



微软公司核心技术书库

Microsoft
Press

Developing International Software
Second Edition



国际化软件开发

(原书第2版)

(美) Dr. International 著
沈凤 等译



机械工业出版社
China Machine Press

微软公司核心技术书库

Microsoft[®]
Press

Developing International Software
Second Edition



国际化软件开发

(原书第2版)

(美) Dr. International 著

沈凤 等译



机械工业出版社
China Machine Press

本书是一本经典之作，主要讲述了开发国际化软件所涉及的根本问题，深入剖析了怎样开发全球可用的应用程序。书中涵盖了最新的技术和内幕，并且是为那些希望为Microsoft Windows 2000和Windows XP平台编写全球可用代码的人员编写的。本书说明了如何以最少的资金和精力，将应用程序本地化、怎样确定文化特定的重要问题、怎样避开和国际化有关的陷阱以及怎样使用最佳的技术和编码方法等。

Dr. International: Developing International Software, Second Edition (ISBN:0-7356-1583-7).

Copyright © 2003 by Microsoft Corporation.

Original English language edition copyright © 2003 by Dr. International.

Published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U.S.A.

All rights reserved.

本书中文简体字版由美国微软出版社授权机械工业出版社出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

版权登记号：图字：01-2003-0286

图书在版编目 (CIP) 数据

国际化软件开发：原书第2版/ (美) 英特耐辛博士著；沈凤等译. -北京：机械工业出版社，2003.8

(微软公司核心技术书库)

书名原文：Developing International Software, Second Edition

ISBN 7-111-12482-0

I. 国… II. ①英… ②沈… III. 软件开发-研究 IV. TP311.52

中国版本图书馆CIP数据核字 (2003) 第050096号

机械工业出版社 (北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑：王高翔 武恩玉

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2003年8月第1版第1次印刷

787mm × 1092mm 1/16 · 23.25印张

印数：0 001- 4 000册

定价：49.00元 (附光盘)

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线电话 (010) 68326294

前言

我在1994年12月完成了本书的第1版。8年之后，我仍然收到很多电子邮件，询问怎样运行或者开发可以在全球使用的软件，通常，我都把这些邮件转发给微软Windows 分部专业的“全球可用指南”（WRG, World-Ready Guides）工作组（他们还负责<http://www.microsoft.com/globaldev>上的Dr. International专栏）。由于到目前为止，问得最多的问题是：“什么时候才会出新书？”所以，当WRG小组承担起修订该书的任务时，我非常激动。本书更全面地阐述了和这个主题相关的内容，这很有必要，并且它还重点关注基于Web的软件，这是自本书的第1版之后出现的一种现象。在1995年，开发人员只要掌握了Microsoft Win32应用程序接口（API）的基本技巧和知识就能开发全球可用的软件，但在当今无处不在的Internet大环境下，开发人员开始寻求了解其他Microsoft产品，如Microsoft Office、Microsoft Internet信息服务（IIS）、Microsoft SQL以及Visual Studio .NET，以及可扩展标记语言（XML, Extensible Markup Language）等工业标准的核心技术。看到在过去的几年中，这些产品和技术已扩展了其国际功能的支持，真是振奋人心。

Windows内的国际支持也在逐步成熟。虽然其根基——Unicode和国家语言支持API（NLS API）——还保持原样，但每个新版本的功能都得到了加强，包含新的区域、区域信息，以及更完善的文本布局和字体功能。本书深入地研究了这些更新的内容，并在得到极大改善的附录中列出了有关这些内容的详细信息。同时，我还很高兴地看到本书用整整一部分来讨论测试问题。考虑到对产品可靠性和安全性的需求不断增长（对跨国公司来说尤其如此），所以没有什么比完整的、有系统的软件测试方法更重要的了。

