



国际语言环境指南

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

文件号码: 819-0402-10
2005 年 1 月

版权所有 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 保留所有权利。

本产品或文档受版权保护，其使用、复制、发行和反编译均受许可证限制。未经 Sun 及其许可证颁发机构的书面授权，不得以任何方式、任何形式复制本产品或本文档的任何部分。第三方软件，包括字体技术，均已从 Sun 供应商处获得版权和使用许可。

本产品的某些部分从 Berkeley BSD 系统派生而来，经 University of California 许可授权。UNIX 是由 X/Open Company, Ltd. 在美国和其他国家/地区独家许可的注册商标。

Sun、Sun Microsystems、Sun 徽标、docs.sun.com、AnswerBook、AnswerBook2、SunOS 是 Sun Microsystems, Inc. 在美国和其他国家/地区的商标或注册商标。和 Solaris 是 Sun Microsystems, Inc. 在美国和其他国家/地区的商标、注册商标或服务标记。所有 SPARC 商标的使用均需获得许可，它们是 SPARC International, Inc. 在美国和其他国家/地区的商标或注册商标。带有 SPARC 商标的产品均基于 Sun Microsystems, Inc. 开发的体系结构。Netscape Navigator 是 Netscape Communications Corporation 在美国和其他国家/地区的商标或注册商标。PostScript 是 Adobe Systems, Incorporated 的商标或注册商标，后者可能在某些司法管辖区域注册。

OPEN LOOK 和 Sun™ 图形用户界面是 Sun Microsystems, Inc. 为其用户和被许可方开发的。Sun 感谢 Xerox 在研究和开发可视或图形用户界面的概念方面为计算机行业所做的超前贡献。Sun 已从 Xerox 获得了对 Xerox 图形用户界面的非独占许可证，该许可证还适用于执行 OPEN LOOK GUI 和在其他方面遵守 Sun 书面许可协议的 Sun 许可证持有者。

本文档按“原样”提供，对所有明示或默示的条件、陈述和担保，包括对适销性、适用性和非侵权性的默示保证，均不承担任何责任，除非此免责声明的适用范围在法律上无效。



050119@10536



目录

序 13

1 Solaris 国际化概述	17
关于 Solaris 国际化体系结构	17
新增的国际化和本地化特征	18
国际化和本地化概述	21
国际化的基本步骤	21
Solaris 界面中的本地化功能	22
什么是语言环境?	23
C 语言环境—缺省的语言环境	24
完全语言环境和部分语言环境	24
受语言环境影响的行为	25
语言环境种类	25
使用语言环境种类进行本地化	26
时间格式	26
日期格式	27
数字格式	27
国际货币格式	28
语言单词和字母差异	30
单词分隔符	30
排序	30
字符集	30
键盘差异	33
纸张大小的差异	33

- 2 常规国际化功能 35
 - 支持代码集独立性 35
 - CSI 方法 36
 - 具有 CSI 特征的命令 36
 - 具有 CSI 特征的库 37
 - 语言环境数据库 37
 - 进程代码格式 38
 - 多字节支持环境 38
 - 动态链接应用程序 38
 - 更改的界面 39
 - ctype 宏 40
 - libc 中的国际化 API 40
 - genmsg 实用程序 47
 - 用户定义的和用户可扩展的代码转换 48
 - 国际化域名 (IDN) 支持 49

- 3 Solaris 环境中的本地化 51
 - 本地化的软件支持 51
 - Solaris 语言环境软件包摘要 51
 - 支持的语言环境 52
 - 语言环境的多键书写序列 58
 - Solaris 环境中的键盘支持 59
 - 在 SPARC 系统上更改键盘 61
 - 在 Intel 系统上更改键盘 63
 - 键盘布局图解 64
 - 新增 Solaris 键盘软件支持 71
 - ▼ 如何访问爱沙尼亚 6 型 USB 键盘支持 72
 - ▼ 如何访问加拿大法语 6 型 USB 键盘支持 72
 - ▼ 如何访问波兰语程序员 5 型键盘支持 72

- 4 支持的亚洲语言环境 73
 - 日语本地化 73
 - 日语语言环境 73
 - 日语字符集 73
 - 日语字体 74
 - 日语输入系统 75
 - ▼ 如何使用 ATOK 输入法 75

日语终端的终端设置	75
日语 iconv 模块	76
用户定义的字符支持	76
部分语言环境和完全语言环境之间的差别	76
印度语本地化	77
▼ 如何使用印度语输入法	77
印度语键盘	78
了解映射	81
基于连续拼音的输入法的映射	81
连续拼音输入法的工作原理	103
泰语本地化	104
泰语输入法	104
泰语键盘布局	104
泰语输入法辅助窗口	105
5 UTF-8 语言环境支持概述	107
Unicode 概述	107
Unicode 语言环境: en_US.UTF-8 支持	108
关于桌面输入法	110
语言选择和输入模式	110
访问输入法	111
输入模式切换键序列	111
英语/欧洲语言输入模式	112
阿拉伯语输入模式	127
西里尔语输入模式	127
希腊语输入模式	128
希伯来语输入模式	135
日语输入模式	136
朝鲜语输入模式	137
简体中文输入模式	137
繁体中文输入模式	137
繁体中文(中国香港特别行政区)输入模式	138
Unicode 十六进制输入模式	138
查表输入模式	139
系统环境	139
语言环境环境变量	139
TTY 环境设置	140
代码转换	143

- DtMail 支持 144
- 程序设计环境 146
 - 用于 X 应用程序的 FontSet 146
 - CDE/Motif 应用程序中的 FontList 定义 147
- 6 复杂文本布局 149**
 - CTL 技术概述 149
 - CTL 体系结构概述 150
 - 对基于 X 库的应用程序的 CTL 支持 150
 - XOC 资源 150
 - 支持 CTL 技术的 Motif 更改 151
 - XmNlayoutDirection 资源 151
 - XmStringDirection 资源 152
 - XmRendition 资源 152
 - XmText 和 XmTextField 资源 154
 - XmTextFieldGetLayoutModifier 资源 157
 - XmTextGetLayoutModifier 资源 157
 - XmTextFieldSetLayoutModifier 资源 158
 - XmTextSetLayoutModifier 资源 158
 - XmStringDirectionCreate 资源 158
 - UIL 参数 159
 - 开发 CTL 应用程序 159
 - 控制布局方向 160
 - 在资源文件中创建绘制表 163
 - 水平制表符 164
 - 鼠标选定 165
 - 键盘选定 166
 - 文本资源和几何图形 166
 - 移植说明 166
- 7 具有 mp 的打印过滤器增强模式 169**
 - 用于 UTF-8 的打印 169
 - mp 打印过滤器增强模式概述 170
 - 与特定于语言环境的字体配置文件 mp.conf 一起使用 mp 170
 - 与特定于语言环境的 PostScript Prolog 文件一起使用 mp 171
 - 将 mp 用作 Xprt (X 打印服务器) 客户机 171
 - 使用 mp.conf 配置文件的本地化 171

- ▼ 如何添加打印机驻留的字体 175
- ▼ 如何创建共享对象文件 176
- 添加和定制 prolog 文件 177
 - PostScript 文件定制 177
 - .xpr 文件 180

A iconv 代码转换 185

索引 213

表

表 1-1	合法的 UTF-8 字节序列	20
表 1-2	国际时间格式	26
表 1-3	国际日期格式	27
表 1-4	国际数字惯例	28
表 1-5	国际货币惯例	28
表 1-6	支持欧元的用户语言环境	29
表 1-7	德国语言环境和相应的 LC_MONETARY 操作数	30
表 1-8	通用国际页大小	33
表 2-1	libc 中的消息传送函数	41
表 2-2	libc 中的代码转换函数	41
表 2-3	libc 中的正则表达式	41
表 2-4	libc 中的宽字符类	42
表 2-5	libc 中的修改和查询语言环境函数	42
表 2-6	libc 中的查询语言环境数据	42
表 2-7	libc 中的字符分类和拼写	42
表 2-8	libc 中的字符排序	43
表 2-9	libc 中的货币格式	44
表 2-10	libc 中的日期和时间格式	44
表 2-11	libc 中的多字节处理	44
表 2-12	libc 中的宽字符和字符串处理	45
表 2-13	libc 中的格式化宽字符输入和输出	46
表 2-14	宽字符串 libc	46
表 2-15	libc 中的宽字符输入和输出	47
表 2-16	iconv 代码转换	50
表 3-1	亚洲语言环境	52
表 3-2	澳大拉西亚语言环境	53
表 3-3	中美洲语言环境	54

