所指向的先前已封送的数据的指针。
该结构还需使用StartTrigger.cDataItems元素告诉Win32 API在PortDataPtr中包含多少个数据项。
终止触发器的创建过程和启动触发器的创建过程一样。
至此，代码有两个托管触发器。
触发器指向本机内存，但数据本身、枚举值和指针驻留在托管内存中。
代码必须创建一个触发器数组，并将其放到ServiceTriggersPtr中。
数组通常是使用托管代码创建的，但为本机代码创建数组则与之不同。

第一图书网, tushu007.com

<<Windows 7高级编程>>

编辑推荐

《Windows 7高级编程》深入研究了微软自Vista以来修改的各种元素，以及微软根据用户需求添加的新功能。
作者John Paul Mueller具有丰富的实践经验，他在书中深入讲解了如何使用安全策略，同时还从不同的角度讨论了Windows Firewall的管理。

第一图书网, tushu007.com

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