在过去的8年中，软件市场变得更加全球化，但即使如此，英语仍占据统治地位。我1989年加入微软后不久，当时微软的国际部副总裁举行了一次宴会，庆祝本地化产品的销售收入占了公司整个利润的50%。最近的数目则接近60%。考虑到整个利润的增长，10%的浮动即为数百万美元；但我仍然纳闷为什么该数目没有接近75%或者更多。当然，微软以及其他很多公司都已清楚，全球化软件并非某个部门的责任。例如微软就不再设置国际开发部——每个产品组织都把全球可用当成其开发的核心要求。在雷蒙德市，我们款待来自于原始设备制造商（OEM, Original Equipment Manufacturers）合作伙伴和国际子公司的国际工作组，以保证我们能在一致连续的基础上表述全球可用需求。技术和组织相结合收效甚丰，其中就包括同时发布几个不同语言版本的Microsoft Windows XP。但是我仍然认为，讲英语的开发人员在其内心仍倾向于关注“轻松自如”。这已经超出了全球可用的概念——总的来说，我们的产业需要为用户创造更好的生活环境。公司部署软件开发的情景远比开发人员在设想怎样应用他们的设计和算法时所意识到的情景复杂。

我希望本书不仅能成为不可或缺的参考书，而且能成为每个认真对待客户需求的软件开发组织的必备读物。虽然应付某些挑战（如共享数据以及混合和匹配本地化的应用程序和本地化的操作系统）比以前更容易了，但是，使得全球交互实现起来更简单，使用时功能更强大，却

仍是软件开发团体的责任。启用这类交互的工具仍将得到持续的改进，但即使基础架构已很完美，开发人员也仍需为商业界和消费者恰当地利用这些优势，以从中获得相应的好处。到目前为止，我们这些信奉全球可用性的人已在强调“使软件正常运行”和“减少为此而需付出的劳累”。遵照本书中列出的方法，“让软件正常运行”就是理所当然的了。作为开发人员，你最好把重点放在尚在探索之中的多文化、多语言群件以及它们所展示的尚未被撷取的机会之上。“正常运行”还不够。“现在，那将是改变人们处事方式的伟大想法”是更迷人的提议。

——Nadine Kano
于华盛顿州雷蒙德市

参加本书翻译、录排、校对工作的人员还有：刘波、陈万杰、李新宏、赵志超、施龙飞、帅鹏、刘建成、杨志强、李海涛、赫晓峰、刘海华、李红玲、白红利、金荣学、薛彪、叶哲等。

怎样使用本书

如书名所示，本书是一本关于软件国际化的参考书，虽然它假定程序员、测试人员以及项目经理在其各自的领域中已具有专业水准，但其仍然是为国际化领域的初、中级水平的人员而写的。从平台上来说，主要针对Microsoft Windows XP以及Microsoft Windows 2000，除了有关向后兼容的主题之外，不涉及Microsoft Windows 95、Microsoft Windows 98以及Microsoft Windows千禧版（Me）。本书涉及的编程语言，在开发Microsoft Win32和控制台应用程序时为C和C++；在处理动态HTML（DHTML，Dynamic HTML）时为Microsoft Visual Basic Scripting Edition（VBScript）；在开发Microsoft .NET框架内的应用程序时为Microsoft C#。

本书的组织方式

本书的内容分为七部分：

- 第一部分：简介
- 第二部分：全球化
- 第三部分：本地化能力
- 第四部分：本地化
- 第五部分：测试
- 第六部分：工具和技术
- 第七部分：附录（在光盘中）

第一部分向读者介绍软件国际化的基本原则，以及本书所使用的概念。第二部分在该书中占据的篇幅很大，它讲述至关重要的全球化问题，提供了有关这个问题的基本概念和技术指南。

第三部分研究怎样使得你的软件应用程序和内容可本地化，而第四部分讨论软件和内容本地化的准则，并指出可能妨碍产品本地化的一些问题。第五部分说明怎样进行测试并包含一份方便的测试参考：“国际化测试范例”。第六部分提供了Microsoft给出的创建全球可用应用程序的有用工具和技巧的简单例子，并为你提供了其他参考。最后，第七部分对其他参考资料进行了分类，并给出本书的主要术语（你可以从每部分的简介中找到更多关于本书各部分的目的和内容的信息）。

除了主术语表之外，每章本身还有一个内容较少的术语表。万一你不熟悉该章所出现的术语，那么可以方便地参考这些术语表。主术语表包括各章的术语表以及其他一些术语。

随书光盘

本书有一张随书光盘，它包含本书的电子版、与全球化有关的工具和实用工具、示例应用程序以及其他资源——代码页表、Unicode编码点和Windows平台上可用的代码页之间的映射表、HTM自动布局（HAL，HTM Auto Layout）、Microsoft IME 应用程序接口（Microsoft IME