表 3-4	中欧语言环境	54
表 3-5	东欧语言环境	54
表 3-6	中东语言环境	55
表 3-7	北非语言环境	56
表 3-8	北美语言环境	56
表 3-9	北欧语言环境	56
表 3-10	南美语言环境	57
表 3-11	南欧语言环境	57
表 3-12	西欧语言环境	58
表 3-13	使用书写键创建的发音字符	58
表 3-14	对地区键盘的支持	59
表 3-15	4、5 和 5c 型键盘的布局	61
表 4-1	日语位图字体	74
表 4-2	日语 TrueType 字体	75
表 5-1	输入模式切换键序列	112
表 5-2	常用的 Latin-1 书写键序列	113
表 5-3	常用的 Latin-2 书写键序列	117
表 5-4	常用的 Latin-3 书写键序列	118
表 5-5	常用的 Latin-4 书写键序列	119
表 5-6	常用的 Latin-5 书写键序列	120
表 5-7	常用的 Latin-9 书写键序列	120
表 5-8	基于重音死键的书写键序列	121
表 5-9	希腊语输入模式的书写键序列	129
表 5-10	使用三个键的希腊语输入模式的书写键序列	134
表 5-11	使用四个键的希腊语输入模式的书写键序列	135
表 5-12	en_US.UTF-8 所支持的 STREAMS 模块	140
表 5-13	en_US.UTF-8 所支持的 64 位 STREAMS 模块	140
表 6-1	XmRendition 中的新资源	153
表 6-2	Xm CTL 中的新资源	154
表 6-3	UIL	159
表 7-1	可选关键字/值对	175
表 7-2	STARTCOMMON/ENDCOMMON 关键字值	183
表 A-1	与 Unicode 相关的 iconv 可用代码转换模块	185
表 A-2	与 Unicode 和 IBM/Microsoft EBCDIC 及 PC 代码页相关的 iconv 可用代码转换模块	207
表 A-3	可用的 iconv 代码转换—IBM 和 Microsoft EBCDIC/PC 代码页到 UTF-8	209

图形

图 1-1	Solaris 操作系统 中语言环境的功能和结构	21
图 2-1	IDN 至 ACE 转换	49
图 2-2	ACE 至 IDN 转换	49
图 3-1	阿拉伯语键盘	64
图 3-2	比利时语键盘	64
图 3-3	西里尔语（俄语）键盘	64
图 3-4	丹麦语键盘	65
图 3-5	芬兰语键盘	65
图 3-6	法语键盘	65
图 3-7	德语键盘	66
图 3-8	意大利语键盘	66
图 3-9	日语键盘	66
图 3-10	朝鲜语键盘	67
图 3-11	荷兰（荷兰语）键盘	67
图 3-12	挪威语键盘	67
图 3-13	葡萄牙语键盘	68
图 3-14	西班牙语键盘	68
图 3-15	瑞典语键盘	68
图 3-16	瑞士（法语）键盘	69
图 3-17	瑞士（德语）键盘	69
图 3-18	繁体中文键盘	69
图 3-19	土耳其语 F 型键盘	70
图 3-20	土耳其语 Q 型键盘	70
图 3-21	英国键盘	70
图 3-22	美国键盘	71
图 3-23	美国/UNIX 键盘	71
图 4-1	孟加拉语辅音字母映射	81

图 4-2	孟加拉语元音字母映射	82
图 4-3	其他孟加拉语字符映射	83
图 4-4	古吉拉特语辅音字母映射	84
图 4-5	古吉拉特语元音字母映射	85
图 4-6	其他古吉拉特语字符映射	86
图 4-7	果鲁穆奇语辅音字母映射	87
图 4-8	果鲁穆奇语元音字母映射	88
图 4-9	其他果鲁穆奇语字符映射	89
图 4-10	印度语辅音字母映射	90
图 4-11	印度语元音字母映射	90
图 4-12	其他印度语字符映射	91
图 4-13	埃纳德语辅音字母映射	92
图 4-14	埃纳德语元音字母映射	93
图 4-15	其他埃纳德语字符映射	94
图 4-16	马拉雅拉姆语辅音字母映射	95
图 4-17	马拉雅拉姆语元音字母映射	96
图 4-18	其他马拉雅拉姆语字符映射	97
图 4-19	泰米尔语辅音字母映射	98
图 4-20	泰米尔语元音字母映射	99
图 4-21	Telugu 辅音字母映射	101
图 4-22	Telugu 元音字母映射	101
图 4-23	其他 Telugu 字符映射	102
图 5-1	输入模式选择窗口	111
图 5-2	阿拉伯语键盘	127
图 5-3	西里尔语(俄语)键盘	127
图 5-4	希腊语欧式键盘	128
图 5-5	希腊语 UNIX 键盘	129
图 5-6	希伯来语键盘	135
图 5-7	日语键盘	136
图 5-8	朝鲜语键盘	137
图 5-9	DtMail 的“新建邮件”窗口	145
图 6-1	CTL 体系结构	150
图 6-2	布局方向	160
图 6-3	制表符行为	164

序

《国际语言环境指南》介绍 Solaris™ 操作系统（Solaris 操作系统）中新增的国际化特性。本指南包含有关使用当前 Solaris 发行版生成支持各种语言和文化习惯的全球软件产品的信息。

您可以找到有关语言开发和支持基础的概念性信息。您还可以找到便于全球应用程序开发和全世界语言服务管理的 Solaris 特性的描述。

此外，本指南还提供了指向其他文档的链接，这些文档包含本发行版中有关国际化特性的详细信息。

注意 – 本指南中与操作系统有关的所有信息都与 Solaris 操作系统有关。

本前言包含下列几节。

- 第 13 页 “关于本书”
- 第 14 页 “本指南的组织”
- 第 14 页 “相关书籍和站点”
- 第 15 页 “联机访问 Sun 文档”
- 第 15 页 “印刷约定”
- 第 16 页 “命令示例中的 shell 提示符”

关于本书

本指南是为在当前 Solaris 操作系统中设计和支持全球应用程序的软件开发人员和系统管理员而编写的。

本指南假定您有 C 编程语言的知识。

本指南的组织

本指南中的章节是以如下方式组织的：

- 第 1 章介绍可以在当前 Solaris 发行版中使用的新增国际化和本地化特性。
- 第 2 章提供有关编码集独立性 (CSI)、语言环境数据库、libc API 和其他国际化特性的介绍性信息。
- 第 3 章提供有关在当前 Solaris 操作系统中支持使用的语言环境、字体和键盘的信息。
- 第 4 章介绍了当前 Solaris 发行版中对日语、印度语和泰语的本地化支持信息。
- 第 5 章提供了在当前 Solaris 操作系统中支持使用的有关可用的输入法和代码转换功能的信息。
- 第 6 章介绍了复杂文本布局 (CTL) 扩展，它使 Motif API 能支持要求在逻辑文本表示法和物理文本表示法之间进行复杂转换的写入系统。要求进行复杂转换的写入系统包括阿拉伯语、希伯来语和泰语。
- 第 7 章介绍了打印支持，重点介绍了 mp 打印过滤器。
- 附录 A 包含可用的 iconv 转换的表。

相关书籍和站点

下列书籍提供本指南中讨论主题的详细信息：

- Solaris 国际化：
Tuthill、Bill 和 David Smallberg。《*Creating Worldwide Software: Solaris International Developer's Guide*》第 2 版，Mountain View, California, Sun Microsystems Press, 1997。本书可通过 books@sun.com 和 <http://www.sun.com/books/> 获得。本书提供了 Solaris 操作系统中国际化进程的一般概述。
- Solaris 公用桌面环境：
《*Solaris Common Desktop Environment: Programmer's Guide*》也是 Solaris documentation CD 附带的 CDE 开发者集的一部分。
- 中文和朝鲜语 Solaris 语言环境：
《*Korean Solaris User's Guide*》
《简体中文 Solaris 用户指南》
《繁体中文 Solaris 使用者指南》
- OSF/Motif 应用程序开发：

《OSF/Motif Programmer's Guide, Release 1.2》, Englewood Cliffs, New Jersey, Prentice-Hall, 1993。本书是关于如何使用 OSF/Motif 应用程序编程接口来创建 Motif 应用程序的开放软件基金会 (OSF) 指南。

联机访问 Sun 文档

可以通过 docs.sun.comSM Web 站点联机访问 Sun 技术文档。您可以浏览 docs.sun.com 档案或查找某个具体的书的标题或主题。URL 是 <http://docs.sun.com>。

印刷约定

下表描述了本书中使用的印刷约定。

表 P-1 印刷约定

字体或符号	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 .login 文件。 使用 <code>ls -a</code> 列出所有文件。 <code>machine_name% you have mail.</code>
AaBbCc123	您键入的内容，与计算机屏幕输出的内容相对照	<code>machine_name% su</code> 口令：
<i>AaBbCc123</i>	命令行通配符：使用实际名称或实际值替换	要删除文件，请键入 rm <i>filename</i> 。
<i>AaBbCc123</i>	书名、新词、检索词或要强调的词。	请参见《用户指南》的第 6 章。 这些称为类选项。 必须是超级用户才能执行此操作。

命令示例中的 shell 提示符

以下表格显示了 C shell、Bourne shell 和 Korn shell 的缺省系统提示符和超级用户提示符。

表 P-2 Shell 提示符

shell	提示符
C shell 提示符	machine_name%
C shell 超级用户提示符	machine_name#
Bourne shell 和 Korn shell 提示符	\$
Bourne shell 和 Korn shell 超级用户提示符	#

第 1 章

Solaris 国际化概述

本章介绍 Solaris 国际化和本地化的新增特性和重要概念。本章包含下列主题。

- 第 18 页 “新增的国际化和本地化特征”
- 第 21 页 “国际化和本地化概述”
- 第 23 页 “什么是语言环境？”
- 第 26 页 “使用语言环境种类进行本地化”
- 第 30 页 “语言单词和字母差异”
- 第 33 页 “键盘差异”
- 第 33 页 “纸张大小的差异”