Application Interface), 以及一个国际开发的商业案例。

内容

随书光盘包含如下项目:

- 带有各种与国际化有关的文档的Extras文件夹:
 - Windows和原始设备制造商(OEM)代码页表。
 - Unicode编码点和Windows平台上可用的代码页之间的映射表。
 - 创建在翻译之后不需重新设置大小的HTML页面、对话框以及用户界面的HTML自动布局规则(HAL)文档。
 - 一个国际开发的商业案例。
 - 在功能和API格式方面简要概述Microsoft IME 2002(日语版API)的输入法编辑器(IME, Input Method Editor)文档。
- Samples文件夹包含各种与国际化有关的应用程序和演示工具的范例和代码。其中包括Microsoft Win32、Web以及Microsoft .NET框架应用程序。
- Tools文件夹包含某些便捷的、国际化特有的工具。
- 英文原书的电子版(eBook)。
- 中文版的附录部分。

使用光盘

把随书光盘放到光驱中, 就可以使用该光盘了。自动装载功能将装载启动菜单。如果启动菜单不能自动装载, 那你可以在光盘的根目录下运行StartCD.exe程序。

关于eBook

随书光盘包含该书的电子版, 可以用5.01或更新版本的Microsoft Internet Explorer查看。要使用该eBook, 请在Autorun菜单中选择eBook选项, 或者从eBook文件夹中运行Autorun.exe应用程序并接着执行屏幕上出现的指令。如果你还没有安装Microsoft Internet Explorer 5.01或者更新的版本, 则eBook安装程序会提供Internet Explorer 5.5供你安装。安装程序已被配置成保留你的当前设置和关联。如果使用的操作系统是Microsoft Windows NT 4.0或者更新版本(包括Microsoft Windows 2000和Microsoft Windows XP), 那你需要具有管理员权限才能安装eBook。

注意 本书的附录A~Y是PDF文件(存放在光盘中, 可用Adobe Acrobat查看), 可通过光盘访问。

书中使用的图标

由于第二部分和第三部分分别有大量全球化和本地化资料, 所以使用了下表中列出的图标来帮助组织这些材料。这些图标将有助于你容易地区分Win32编程、Web开发、.NET框架以及控制台或者文本模式程序设计所特有的信息。

图 标	描 述
	Win32编程特有的材料
	Web开发特有的材料
	.NET框架特有的材料
	控制台或文本模式程序设计特有的材料

Web链接

在写作本书时，书中出现的所有Web站点都可以访问。不过，由于Internet的动态特性，不能保证这些链接现在还保持原样。如果你发现其中某个链接不在了，那么尝试在<http://msdn.microsoft.com>、<http://www.microsoft.com/globaldev>，或者http://www.microsoft.com/globaldev/dis_v2上搜索感兴趣的主体。

问题和反馈

从写作时起，本书中所包括的技术和国际化约定就可能已经在发展之中了。附录中的某些信息，例如地址格式和纸张大小，是让你知道各种默认设置因国家不同而不同。当应用到自己的产品中时，一定要重新检查这些信息。关于本书内容的更新和所有纠错信息将发布在[http:// www.microsoft.com/globaldev/dis_v2](http://www.microsoft.com/globaldev/dis_v2)上。请把问题和反馈发送给Dr. International——drintl@microsoft.com。

目 录

前言

怎样使用本书

第一部分 简介

第1章 理解国际化	2
1.1 区域	3
1.2 全球可用性	4
1.3 本地化	6
1.3.1 本地化过程	8
1.3.2 发布国际化产品	9
1.4 小结	11
第2章 设计全球可用程序	12
2.1 识别规范中的全球可用需求	13
2.2 规划全球化	14
2.3 定制功能特征	14
2.3.1 设计可本地化的国际用户界面	16
2.3.2 研究法律问题	17
2.3.3 保持功能特征的易用性	18
2.4 组建产品组	18
2.5 设置全球可用开发环境	20
2.6 减轻翻译人员的工作量	23
2.7 测试上的考虑事项	24
2.8 小结	25

第二部分 全球化

第3章 Unicode	28
3.1 传统的字符编码	29
3.2 考察Unicode	36
3.2.1 Unicode的性能	36
3.2.2 Unicode的功能	39
3.2.3 Windows对Unicode的支持	42
3.3 创建Win32 Unicode应用程序	42

3.3.1 WCHAR, 一种16位的数据类型	43
3.3.2 Win32 API的W函数原型	44
3.3.3 Unicode文本宏	45
3.3.4 C运行库扩展	45
3.3.5 代码页和Unicode之间的转换函数	46
3.3.6 在Visual C++中编译Unicode应用 程序	46
3.3.7 移植到Unicode中	46
3.3.8 移植到Unicode的选项	48
3.3.9 最优方法	49
3.4 Web页面中的编码	50
3.4.1 Web编码选项	50
3.4.2 设置并处理编码	51
3.4.3 用户覆盖	53
3.4.4 Internet Explorer的语言支持	54
3.5 .NET框架中的编码	54
3.6 控制台或文本模式程序设计中的编码	56
3.6.1 CRT控制台I/O	58
3.6.2 Win32文本模式I/O	59
3.7 小结	59
第4章 区域和文化意识	61
4.1 区域的概念	62
4.1.1 区域变量	62
4.1.2 NLS和区域标识	69
4.1.3 .NET框架中的CultureInfo类	71
4.2 检索正确的区域	72
4.2.1 检索Win32中的用户区域	72
4.2.2 检索浏览器语言设置	73
4.2.3 检索当前的CultureInfo	74
4.3 日期和日历格式	75
4.3.1 操纵Win32中的日历和格式化日期	77
4.3.2 Web页面中的日期格式	80

4.3.3 .NET框架中的日期和日历格式	81
4.4 时间格式	84
4.4.1 Win32中的时间格式和时区	86
4.4.2 Web页面中的时间格式	88
4.4.3 .NET框架中的时间格式和时区	89
4.5 货币格式	90
4.5.1 Win32中的货币格式	91
4.5.2 Web页面中的货币格式	93
4.5.3 .NET框架中的货币格式	93
4.6 大小写转换	96
4.6.1 Win32中的大小写转换	97
4.6.2 Web页面中的大小写转换	97
4.6.3 .NET框架中的大小写转换	98
4.7 排序和字符串比较	98
4.7.1 Win32中的字符串比较和排序	98
4.7.2 .NET框架中的字符串比较和 排序	101
4.8 数字格式	107
4.8.1 Win32中的数字格式	108
4.8.2 Web页面中的数字格式	109
4.8.3 .NET框架中的数字格式	110
4.9 地址	111
4.10 纸张大小	111
4.11 电话号码	112
4.12 度量单位	113
4.12.1 Win32中的度量单位	113
4.12.2 .NET框架中的度量单位	113
4.13 控制台的区域意识	113
4.13.1 C运行库	114
4.13.2 Win32和.NET框架	114
4.14 小结	114
第5章 文本输入、输出以及显示	116
5.1 输入语言	117
5.1.1 处理输入语言的技术	118
5.1.2 输入法编辑器	121
5.2 复杂文字	127
5.2.1 复杂文字中的字符	128

5.2.2 Windows对复杂文字的支持	130
5.3 文本显示选项	131
5.3.1 Win32应用程序的文本布局	131
5.3.2 Web内容和.NET框架中的输入、 输出和显示	137
5.4 字体	137
5.4.1 Windows 2000和Windows XP中的 字体支持	137
5.4.2 Win32应用程序中的字体选择	139
5.4.3 Web内容中的字体控制	145
5.5 小结	147
第6章 多语言用户界面	148
6.1 Windows中的多语言用户界面支持	148
6.1.1 可用性	149
6.1.2 多语言用户界面的好处	149
6.2 提供多语言用户界面解决方案的选项	151
6.2.1 方法1: 语言相关二进制文件	151
6.2.2 方法2: 一个多语言资源文件	152
6.2.3 方法3: 辅助DLL	154
6.3 怎样实现自己的辅助DLL	155
6.3.1 Win32应用程序中的多语言用户 界面	155
6.3.2 多语言Web内容	159
6.3.3 .NET框架中的资源处理	162
6.3.4 .NET框架中的资源装载	165
6.3.5 控制台应用程序中的资源处理	166
6.4 小结	168