关于 Solaris 国际化体系结构

当前 Solaris 发行版包含了许多新特性，包括对 UTF-8 语言环境的 Unicode 4.0 支持，增强了键盘支持并改进了 mp 打印过滤器的几项功能。

Solaris 国际化体系结构简化了世界范围内的应用程序和语言服务的开发、部署和管理。一个多语种产品能对 39 种不同语言和 162 种语言环境提供支持。另外，还支持泰语和印度语书写体所需的复杂文本布局环境。也支持阿拉伯语和希伯来语等语言的双向文本功能。

为许多不同 Solaris 语言环境提供了输入法、字符集、编码集转换和其他与语言相关的特性。您可以通过下列标准 API 在多语言环境中部署应用程序。您还可以定制语言属性，更改转换器表或在 Solaris 环境中添加新输入法编辑器。

Solaris X 全球化框架的源代码于 2000 年秋季向开放源代码社区发布。此发行版使您能采用通用的引用实现来增强全球应用程序的兼容性和互操作性。独立于编码集的全球化方法使您能在当地语言和 Unicode 语言环境中进行操作。Solaris 框架提供了跨平台伸缩的能力。一组丰富的数据转换器确保了在各种编码和不同第三方平台之间的互操作性。

Solaris 平台还使跨国公司能在世界范围内调整其服务器管理规模。与竞争性平台不同，Solaris 平台使用基于服务的方法来管理语言服务。服务器管理员可以通过全球网络远程启用语言服务，而无论什么客户系统。通过这一与客户机无关的方法，您就可以进行系统升级而不必更改客户机应用程序。例如，为了读取从巴黎的网络咖啡屋发送的用阿拉伯文书写的电子邮件，用户不必更改本地客户机应用程序。

新增的国际化和本地化特征

下列新特性可以在当前 Solaris 发行版中获得：

- 自动编码查找器

自动编码查找器是一个用于处理全球字符的实用程序。自动编码查找器通过通用接口，提供了一种简便的方法来检测特定文件或字符串的编码。编码检测简化了对各种语言字符编码的访问。有关更多信息，请参见 `auto_ef(1)` 或 `libauto_ef(3LIB)` 手册页。

- 语言环境管理器

语言环境管理器允许您通过命令行界面来查询和配置 Solaris 操作系统的语言环境。使用 `localeadm(1M)` 工具，您可以显示系统中安装的或驻留在特定设备或目录中的语言环境软件包的有关信息。您可按区域来添加和删除当前系统中的语言环境。例如，您可将东欧区的所有语言环境添加到当前系统中。

在引入语言环境管理器前，一旦安装了系统，您就必须添加/删除单个软件包以更改计算机中的语言环境。在处理单个软件包时，由于容易丢失或忽视软件包，所以易于出错。

语言环境管理器是 Solaris 安装程序中语言环境选择逻辑的补充。此安装程序仍被认为是正确安装 Solaris 语言环境的主应用程序。

- mp 的功能增强

在 Solaris 9 操作系统中第一次发布的 mp 打印过滤器代替了当前 Solaris 发行版中的 `xutops` 打印过滤器。在本发行版中，mp 打印过滤器在下列几个主要方面得到了增强。

- 如果在 `mp.conf` 文件中找不到配置字体，则 mp 程序将继续运行，直到它遇到使用未识别字体的字型。
- 在打印 TrueType 字型时，mp 输出的大小会极大地减少。
- mp 使用的字典机制是经过精心调整的，打印时速度较快。
- TrueType 引擎的功能得到了增强，可以处理各种宽度的所有空格字符

注意 – xutops 打印过滤器在 Solaris 操作系统中不再受支持。xutops 打印过滤器以前用于打印 UTF-8 语言环境中的国际化文本。替代 xutops 的 mp 打印过滤器是 xutops 打印过滤器的支持特性的超集。有关更多信息，请参见 mp(1) 手册页。

- 新的欧洲键盘支持

对波兰语程序员 5 型键盘和 Sun Ray™ USB 6 型俄语、爱沙尼亚语和加拿大法语键盘提供了 Sun I/O 键盘支持。

注意 – 当前，对新的欧洲键盘类型而言没有可用的硬件。要使用新的键盘软件，请参考第 71 页“新增 Solaris 键盘软件支持”中的过程。

- Unicode 4.0 支持

下列 UTF-8 语言环境已得到更新以支持新的 4.0 版的 Unicode 标准：

- ar_EG.UTF-8
- de_DE.UTF-8
- en_US.UTF-8
- es_ES.UTF-8
- fi_FI.UTF-8
- fr_BE.UTF-8
- fr_FR.UTF-8
- he_IL.UTF-8
- hi_IN.UTF-8
- it_IT.UTF-8
- ja_JP.UTF-8
- ko_KR.UTF-8
- pl_PL.UTF-8
- pt_BR.UTF-8
- ru_RU.UTF-8
- sv_SE.UTF-8
- th_TH.UTF-8
- tr_TR.UTF-8
- zh_CN.UTF-8
- zh_HK.UTF-8
- zh_TW.UTF-8

新版本的标准另引入了 1,226 个新字符并包含各种标准的和信息性更改。

Unicode 3.2 将更为严格的 UTF-8 字节序列定义为“UTF-8 Corrigendum”：

表 1-1 合法的 UTF-8 字节序列

代码点	第 1 字节	第 2 字节	第 3 字节	第 4 字节
U+0000..U+007F	00..7F			
U+0080..U+07FF	C2..DF	80..BF		
U+0800..U+0FFF	E0	A0..BF	80..BF	
U+1000..U+CFFF	E1..EC	80..BF	80..BF	
U+D000..U+D7FF	ED	80..9F	80..BF	
U+D800..U+DFFF	错误形式			
U+E000..U+FFFF	EE..EF	80..BF	80..BF	
U+10000..U+3FFFF	F0	90..BF	80..BF	80..BF
U+40000..U+FFFFF	F1..F3	80..BF	80..BF	80..BF
U+100000..U+10FFFF	F4	80..8F	80..BF	80..BF

这些序列不包含 U+D800 和 U+DFFF 之间的代理代码点。这些序列还限制了任何其他非法的字节值。为遵守此新定义，Unicode 语言环境方法和 UTF-8 iconv 模块的功能已得到增强，以检测新定义的 UTF-8 无效字节序列。

■ 泰语键盘布局

下列键盘布局支持泰语输入法：

- Kedmanee (TIS820-2531) 键盘布局。Kedmanee 布局是设计用于打字机上的，而非用于计算机键盘。打字机键盘的有限键数表明，布局中部分特殊泰语字符不可用。TIS820-2531 已改变了 Kedmanee 布局，使其能用于计算机键盘。
- TIS820-2538 键盘布局。这个增强了的 Kedmanee 布局是一个更新版本的 TIS820-2531 布局，它包含一些在原 Kedmanee 布局中不可用的泰语特殊字符。当前，TIS820-2538 是由泰国行业标准协会颁布的唯一的泰语键盘布局标准。
- Pattajoti 键盘布局。Pattajoti 布局也是设计用在打字机上的，但指压分布比较合理。Pattajoti 是由皇家灌溉局官员发明的，现在仍在这个部门中得到广泛使用。
- 适用于泰语输入法的可配置的键盘布局，用户定义的键盘布局。

■ 对印度语的输入法支持

与中文输入法使用的界面类似的代码表输入法界面可以在本发行版中获得。基于 IIMF SDK 和 SunIM 语言界面的印度语输入法提供了下列新特性：

- 基于语音、音译的输入法和键盘布局。支持的键盘布局在 ISCII 标准中被定义为 INSCRIPT 键盘布局。
- 标准输入法切换。
- 印度语手写体包含印度语、泰米尔语、埃纳德语、马拉雅拉姆语、泰卢固语、古吉拉特语、旁遮普语和孟加拉语。您可以通过按 F5 键更改输入手写体。
- 新键盘布局的插件机制便于在将来进行扩展。插件配置文件将通过语言引擎模块进行加载。

国际化和本地化概述

国际化和本地化是两个不同的过程。**国际化**是指使软件可在多个语言或地区之间移植的过程，而**本地化**则是指使软件适用于特定语言或地区的过程。国际化的软件可以使用在运行时根据特定文化要求修改程序行为的接口来开发。本地化涉及到建立联机信息以支持某个语言或地区（称为**语言环境**）。

与必须彻底重新编写才能用于不同当地语言和习惯的软件不同，国际化软件不需要进行重新编写。国际化的软件可在不进行更改的情况下从一个语言环境移植到另一个语言环境。**Solaris** 系统是国际化的系统，提供了创建国际化软件所需的基础结构和接口。

国际化的基本步骤

国际化应用程序的可执行映像可在各语言和地区之间进行移植。要对软件进行国际化：

- 使用本手册中描述的接口创建具有通过动态重新编译即可修改的环境的软件。
- 将软件分为可执行代码和用户可以看到的所有消息。将信息字符串保存在信息目录中。

针对某一语言或地区对信息字符串进行翻译。**语言环境**包括信息字符串和指定排序的方法。

要使用某个产品的本地化版本，用户需要设置一些环境变量。然后，此产品将显示已转换为该语言环境语言的信息。日期、时间、货币和其他信息根据特定于语言环境的惯例进行格式设置和显示。信息翻译和联机帮助内容在各个不同的层中提供，如下图所述。