第三部分 本地化能力

第7章 软件本地化能力准则	170
7.1 分离可本地化的资源	170
7.2 字符串处理	174
7.2.1 避免运行时组合字符串	174
7.2.2 当必须使用变量时, 使用惟一的 名称	175
7.2.3 不要组合多个变量	178
7.2.4 使句子保持在单个字符串中	178

7.2.5 查看字符串缓冲区大小	179
7.3 考虑UI的本地化能力	180
7.3.1 重新调整元素	180
7.3.2 UI控件的本地化能力	186
7.3.3 图像和图标	188
7.4 小结	192
第8章 镜像	194
8.1 技术	194
8.2 Win32应用程序中的镜像	195
8.2.1 启用资源镜像	196
8.2.2 启用代码镜像	196
8.2.3 镜像和属性页	199
8.2.4 映射坐标	200
8.2.5 处理方向敏感性图形	201
8.2.6 镜像和图像列表控件	205
8.3 Web页中的镜像	206
8.3.1 有向图像	209
8.3.2 文本的强制左对齐	210
8.3.3 绝对CSS定位	210
8.3.4 表提供的可翻转性	211
8.4 .NET框架中的镜像	211
8.4.1 被镜像的Web表单	211
8.4.2 被镜像的Windows窗体	212
8.4.3 消息框	214
8.5 小结	215
第9章 内容本地化能力准则	216
9.1 简化内容	216
9.2 遵循基本的书写风格原则	217
9.3 重视文化和本地敏感性	218
9.4 编写易于循环使用的程序并降低本地化代价	218
9.5 重视艺术作品和多媒体的文化敏感性	219
9.6 设计具有全球性功能的帮助系统	220
9.7 小结	220

第四部分 本地化

第10章 本地化简介	222
------------------	-----

10.1 本地化要素	223
10.1.1 文本	223
10.1.2 布局	224
10.1.3 图形和多媒体	224
10.1.4 键盘快捷键	225
10.1.5 字体	225
10.1.6 区域数据和字符设置	225
10.1.7 构建过程和包	225
10.2 本地化工具	226
10.3 本地化工作组	226
10.4 建立本地化准则	228
10.4.1 保证有效的本地化	228
10.4.2 安排国际发布	228
10.4.3 本地化外购	229
10.5 内容本地化	230
10.6 小结	230

第五部分 测试

第11章 测试全球可用性	234
11.1 全球可用性的质量保证	234
11.2 全球可用的测试方法	235
11.3 测试的全球化	237
11.3.1 准备测试	237
11.3.2 进行测试	240
11.4 本地化能力测试	242
11.5 本地化测试	244
11.5.1 本地化测试的要点	244
11.5.2 本地化测试的平台	244
11.5.3 UI的本地化测试	245
11.6 测试工具的作用	245
11.7 小结	247
11.8 国际化测试样例	247
第12章 用伪本地化测试本地化能力	252
12.1 本地化能力故障目录	252
12.2 伪本地化的常见特征	253
12.3 伪本地化的其他用途	254
12.3.1 图形和音频的伪本地化	255

12.3.2 内容（帮助文件、Web内容） 的伪本地化	255
12.3.3 BiDi语言的伪镜像	255
12.3.4 注释模型	256
12.4 小结	256

第六部分 工具和技术

第13章 图形设备接口+（GDI+）	259
13.1 技术概况	259
13.1.1 GDI+的新特性	259
13.1.2 编程模型	260
13.2 可用性	260
13.3 国际化特征	260
13.3.1 GDI+中的排印处理	260
13.3.2 字体回退	261
13.3.3 图形保真和ClearType	263
13.4 国际化最优方法	263
13.4.1 考虑各种文字间的差异	264
13.4.2 不要一次显示一个字符	264
13.5 其他资源	264
13.6 小结	264
第14章 超文本标记语言（HTML）	265
14.1 技术概况	265
14.2 国际化特征	266
14.2.1 字符实体引用	266
14.2.2 表单提交	267
14.2.3 URL编码	268
14.2.4 层叠样式单	268
14.2.5 LANG属性	269
14.2.6 其他语言属性	270
14.2.7 双向布局	270
14.2.8 BDO元素	270
14.2.9 用CSS设置方向	270
14.3 国际化最优方法	273
14.4 其他资源	273
14.5 小结	273