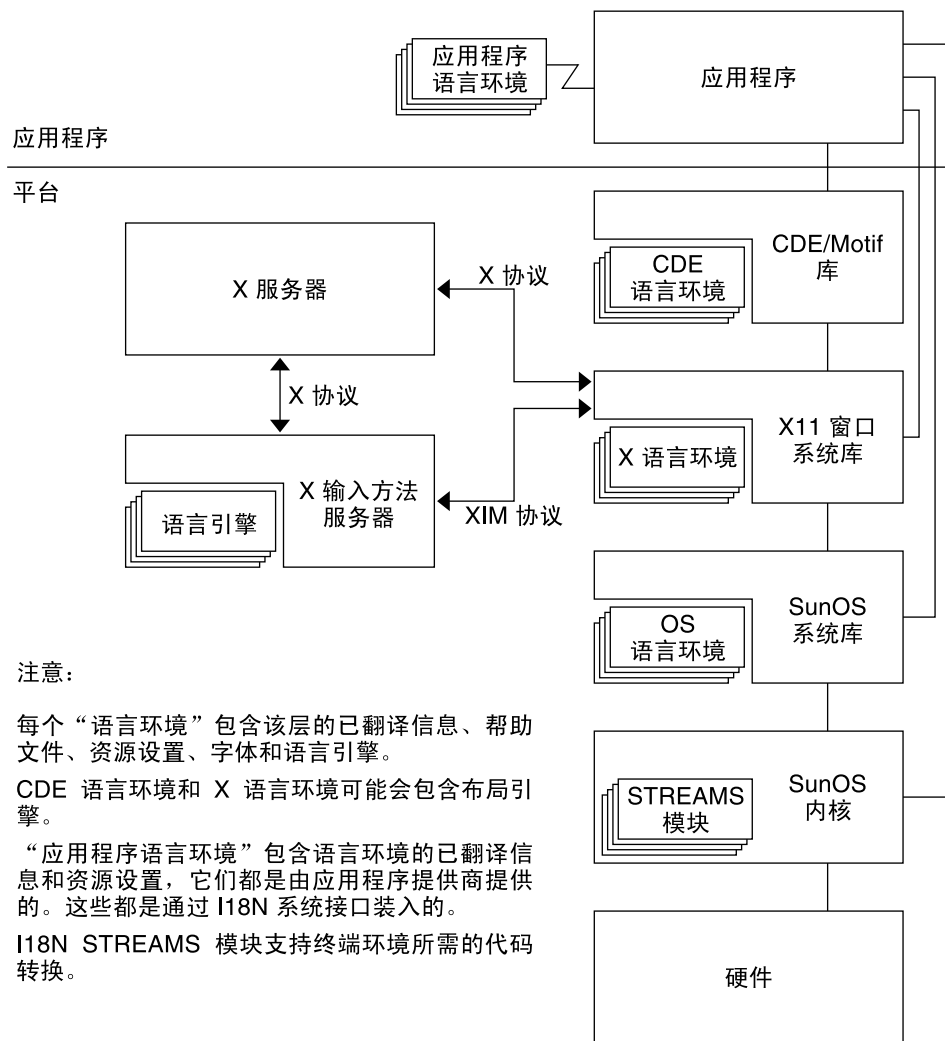


图 1-1 Solaris 操作系统中语言环境的功能和结构

Solaris 界面中的本地化功能

OS（操作系统）语言环境层提供基本的语言环境数据库和功能，这些数据库和功能在应用程序运行时插入到 OS 系统接口中。应用程序通过标准 API 访问这些 OS 语言环境模块。

X11 语言环境层将此接口提供给 X 输入法，并将 X 输出方法提供给 X11 应用程序，以便可以进行本地文本输入和显示。这些字体使应用程序能够显示各种语言中的字符。

CDE/Motif 构建在 X11 窗口系统之上。因此，CDE/Motif 可以通过 X11 API 使用 X11 语言环境功能。对于 CDE 应用程序，Solaris 本地化具有各种特定于语言环境的配置，从而使桌面能够在目标语言环境中工作。在各个不同的层中提供了信息翻译和联机帮助内容。

什么是语言环境？

程序的**语言环境**是应用程序的一个重要概念。语言环境是指当地语言环境的显式模型和定义。ANSI C 语言标准的库定义中明确定义并包含了语言环境的概念。

一个语言环境包括许多种类，对于这些种类，存在与国家/地区有关的格式设置或其他规范。一个程序的语言环境对程序的编码集、日期和时间格式设置惯例、货币惯例、十进制格式设置惯例和排序顺序进行定义。

语言环境可以由基本语言、使用国家（地区）和可选的编码集组成。通常采用代码集。例如，德语是 `de`（Deutsch 的缩写），而瑞士德语是 `de_CH`（CH 是 Confederation Helvetica 的缩写）。这种转换使得能够区分国家/地区之间的特定差异，如货币单位表示法。

一个特定语言可以与多个语言环境相关联，这使得能够区分地区间的差异。例如，美国的英语用户可以选择 `en_US` 语言环境（美国英语），而英国的英语用户可选择 `en_GB`（英国英语）。

语言环境名称通常由 `LANG` 环境变量指定。语言环境种类从属于 `LANG`，但也可单独设置，这时它们将覆盖 `LANG`。如果设置了 `LC_ALL` 操作数，则它会覆盖 `LANG` 和所有不同的语言环境种类。

语言环境命名惯例如下：

语言[_地区][.代码集] [@modifier]

其中，双字母**语言**代码来自 ISO 639，并且双字母**地区**代码来自 ISO 3166，**代码集**是在此语言环境中使用的代码集的名称，而**修饰符**是将此语言环境与不带此修饰符的语言环境相区分的特征名称。

所有 Solaris 产品语言环境都保留了具有 US-ASCII 代码值的可移植字符集 (PCS) 字符。

有关可移植字符集的更多信息，请参考“X/Open CAE 规范：系统接口定义，第 5 期” (ISBN 1-85912-186-1)。

一个语言环境可以有多个语言环境名称。例如，POSIX 与 C 是相同的语言环境。

C 语言环境—缺省的语言环境

C 语言环境，也称为 POSIX 语言环境，是所有 POSIX 兼容系统的 POSIX 系统的缺省语言环境。Solaris 操作系统是 POSIX 系统。单个 UNIX 规范，版本 3，定义 C 语言环境。请进行注册以在下列网站读取和下载规范

: <http://www.unix.org/version3/online.html>.

您可以采用两种方法中的一种指定在 C 语言环境中运行的国际化程序：

- 取消对所有语言环境的环境变量的设置。

```
system% unsetenv LC_ALL LANG LC_CTYPE LC_COLLATE LC_NUMERIC \  
LC_TIME LC_MONETARY LC_MESSAGES
```

取消对所有语言环境的环境变量的设置。在 C 语言环境中运行应用程序。

- 将语言环境显式设置为 C 或 POSIX。

```
system% setenv LC_ALL C  
system% setenv LANG C
```

某些应用程序检查 LANG 环境变量，而不是实际调用 `setlocale(3C)` 来引用当前语言环境。此时，`setenv` 通过指定 `LC_ALL` 和 `LANG` 语言环境的环境变量来显式设置 C 语言环境。有关语言环境的环境变量之间的优先级关系，请参见 `setlocale(3C)` 手册页。

要检查终端环境中的当前语言环境设置，请运行 `locale(1)` 命令。

```
system% locale
```

完全语言环境和部分语言环境

完全 Solaris 语言环境具有列出的所有功能和相关语言的本地化系统信息。部分语言环境未安装任何本地化信息。只要安装了相关语言的本地化信息，Solaris 环境中的所有语言环境便能够显示这些本地化信息。例如，下列语言环境既可以是部分语言环境，也可以是完全语言环境：

- `de_DE.ISO8859-1`
- `de_DE.ISO8859-15`
- `de_DE.UTF-8`
- `de_AT.ISO8859-1`
- `de_AT.ISO8859-15`
- `de_CH.ISO8859-1`

在从 Language CD 中安装了德语信息翻译后，以上所有语言环境均成为**完全语言环境**，因为它们可以访问完全翻译了的桌面。此 Language CD 包含下列语言和语言环境的信息翻译：

- 德语
- 法语
- 西班牙语

- 瑞典语
- 意大利语
- 日语
- 朝鲜语
- 简体中文语言环境
- 繁体中文语言环境

所有部分语言环境都可从 Software CD 中获得。信息翻译可从 Languages CD 中获得。

所有英语语言环境也是完全语言环境，并可从 Software CD 中获得。

受语言环境影响的行为

在格式化数字、书写日期和时间、分隔单词和短语或引用书面和口语材料方面，不同文化通常有不同的习惯。语言环境确定不同地区对下列操作、文件、格式和表达式的不同处理方式：

- 文本数据的编码和处理
- 资源文件的语言标识和编码
- 文本字符串的绘制和布局
- 客户机间的文本的互换
- 满足所选书写体的代码集和文本处理要求的输入法选择
- 文化特定的字体和图标文件
- 操作和文件类型
- 用户界面定义 (UID) 文件
- 日期和时间格式
- 数字格式
- 货币格式
- 排序顺序
- 特定于语言环境的正则表达式处理
- 信息性和诊断性信息与交互式响应的格式

Solaris 环境将语言和与文化有关的信息与应用程序分开，并在应用程序以外将这些信息保存起来。使用此方法就不会针对每个市场对应用程序进行翻译、重新编写或重新编译了。进入新市场的唯一要求就是将外部信息本地化为本地语言和习惯。

语言环境种类

语言环境种类如下所示：

LC_CTYPE	控制字符处理功能的行为。
LC_TIME	指定日期和时间格式，包括月份名称、每周日期和通用的完整表示法和缩写表示法。
LC_MONETARY	指定货币格式，包括此语言环境的货币符号、千位分隔符、符号位置、小数位数等。

LC_NUMERIC	指定十进制分隔符（或基数字符）、千位分隔符和分组。
LC_COLLATE	指定排序顺序和此语言环境的正则表达式定义。
LC_MESSAGES	指定编写本地化信息的语言及此语言环境的肯定和否定响应（yes 和 no 字符串和表达式）。
LO_LTYPE	指定提供有关语言绘制信息的布局引擎。语言绘制（或文本绘制）取决于书写体的形状和方向属性。

使用语言环境种类进行本地化

产品的本地化应通过向该目标语言或地区的本地用户咨询来完成。某些信息样式和格式在开发人员看来可能非常直观和常见，而在用户看来却显得很别扭或者是错误的，甚至会令人不快。下列各节描述了 Solaris 操作系统中的某些元素，您可以定制这些元素以满足产品的本地化要求。

时间格式

下表显示了一些不同的语言环境中 11:59 P.M 的写法。

表 1-2 国际时间格式

语言环境	格式
加拿大语	23:59
芬兰语	23.59
德语	23.59 Uhr
挪威语	23.59
泰语	23:59
英国英语	23:59

时间用 12 小时制和 24 小时制两种方式表示。小时和分钟分隔符可以是冒号 (:) 或句点 (.)。

在不同的国家/地区之间以及国家/地区内部存在着时区划分。尽管时区可以根据早于或晚于协调通用时间 (UTC) 或格林威治时间 (GMT) 的小时数来描述，但该数字并不总是整数。例如，纽芬兰所在的时区与相邻时区相差半小时。

夏时制时间 (DST) 开始和结束的日期可能因国家/地区的不同而不同。很多国家/地区根本不实行夏时制。此外，夏时制时间还可能在某一个时区内部存在差别。例如，美国不同的州有不同的夏时制时间。

日期格式

下表显示在世界各地使用的一些日期格式。即使在一个国家/地区内部，日期格式可能也有所不同。

表 1-3 国际日期格式

语言环境	惯例	实例
加拿大语（英语）	dd/mm/yy	24/08/01
丹麦语	yyyy-mm-dd	2001-08-24
芬兰语	dd.mm.yyyy	24.08.2001
法语	dd/mm/yyyy	24/08/2001
德语	yyyy-mm-dd	2001-08-24
意大利语	dd/mm/yy	24/08/01
挪威语	dd-mm-yy	24-08-01
西班牙语	dd-mm-yy	24-08-01
瑞典语	yyyy-mm-dd	2001-08-24
大不列颠	dd/mm/yy	24/08/01
美国	mm-dd-yy	08-24-01
泰语	dd/mm/yyyy	24/08/2001

数字格式

英国和美国是世界上使用句点表示小数点位置的为数不多的几个地方中的两个。而其他许多国家/地区都使用逗号。小数点分隔符又称为**基数**字符。同样，英国和美国使用逗号分隔千位组，而其他许多国家/地区则使用句点，同时还有一些国家/地区使用短空格来分隔千位组。

包含特定于某一语言环境的格式的数据文件在传送到其他语言环境的系统时经常被错误地解码。例如，一个包含法语格式数字的文件不能用于特定于英国的程序。

下表显示了一些常用的数字格式。

表 1-4 国际数字惯例

语言环境	较大数字
加拿大语（英语）	4,294,967.00
丹麦语	4.294 967.295,00
芬兰语	4 294 967 295,00
法语	4 294 967 295,00
德语	4,294,967.00
意大利语	4.294.967,00
挪威语	4.294.967.295,00
西班牙语	4.294.967.295,00
瑞典语	4 294 967 295,00
大不列颠	4,294,967,295.00
美国	4,294,967,295.00
泰语	4,294,967,295.00

注意 – 没有特定的语言环境惯例用来指定列表中数字的分隔方式。

国际货币格式

货币单位和表示规则在世界各地差别很大。本地和国际货币符号可能是不同的。下表显示了一些国家/地区的货币格式。

表 1-5 国际货币惯例

语言环境	货币	实例
加拿大语（英语）	美元 (\$)	\$1,234.56
加拿大语（法语）	美元 (\$)	1 234,56\$
丹麦语	克朗 (kr)	Kr 1.234,56
芬兰语	欧元 (€)	€ 1 234,56
法语	欧元 (€)	€ 1,234
日语	日元 (¥)	¥ 1,234

表 1-5 国际货币惯例 (续)

语言环境	货币	实例
挪威语	克朗 (kr)	kr 1.234,56
瑞典语	克朗 (Kr)	1 234,56 Kr
大不列颠	英镑 (£)	£1,234.56
美国	美元 (\$)	\$1,234.56
泰语	铢	2539 铢
欧元	欧元 (€)	€ 5,000

当前发行版支持欧元。本地货币符号仍然可用，以保持向后兼容性。

表 1-6 支持欧元的用户语言环境

地区	语言环境名称	ISO 代码集
奥地利	de_AT.ISO8859-15	8859-15
比利时 (法语)	fr_BE.ISO8859-15	8859-15
比利时 (佛兰德语)	nl_BE.ISO8859-15	8859-15
丹麦	da_DK.ISO8859-15	8859-15
爱沙尼亚	et_EE.ISO8859-15	8859-15
芬兰	fi_FI.ISO8859-15	8859-15
法国	fr_FR.ISO8859-15	8859-15
德国	de_DE.ISO8859-15	8859-15
大不列颠	en_GB.ISO8859-15	8859-15
爱尔兰	en_IE.ISO8859-15	8859-15
意大利	it_IT.ISO8859-15	8859-15
荷兰	nl_NL.ISO8859-15	8859-15
葡萄牙	pt_PT.ISO8859-15	8859-15
西班牙加泰罗尼亚	ca_ES.ISO8859-15	8859-15
西班牙	es_ES.ISO8859-15	8859-15
瑞典	sv_SE.ISO8859-15	8859-15
美国	en_US.ISO8859-15	8859-15

欧洲语言环境基于 ISO8859-15 代码集。

请记住，**经过转换**的货币金额可能比原始货币金额占用更多或更少的空间，例如 \$1,000 可能变为 €1.307.000。

欧元地区中语言环境的当前语言环境设置的状态由 locale 实用程序的 LC_MONETARY 操作数表示。例如，下表显示了“德国”的状态。

表 1-7 德国语言环境和相应的 LC_MONETARY 操作数

语言环境	LC_MONETARY
de_DE.ISO8859-1	DM
de_DE.ISO8859-15	Euro
de_DE.UTF-8	Euro
de_DE.ISO8859-15@euro	Euro
de_DE.UTF-8@euro	Euro

语言单词和字母差异

本部分描述各种语言之间的重要差异。

单词分隔符

在英语中，单词通常用空格字符来分隔。而在中文、日语和泰语等语言中，单词之间通常没有分隔符。

排序

特定字符的排序顺序在所有语言中并非都是相同的。例如，在德国，“ö”与普通的“o”的排序顺序相同，而在瑞典，两者却分别进行排序，其中前者位于字母表的末尾。在某些语言中，字符具有确定字符序列优先级的权值。例如，泰语词典通过粗细不同的字符序列来定义排序。

字符集

各字符集在字母字符和特殊字符的数量上可能有所不同。英语字母表只包含 26 个字符，而某些语言却包含更多字符。例如，日语可以包含 20,000 多个字符，而中文可能包含比这一数据还要多的字符。

西欧字母表

大多数西欧国家/地区的字母表类似于英语国家/地区使用的标准的 26 个字符的字母表，但往往还包含一些附加基本字符、一些带标记（或带重音）的字符和一些连字符。

日语文本

日语文本由三种不同的书写体混合而成：

- 源自中文的日语汉字
- 平假名和片假名，即两种语音书写体（或音节）

尽管平假名中的每个字符在片假名中都有一个等效字符，但平假名是最常用的书写体，具有类似草书而不是块状的字母形式。日语汉字字符用于拼写词根。片假名通常用于表示“外来”词，即从日语以外的其他语言中引入的词。

日语汉字包含成千上万个字符，但其中常用字符数正在逐年减少。现在经常使用的字符只有 3500 个，而普通日本作家的日语汉字字符的词汇量大约为 2000 个。然而，按照日本工业标准 (JIS) 的要求，计算机系统必须支持 7000 个以上的常用字符。此外，还有大约 170 个平假名和片假名字符。平均来讲，日语文本中有 55% 是平假名，35% 是日语汉字，10% 是片假名。日语文本中也包含阿拉伯数字和罗马字母。

尽管可以完全避免使用日语汉字，但大多数日语读者发现一篇不带任何日语汉字的文本很难理解。

朝鲜语文本

朝鲜语文本可使用称为“朝鲜语文字”的语音文字系统进行拼写。朝鲜语文字包含 11,000 个字符，这些字符由称为“字母”的辅音和元音组成。朝鲜语计算机系统中通常使用的字符大约有 3000 个，这些字符来自整个朝鲜语文字字符词汇。朝鲜语还使用名为“朝鲜语汉字”的象形字，这些象形字基于中国发明的字符集。朝鲜语文本需要 6000 多个朝鲜语汉字字符。当朝鲜语文字表意不太明确时，常使用朝鲜语汉字来避免混淆。朝鲜语文字字符由组合在一起的辅音和元音构成。两者组合在一起可构成音节，即朝鲜语文字字符。朝鲜语文字字符通常排列在一个方框中，以便形成的字符组与朝鲜语汉字字符占用相同的空间。朝鲜语文本中还包含阿拉伯数字、罗马字母和特殊符号字符。