第15章 Microsoft Internet信息服务（IIS）

和Microsoft 活动服务器

页面（ASP）	275
15.1 技术概况	275
15.2 可用性	275
15.3 国际化特征	275
15.3.1 代码页设置	276
15.3.2 请求集合	278
15.3.3 URI处理	279
15.3.4 其他方法和组件	279
15.3.5 日志	279
15.4 IIS 4和IIS 5的有关问题	280
15.5 国际化最优方法	280
15.6 其他资源	280
15.7 小结	282
第16章 Microsoft Office	283
16.1 工具概况	283
16.2 可用性	284
16.3 国际化特征	284
16.3.1 保证可用40多种语言	284
16.3.2 支持亚洲语言输入	284
16.3.3 支持在各种语言中排序	285
16.3.4 正确的日期和货币格式	285
16.3.5 支持任何编码形式的文本文件和 Web页面I/O	285
16.3.6 为特定市场定制的应用程序专有的 特征	285
16.4 国际化最优方法	286
16.4.1 维护单一代码库	286
16.4.2 考虑你的用户	286
16.4.3 记住使得应用程序可本地化	286
16.5 替代工具	287
16.6 其他资源	287
16.7 小结	287
第17章 MLang	289
17.1 技术概况	289
17.2 可用性	289
17.3 国际化特征	289

17.3.1 代码页和区域枚举	290	20.8 小结	310
17.3.2 字符集转换	291	第21章 Rich Edit	311
17.3.3 代码页检测	292	21.1 技术概况	311
17.3.4 字体链接	292	21.2 可用性	313
17.4 替代技术	293	21.3 国际化特征	314
17.5 其他资源	293	21.3.1 字体绑定	314
17.6 小结	293	21.3.2 Unicode代用对	315
第18章 Unicode的Microsoft层 (MSLU)	295	21.3.3 接口	315
18.1 工具概况	295	21.3.4 对输入法编辑器的支持	316
18.1.1 MSLU的工作方式	297	21.4 问题	317
18.1.2 怎样 (以及何时) 进行覆盖	297	21.5 国际化最优方法	317
18.2 可用性	299	21.5.1 考虑向后兼容非Unicode文本	318
18.3 其他资源	299	21.5.2 考虑字体缩放	318
18.4 小结	300	21.5.3 了解怎样处理Multicode序列	318
第19章 Microsoft .NET框架	301	21.6 替代技术	319
19.1 平台概况	301	21.7 其他资源	319
19.2 可用性	301	21.8 小结	319
19.3 国际化特征	302	第22章 Microsoft SQL Server	320
19.4 其他资源	302	22.1 技术概况	320
19.5 小结	303	22.2 可用性	320
第20章 OpenType字体	304	22.3 国际化特征	320
20.1 技术概况	304	22.4 国际化最优方法	326
20.2 可用性	304	22.4.1 始终使用Unicode数据类型	326
20.3 国际化特征	305	22.4.2 只要有可能就标准化站点校对	327
20.3.1 置换可替换的字形形式	306	22.4.3 如有可能就配置Unicode服务器 和客户	327
20.3.2 OpenType的工作方式	307	22.4.4 考虑存储空间和存储性能	328
20.4 问题	308	22.4.5 确定系统表中的对象标识符	329
20.5 国际化最优方法	308	22.5 其他资源	329
20.5.1 重视所有语言特征	308	22.6 小结	329
20.5.2 注意查询次序	309	第23章 Microsoft Windows文本服务框架 (TSF)	330
20.5.3 考虑存储效率	309	23.1 技术概况	330
20.5.4 使用设备表在尺寸很小的点上 放置发音符号	309	23.1.1 TSF的性能	330
20.5.5 熟悉关于OpenType字体的所有 建议	309	23.1.2 体系结构和对象模型	331
20.6 替代技术	309	23.1.3 在应用程序中使用TSF	334
20.7 其他资源	309	23.2 可用性	334

23.3 国际化特征	334
23.3.1 为一般用户提供对新输入方法的支持	334
23.3.2 对东亚语言的语音和手写体支持	334
23.3.3 无需设备特定或者语言特定的API	335
23.4 其他资源	335
23.5 小结	335
第24章 Uniscribe	337
24.1 技术概况	337
24.1.1 文本行支持	337
24.1.2 文本库支持	337
24.1.3 Uniscribe的版本	338
24.2 可用性	338
24.3 国际化特征	338
24.3.1 字距调整	338
24.3.2 连字	339
24.3.3 组合标记和字符记录	340
24.3.4 RTL和LTR阅读顺序	340
24.3.5 字符和字形间的多对多关系	341
24.3.6 插入符定位	341
24.4 其他资源	341
24.5 小结	342
第25章 Microsoft Visual Studio .NET	343
25.1 工具概况	343
25.2 可用性	343
25.3 国际化特征	344
25.3.1 支持各种编码和字体链接	344
25.3.2 支持本地化的Windows窗体	344
25.3.3 支持本地化的Web应用程序	345
25.3.4 配置工程	346
25.3.5 Visual C++对各种字符编码的支持	346
25.3.6 区域模型	346
25.3.7 MFC和ATL库	346
25.4 其他资源	347
25.5 小结	347
第26章 可扩展标记语言(XML)	348
26.1 技术概况	348

26.2 可用性	349
26.3 国际化特性	350
26.4 国际化最优方法	352
26.4.1 使用UTF-8	352
26.4.2 保证XML数据区域中立以及文化 中立	352
26.4.3 利用MSXML技巧	352
26.4.4 利用System.XML技巧	355
26.5 其他资源	357
26.6 小结	357

第七部分 附录（在光盘中）

附录A	Windows XP中的NLS信息——概述	361
附录B	Windows XP中的NLS信息——详述各个区域的约定	371
附录C	映射到代码页的区域表	506
附录D	语言标识符表	513
附录E	GetLocaleInfo用到的LCTYPE常量	522
附录F	选取的一些语言的排序方法	531
附录G	Unicode编码点分配	532
附录H	Microsoft Windows 2000和Microsoft Windows XP的代码页支持	535
附录I	代码页表	539
附录J	编码Web文档	632
附录K	Microsoft .NET框架中的CultureInfo名称和标识符	640
附录L	Microsoft .NET框架中的RegionInfo名称和标识符	646
附录M	挑选的国际标点符号	652
附录N	有危险的字符	654
附录O	FONTSIGNATURE结构	666
附录P	Microsoft Windows 2000和Microsoft Windows XP支持的键盘和IME列表	674

附录Q 国际键盘布局	678	附录V Microsoft Windows的本地化版本 ...	755
附录R Microsoft Windows XP中的时区.....	730	附录W 准备开发国际化软件的商业案例 ...	759
附录S 国际地址格式	732	附录X Web资源	762
附录T 度量衡及换算	750	附录Y 其他读物	765
附录U 纸张和信封的类型和大小	752	术语表	766

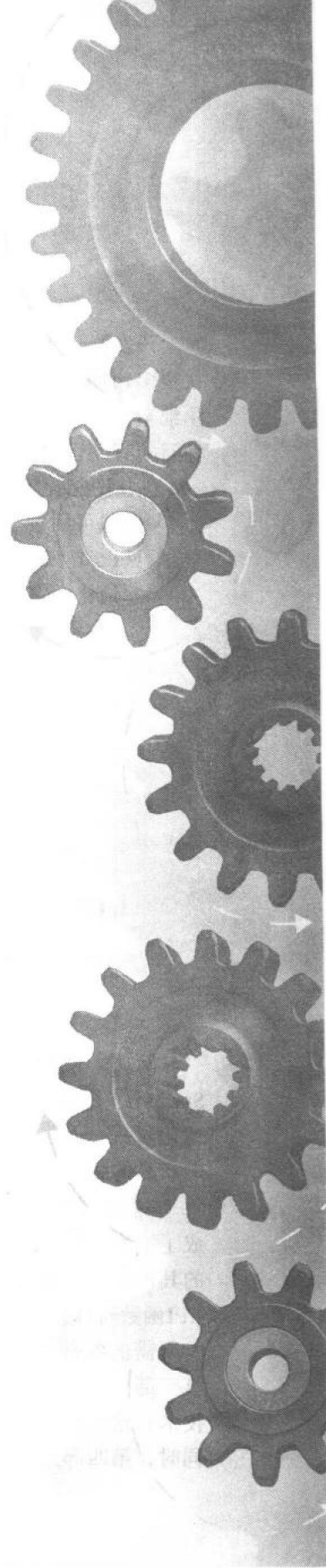
第一部分 简介

第1章 理解国际化

第2章 设计全球可用程序

第一部分明确为国际化产品创建和设计全球可用软件的定义，并给出至关重要的规划技巧。第1章“理解国际化”中，介绍与创建国际化产品有关的两个关键过程：全球可用性和本地化，但在此把全球可用性进一步分成了全球化和本地化能力。同时还考查了其他一些概念：例如区域、区域意识以及本地化。最后，第1章还概述了在发布国际产品时需要考虑的更多问题。