泰语文本

一个泰语字符可以在显示屏上定义为包含四个显示单元的列位置。每个列位置最多可包含三个字符。显示单元的组成基于泰语字符的分类。某些泰语字符可由其他字符的分类组成。如果这两个字符组合在一起，则它们就位于同一单元中。否则，它们将分别位于不同的单元中。

中文文本

中文通常全部由称为汉字的象形书写体中的字符组成。

- 在中国 (PRC), GB2312 (zh 语言环境) 中大约有 7000 个常用的汉字字符, GBK 字符集 (zh.GBK 语言环境) 中有 20,000 多个字符, GB18030-2000 字符集 (zh_CN.GB18030 语言环境) 中大约有 30,000 个字符, 包括在 Unicode 3.0 中定义的所有 CJK 扩展 A 字符。
- 在中国台湾地区, 最常用的字符集是 CNS11643-1992 (zh_TW 语言环境) 和 Big5 (zh_TW.BIG5 语言环境)。它们共同使用大约 13,000 个汉字字符。
- 在香港特别行政区, Big5 字符集中添加了 4702 个字符, 从而形成 Big5-HKSCS 字符集 (zh_HK.BIG5HK)。

如果字符不是根字符, 则它通常由两个或多个部分组成, 而两个部分最为常见。在两部分字符中, 一部分通常表示意义, 一部分则表示发音。有时两部分都表示意义。偏旁部首是最重要的元素, 而字符传统上是按偏旁部首进行排列的, 偏旁部首的数量有几百个。一个发音可用多个不同的字符表示, 这些字符是不可交替使用的。一个字符可以有多个发音。

在给定上下文中使用某些字符比使用其他字符更为适合。适当的字符通过使用声调从语音上加以区分。与此相反, 日语口语和朝鲜语口语则没有声调。

几种表示中文的语言系统。在中国, 最常用的是**拼音** (使用罗马字符), 它在西方经常用于表示地点, 如 **Beijing**。威托玛式系统是一种比较陈旧的语音系统, 曾经用于表示地名 (如 **Peking**)。而在中国台湾地区, 经常使用的则是**注音** (或**汉语拼音字母**), 它是一种具有独特字母形式的语音字母表。

希伯来语文本

希伯来语文本用于拼写希伯来语和意第绪语书写体。希伯来语是一种双向书写体。希伯来语字母是从右向左进行读写的, 而数字则是从左向右读的。嵌入到希伯来语文本中的所有英语文本也是从左向右读的。

希伯来语使用 27 个字符的字母表, 并使用标准拉丁语 (或英语) 字母表中的标点符号和数字。希伯来语文本还包含元音和发音标记。这些标记可以通过基本字符内部的点 (**dagesh**)、字符下的元音标记或字符左上角的重音等形式出现。这些标记通常只用于礼拜式文本中, 而很少在日常生活中见到。希伯来语中没有大写字母。

印度语文本

印度语文本用称为“天城书” (意思是“神的手笔”) 的书写体书写。印度语是一种语音语言, 并通过一系列音节进行拼写。每个音节都由三种类型的字母片段 (天城书字符) 构成: 辅音字母、独立元音和依赖元音符号。音节本身由辅音和元音核心组成, 同时还包含一个可选的前缀辅音。与英语 (从基准线开始) 不同, 天城书字符从写在字符顶部的水平线 (称为头笔) 悬垂向下书写。这些字符可根据它们的上下文进行组合或更改。同希伯来语一样, 印度语文本不区分大小写字母。

键盘差异

美国键盘上的所有字符并非都在其他键盘中出现。同样，其他键盘经常包含许多在美国键盘上看不到字符。

由于输入由 Solaris 操作系统处理，因此可使用任何键盘输入任何语言环境中的字符。

注意 – 在 SPARC® 机器上，书写键可用于生成所支持的任何 ISO8859 字符集中的任何带有音调符号的拉丁字符。书写键可在基于拉丁文的语言环境中使用，但不能用于朝鲜语、中文或日语语言环境（UTF-8 语言环境除外）。

纸张大小的差异

在每个国家/地区中，经常使用几种纸张大小。通常情况下，其中的某个大小要比其他的大小更通用。大多数国家/地区都遵循 ISO 标准 216：“书写纸和某些种类的打印材质-缩减大小-A 和 B 大小系列。”

国际化应用程序不应假设可以使用的页大小进行假设。Solaris 系统不支持对输出页大小进行跟踪。由应用程序负责跟踪输出页大小。下表显示了通用的国际页大小。

表 1-8 通用国际页大小

纸型	尺寸	国家/地区
ISO A4	21.0 厘米 X 29.7 厘米	除美国以外的任何地方。
ISO A5	14.8 厘米 X 21.0 厘米	除美国以外的任何地方。
JIS B4	25.9 厘米 X 36.65 厘米	日本
JIS B5	18.36 厘米 X 25.9 厘米	日本
美国信函	8.5 英寸 X 11 英寸	美国和加拿大
美国标准	8.5 英寸 X 14 英寸	美国和加拿大

第 2 章

常规国际化功能

本章讨论 Solaris 操作系统中包含的若干个国际化功能。本章包含以下主题：

- 第 35 页 “支持代码集独立性”
- 第 37 页 “语言环境数据库”
- 第 38 页 “进程代码格式”
- 第 38 页 “多字节支持环境”
- 第 38 页 “动态链接应用程序”
- 第 39 页 “更改的界面”
- 第 40 页 “ctype 宏”
- 第 40 页 “libc 中的国际化 API”
- 第 47 页 “genmsg 实用程序”
- 第 48 页 “用户定义的和用户可扩展的代码转换”
- 第 49 页 “国际化域名 (IDN) 支持”

支持代码集独立性

EUC 是扩展 UNIX® 代码的缩写。Solaris 操作系统支持非 EUC 编码，如日本的 PC-Kanji（更常用的叫法是 Shift_JIS）、中国台湾地区的 Big5 以及中华人民共和国的 GBK。由于很大一部分计算机市场需要非 EUC 代码集支持，所以当前 Solaris 环境提供一个完整的框架，以同时支持 EUC 和非 EUC 代码集。这种支持称为**代码集独立性**或 CSI。

CSI 的目标是从 Solaris 操作系统库和命令中移除对特定代码集或编码方法的依赖性。CSI 体系结构使 Solaris 操作系统可支持所有 UNIX 文件系统安全编码。CSI 支持许多新的代码集，如 UTF-8、PC-Kanji 和 Big5。

CSI 方法

代码集独立性使应用程序和平台软件开发者能够使其代码独立于任何编码（如 UTF-8），同时 CSI 还提供了无须修改源代码即可采用任何新编码的能力。该体系结构方法与 Java™ 国际化的不同之处在于：应用程序无需依赖于 UTF-16。

许多现有的国际化应用程序（例如，Motif）自动从基础系统继承 CSI 支持。这些应用程序无须修改即可在新的语言环境中工作。

CSI 本身独立于任何代码集。但是，下列关于文件代码编码（代码集）的假定依然适用于当前 Solaris 系统：

- 文件代码是 ASCII 的超集。
- NULL 字节值 (0x00) 不显示为多字节字符字节的一部分，从而支持空终止多字节字符串。
- ASCII 斜线字符字节值 (0x2f) 不显示为多字节字符字节的一部分，从而支持 UNIX 路径名。

具有 CSI 特征 的命令

本节列出了当前 Solaris 环境中具有 CSI 特征的命令。每个命令的手册页都有一个属性部分，指明该命令是否具有 CSI 特征。

所有命令都位于 /usr/bin 目录中，除非另有说明。

/usr/lib/diffh	/usr/xpg4/bin/pr	compress
/usr/sbin/accept	/usr/xpg4/bin/rm	cpio
/usr/sbin/reject	/usr/xpg4/bin/sed	csch
/usr/ucb/lpr	/usr/xpg4/bin/sort	csplit
/usr/xpg4/bin/awk	/usr/xpg4/bin/tail	cut
/usr/xpg4/bin/cp	/usr/xpg4/bin/tr	diff
/usr/xpg4/bin/date	/usr/xpg4/bin/vedit	diff3
/usr/xpg4/bin/du	/usr/xpg4/bin/vi	disable
/usr/xpg4/bin/ed	/usr/xpg4/bin/view	echo
/usr/xpg4/bin/edit	acctcom	expand
/usr/xpg4/bin/egrep	apropos	file
/usr/xpg4/bin/env	batch	find
/usr/xpg4/bin/ex	bdiff	fold
/usr/xpg4/bin/expr	cancel	ftp
/usr/xpg4/bin/fgrep	cat	gencat
/usr/xpg4/bin/lp	catman	geteopt
/usr/xpg4/bin/ls	chgrp	getoptcvt
/usr/xpg4/bin/more	chmod	head
/usr/xpg4/bin/mv	chown	join
/usr/xpg4/bin/nice	cmp	jsh
/usr/xpg4/bin/nohup	col	kill
/usr/xpg4/bin/od	comm	ksh

lp	rcp	tabs
man	red	tar
mkdir	remsh	tee
msgfmt	rksh	touch
news	rsh	tty
nroff	rsmdir	uncompress
pack	script	unexpand
paste	sdiff	uniq
pcat	settime	unpack
pg	sh	wc
printf	split	whatis
priocntl	strconf	write
ps	strings	xargs
pwd	sum	zcat