在第2章“设计全球可用程序”中，你将了解设计全球可用软件的固有步骤。更特别地，本章重点列出了在编写规范、定制特征，以及为单一二进制、全球可用程序设计用户界面和代码时需要考虑的问题。第2章还探讨了建立项目组 and 开发环境的策略，并给出与测试和翻译问题有关的技巧。简而言之，第一部分概述对创建国际化产品很重要的基本概念，重点讨论开发全球可用软件。这些概念是本书后面所出现的很多技术细节的骨架。



第1章 理解国际化

开发国际化产品是一种不断均衡的行为。开发人员及其管理人员通常大大低估了创建可以在很多市场使用的、全球可用的单一二进制应用程序或者高质量的外语版本所需的工作量和关注细节。如果你是开发人员，则要确保管理层了解软件开发涉及到的内容。熟悉可能遇到的各种问题有助于你在开发周期开始时就做出正确的决定，从而节省时间和金钱，而不会造成遗憾。

创建国际化软件的过程包括两个方面（如图1-1所示）：

- **全球可用性**。它包括通用编程和设计，以及两个主要的领域——全球化和本地化能力（参见1.2节，了解全球化和本地化能力）。
- **本地化**。它包括为某一特定市场翻译和定制产品。

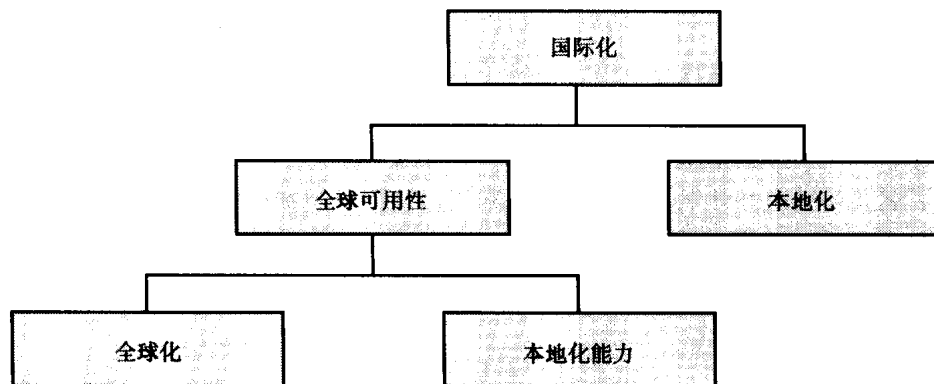


图1-1 软件国际化过程

例如，使用由Microsoft Win32 API所提供的国家语言支持（NLS，National Language Support）就是一个全球化步骤，而修改用户界面（UI）要素、翻译文本以及标准化术语则是本地化过程。开发人员编写代码，所以他们主要关注全球化问题；但由于代码和特征的设计将影响产品的翻译和定制，所以开发人员也应该了解基本的本地化概念。

本书的前两章介绍Microsoft Windows平台上的软件国际化，这将对读者有所帮助，它们构成了本书的第一部分“简介”。这两章讨论创建全球化应用程序的一般概念和技术，定义在本书的其他部分中将使用的很多术语，并为本书后面涉及到的有关使用基于Windows的工具和Win32 API函数特殊技术细节的章节提供了一个框架。第二部分“全球化”中，包括对软件进行全球化时所需的各种步骤，从处理各种不同的语言和文字到以符合特定文化习惯的方式来显示数据。第三部分“本地化能力”中，说明了开发人员在设计和编写软件时可以使用的各种本地化能力技术，这些技术可以使得软件本地化工作人员的本地化过程更容易而且更具有成本效益。与此同时，第四部分“本地化”中，提供了本地化过程自身的背景。