具有 CSI 特征的库

libc (/usr/lib/libc.so) 中的几乎所有的函数都具有 CSI 特征。但是，libc 中的下列函数不具有 CSI 特征，因此它们是依赖于 EUC 的函数：

- csetcol()
- csetlen()
- csetno()
- euccol()
- euclen()
- eucscol()
- getwidth()
- wcsetno()

在当前 Solaris 环境中，libgen /usr/ccs/lib/libgen.a 和 libcurse /usr/ccs/lib/libcurses.a 已经被国际化，但不具有 CSI 特征。

语言环境数据库

语言环境数据库的格式和结构是专用的，并且在将来的发行版中可能被修改。在开发国际化应用程序时，请使用 libc 中的国际化 API。在第 40 页“libc 中的国际化 API”中，而不是在指向语言环境数据库的链接中介绍这些 API。

注意 – 在 Solaris 环境中工作时，请使用当前 Solaris 发行版附带的语言环境数据库。不要使用以前的 Solaris 版本中的语言环境。

进程代码格式

进程代码格式（在 Solaris 操作系统 产品中，也称为宽字符代码格式）是专用的，并且在将来的发行版中可能被修改。因而，在开发国际应用程序时，不要假定进程代码格式是相同的。相反，请使用 libc 中的国际化 API，如第 40 页“libc 中的国际化 API”中所述。

注意 – 所有 Unicode 语言环境的进程代码都在 UTF 32 表示法中。有关 UTF 32 的详细信息，请参阅 Unicode 标准附件 19 号：UTF 32 和 Unicode 标准附件 27 号：Unicode 3.1（它们可以从 Unicode 协会或 <http://www.unicode.org/> 得到）。

多字节支持环境

多字节字符是无法存储在单个字节中的字符，如中文、日语或朝鲜语字符。这些字符需要 2、3 或 4 个字节的存储空间。可在 ISO/IEC 9899:1990 3.13 款中找到更准确的定义。

ANSI C 修订 1（也称为 ISO/IEC 9899:1990）添加了新的国际化特征，一并称为多字节支持环境 (MSE)。修订 1 为具有状态的多字节代码集定义了附加的国际化 API，这同时也是为了更好地支持宽字符处理。

程序设计模型使这些多字节字符能够作为逻辑单元读入，然后在内部存储为宽字符。程序可将这些宽字符作为逻辑实体进行处理。最后，这些宽字符可在经过适当的转换后作为逻辑单元写出。

此过程类似于将单字节字符读入、进行处理再写出的方式。MSE 使人们能够使用与处理单字节字符相同的程序设计模型来处理多字节字符的程序。

动态链接应用程序

可以使用动态链接或静态链接，将应用程序与系统库（如 libc）链接。必须动态链接所有需要系统库中的国际化功能的应用程序。如果应用程序是被静态链接的，则在使用 `setlocale` 函数将语言环境设置为除 C 和 POSIX 以外的语言环境时，操作将失败。静态链接的应用程序只能在 C 和 POSIX 语言环境中运行。

缺省情况下，链接程序将尝试动态链接应用程序。如果链接程序和编译程序的命令行选项包括 `-Bstatic` 或 `-dn` 指定内容，则应用程序可能被静态链接。可以使用 `/usr/bin/ldd` 命令检查现有的应用程序是否为动态链接的。

例如，下列命令的响应表明 `/sbin/sh` 命令不是动态链接的程序：

```
% /usr/bin/ldd /sbin/sh
ldd: /sbin/sh: file is not a dynamic executable or shared object
```

下列命令的响应表明 `/usr/bin/ls` 命令已经与以下两个库动态链接：`libc.so.1` 和 `libdl.so.1`。

```
% /usr/bin/ldd /usr/bin/ls
libc.so.1 => /usr/lib/libc.so.1
libdl.so.1 => /usr/lib/libdl.so.1
```

更改的界面

`libw` 和 `libintl` 已移至 `libc`，不再位于 `libw` 和 `libintl` 中。

共享对象确保现有应用程序的运行兼容性，并且和归档文件一起为生成应用程序提供了编译环境兼容性。然而，您不必再针对 `libw` 或 `libintl` 生成应用程序。

以下列表说明了 `libw` 中的桩模块入口点：

<code>fgetwc</code>	<code>iswpunct</code>	<code>wscncat</code>	<code>wscoll</code>
<code>fgetws</code>	<code>iswspace</code>	<code>wcsncmp</code>	<code>wscopy</code>
<code>fputwc</code>	<code>iswupper</code>	<code>wcsncpy</code>	<code>wscspn</code>
<code>fputws</code>	<code>iswxdigit</code>	<code>wcsprbrk</code>	<code>wsdup</code>
<code>getwc</code>	<code>putwc</code>	<code>wcsrchr</code>	<code>wslen</code>
<code>getwchar</code>	<code>putwchar</code>	<code>wcsspn</code>	<code>wsncasecmp</code>
<code>getws</code>	<code>putws</code>	<code>wcstod</code>	<code>wsncat</code>
<code>isenglish</code>	<code>strtows</code>	<code>wcstok</code>	<code>wsncmp</code>
<code>isideogram</code>	<code>towlower</code>	<code>wcstol</code>	<code>wsncpy</code>
<code>isnumber</code>	<code>towupper</code>	<code>wcstoul</code>	<code>wspbrk</code>
<code>isphonogram</code>	<code>ungetwc</code>	<code>wcswcs</code>	<code>wsprintf</code>
<code>isspecial</code>	<code>watoll</code>	<code>wcswidth</code>	<code>wsrchr</code>
<code>iswalnum</code>	<code>wscat</code>	<code>wcsxfrm</code>	<code>wsscanf</code>
<code>iswalpha</code>	<code>wchr</code>	<code>wctype</code>	<code>wsspn</code>
<code>iswcntrl</code>	<code>wsclen</code>	<code>wcwidth</code>	<code>wstod</code>
<code>iswctype</code>	<code>wscmp</code>	<code>wscasecmp</code>	<code>wstok</code>
<code>iswdigit</code>	<code>wscoll</code>	<code>wscat</code>	<code>wstol</code>
<code>iswgraph</code>	<code>wscopy</code>	<code>wschr</code>	<code>wstoll</code>
<code>iswlower</code>	<code>wscspn</code>	<code>wscmp</code>	<code>wstostr</code>
<code>iswprint</code>	<code>wcsftime</code>	<code>wscol</code>	<code>wsxfrm</code>

以下列表说明了 `libintl` 中的桩模块入口点：

`bindtextdomain`

```
dcgettext
dgettext
gettext
textdomain
```

ctype 宏

字符分类和字符转换宏已在 `/usr/include/ctype.h` 中进行定义。当前 Solaris 环境提供一组 `ctype` 宏，该组宏支持由 XPG4 定义的字符分类和转换语义。要使所有 XPG4 和 XPG4.2 应用程序自动访问新的宏，必须满足下列条件之一：

- 定义了 `_XPG4_CHAR_CLASS`。
- 已经定义 `_XOPEN_SOURCE` 和 `_XOPEN_VERSION=4`。
- 已经定义 `_XOPEN_SOURCE` 和 `_XOPEN_SOURCE_EXTENDED=1`。

因为 `_XOPEN_SOURCE`、`_XOPEN_VERSION` 和 `_XOPEN_SOURCE_EXTENDED` 除了引入新的 `ctype` 宏以外，还引入了其他的 XPG4 相关功能，所以非 XPG4 或 XPG4.2 应用程序应该使用 `__XPG4_CHAR_CLASS__`。

还存在相应的 `ctype` 函数。当前 Solaris 环境函数也支持 XPG4 语义。

libc 中的国际化 API

当前 Solaris 环境提供两组 API：

- 多字节（文件代码）
- 宽字符（进程代码）

宽字符代码是固定宽度的逻辑实体单元。因此，在使用多字节字符时，不必跟踪维护正确的字符边界。

当程序从文件获取输入时，可以使用 `fscanf` 和 `fwscanf` 等输入函数直接将文件的多字节数据转换为宽字符进程代码，或者在输入后使用 `mbtowc` 和 `mbsrtowcs` 等转换函数进行转换。要将输出数据从宽字符格式转换为多字节字符格式，请使用 `fwprintf` 和 `fprintf` 等输出函数，或者在输出之前应用 `wctomb` 和 `wcsrtombs` 等转换函数。

本章剩余部分中的表格介绍了当前 Solaris 系统中包含的国际化 API。

下表介绍了 `libc` 中的消息传送函数 API。

表 2-1 libc 中的消息传送函数

库例程	说明
<code>bindtextdomain()</code>	绑定信息域的路径
<code>catclose()</code>	关闭信息目录
<code>catgets()</code>	读取程序信息
<code>catopen()</code>	打开信息目录
<code>dcgettext()</code>	指定了域和种类，从信息目录中获取信息
<code>dgettext()</code>	指定了域，从信息目录中获取信息
<code>gettext()</code>	从信息数据库中检索文本字符串
<code>textdomain()</code>	设置并查询当前域

下表介绍了 libc 中的代码转换函数 API。

表 2-2 libc 中的代码转换函数

库例程	说明
<code>iconv()</code>	转换代码
<code>iconv_close()</code>	解除分配转换描述符
<code>iconv_open()</code>	分配转换描述符

下表介绍了 libc 中的正则表达式 API。

表 2-3 libc 中的正则表达式

库例程	说明
<code>fnmatch()</code>	匹配文件名或路径名
<code>regcomp()</code>	编译正则表达式
<code>regerror()</code>	提供从错误代码到错误信息的映射
<code>regexec()</code>	执行正则表达式匹配
<code>regfree()</code>	释放由 <code>regcomp()</code> 分配的内存

下表介绍了 libc 中的宽字符函数 API。

表 2-4 libc 中的宽字符类

库例程	说明
wctrans()	定义字符映射
wctype()	定义字符类

下表列出 libc 中的修改和查询语言环境。

表 2-5 libc 中的修改和查询语言环境函数

库例程	说明
setlocale()	修改和查询程序的语言环境

下表列出 libc 中的查询语言环境数据。

表 2-6 libc 中的查询语言环境数据

库例程	说明
localeconv()	获取当前语言环境的货币和数字格式信息
nl_langinfo()	获取当前语言环境的语言和文化信息

下表介绍了 libc 中的字符分类函数 API。

表 2-7 libc 中的字符分类和拼写

库例程	说明
isalnum()	字符是字母或数字吗?
isalpha()	字符是字母吗?
isascii()	字符是 ASCII 字符吗?
iscntrl()	字符是控制字符吗?
isdigit()	字符是数字吗?
isenglish()	宽字符在辅助代码集的英语字母表中吗?
isgraph()	字符是可见字符吗?
isideogram()	宽字符是表意符号吗?
islower()	字符是小写吗?
isnumber()	宽字符是辅助代码集中的数字吗?
isphonogram()	宽字符是音标吗?

表 2-7 libc 中的字符分类和拼写 (续)

库例程	说明
isprint()	字符是可打印字符吗?
ispunct()	字符是标点符号吗?
isspace()	字符是空格吗?
isspecial()	特殊宽字符在辅助代码集中吗?
isupper()	字符是大写吗?
iswalnum()	宽字符是字母字符或数字吗?
iswalpha()	宽字符是字母吗?
iswascii()	宽字符是 ASCII 字符吗?
iswcntrl()	宽字符是控制字符吗?
iswdigit()	宽字符是数字吗?
iswgraph()	宽字符是可见字符吗?
iswlower()	宽字符是小写吗?
iswprint()	宽字符是可打印字符吗?
iswpunct()	宽字符是标点符号吗?
iswspace()	宽字符是空白吗?
iswupper()	宽字符是大写吗?
iswxdigit()	宽字符是十六进制数字吗?
isxdigit()	字符是十六进制数字吗?
tolower()	将大写字符转换为小写字符。
toupper()	将小写字符转换为大写字符。
towctrans()	宽字符映射。
towlower()	将大写宽字符转换为小写宽字符。
towupper()	将小写宽字符转换为大写宽字符。

下表介绍了 libc 中的字符排序函数 API。

表 2-8 libc 中的字符排序

库例程	说明
strcoll()	排序字符串

表 2-8 libc 中的字符排序 (续)

库例程	说明
strxfrm()	变换字符串以便进行比较
wscoll()	排序宽字符串
wcsxfrm()	变换宽字符串以便进行比较

下表介绍了 libc 中的货币处理函数 API。

表 2-9 libc 中的货币格式

库例程	说明
localeconv()	获取当前语言环境的货币格式信息
strfmon()	将货币值转换为字符串表示形式

下表介绍了 libc 中的日期和时间格式。

表 2-10 libc 中的日期和时间格式

库例程	说明
getdate()	转换用户格式日期和时间。
strftime()	将日期和时间转换为字符串表示形式。%u 转换函数符合“系统界面和标题”的第 4 期，版本 2 中的 X/Open CAE 规范。该函数使用十进制数字 [1,7] 表示一周中的每一天，现在，1 表示星期一。
strptime()	日期和时间转换。

下表介绍了 libc 中的多字节处理函数 API。

表 2-11 libc 中的多字节处理

库例程	描述
btowc()	单字节到宽字符转换
mblen()	获取字符中的字节数
mbrlen()	获取字符中的字节数 (可重新开始)
mbrtowc()	将字符转换为宽字符代码 (可重新开始)
mbsinit()	确定转换对象状态
mbsrtowcs()	将字符串转换为宽字符串 (可重新开始)

表 2-11 libc 中的多字节处理 (续)

库例程	描述
mbstowcs()	将字符串转换为宽字符串
mbtowc()	将字符转换为宽字符代码

下表介绍了 libc 中的宽字符和字符串处理。

表 2-12 libc 中的宽字符和字符串处理

库例程	描述
wcrtomb()	将宽字符代码转换为字符 (可重新开始)
wcscat()	并置宽字符串
wcschr()	在宽字符串中查找字符
wcscmp()	比较宽字符串
wcscpy()	复制宽字符串
wcscspn()	返回一个宽字符串不在另一个宽字符串中的跨度
wcslen()	获取宽字符串的长度
wcsncat()	并置宽字符串, 并置长度为 n
wcsncmp()	比较宽字符串, 比较长度为 n
wcsncpy()	复制宽字符串, 复制长度为 n
wcspbrk()	返回指向一个位于另一个宽字符串中的宽字符串的指针
wcsrchr()	从右边开始在宽字符串中查找字符
wcsrtombs()	将宽字符串转换为字符串 (可重新开始)
wcsspn()	返回一个宽字符串在另一个宽字符串中的跨度
wcstod()	将宽字符串转换为双精度
wcstok()	在整个宽字符串中移动标记
wcstol()	将宽字符串转换为长整数
wcstombs()	将宽字符串转换为多字节字符串
wcstoul()	将宽字符串转换为无符号长整数
wscwcs()	在宽字符串中查找字符串
wcswidth()	确定宽字符串的列位置数
wctob()	宽字符到单字节转换

表 2-12 libc 中的宽字符和字符串处理 (续)

库例程	描述
wctomb()	将宽字符转换为多字节字符
wcwidth()	确定宽字符的列位置数
wscol()	返回宽字符串的显示宽度
wsdup()	复制宽字符串

下表介绍了 libc 中的格式化宽字符输入和输出。

表 2-13 libc 中的格式化宽字符输入和输出

库例程	描述
fwprintf()	打印格式化宽字符输出
fwscanf()	转换格式化宽字符输入
swprintf()	打印格式化宽字符输出
swscanf()	转换格式化宽字符输入
vfwprintf()	stdarg 参数列表的宽字符格式化输出
vswprintf()	stdarg 参数列表的宽字符格式化输出
wprintf()	打印格式化宽字符输出
wscanf()	转换格式化宽字符输入
wsprintf()	根据格式生成宽字符串
wsscanf()	格式化输入转换

下表介绍 libc 中的宽字符串函数 API。

表 2-14 宽字符串 libc

库例程	描述
wcsstr()	查找宽字符子串
wmemchr()	在内存中查找宽字符
wmemcmp()	在内存中比较宽字符
wmemcpy()	在内存中复制宽字符
wmemmove()	在具有重叠区域的内存中复制宽字符
wmemset()	在内存中设置宽字符

表 2-14 宽字符串 libc (续)

库例程	描述
wscasecmp()	比较宽字符串，忽略大小写差异
wsncasecmp()	进程代码串操作

下表介绍了 libc 中的宽字符输入和输出。

表 2-15 libc 中的宽字符输入和输出

库例程	说明
fgetwc()	从流中获取多字节字符，并转换为宽字符
fgetws()	从流中获取多字节串，并转换为宽字符
fputwc()	将宽字符转换为多字节字符，并放入流中
fputws()	将宽字符转换为多字节串，并放入流中
fwide()	设置流定向
getwchar()	从 stdin 中获取多字节字符，并转换为宽字符
getws()	从 stdin 中获取多字节串，并转换为宽字符
putwchar()	将宽字符转换为多字节字符，并放入 stdin 中
putws()	将宽字符转换为多字节串，并放入 stdin 中
ungetwc()	将宽字符推送回输入流中

genmsg 实用程序

新的 genmsg 实用程序可以与 catgets() 系列函数一起使用，以创建国际化的源信息目录。该实用程序在源程序文件中检查对 catgets 中函数的调用，然后根据找到的信息生成源信息目录。例如：

```
% cat example.c
...
/* NOTE: %s is a file name */
printf(catgets(catd, 5, 1, "%s cannot be opened.));
/* NOTE: "Read" is a past participle, not a present
    tense verb */
printf(catgets(catd, 5, 1, "Read"));
...
% genmsg -c NOTE example.c
The following file(s) have been created.
```

```
new msg file = "example.c.msg"
% cat example.c.msg
$quote "
$set 5
1          "%s cannot be opened"
  /* NOTE: %s is a file name */
2          "Read"
  /* NOTE: "Read" is a past participle, not a present
         tense verb */
```

在上面的实例中，对源文件 `example.c` 运行 `genmsg`，产生一个名为 `example.c.msg` 的源信息目录。带 `NOTE` 参数的 `-c` 选项使 `genmsg` 在目录中包含注释。如果源程序中的注释包含指定的字符串，则该注释在信息目录中出现在下一个从对 `catgets` 的调用中提取的字符串的后面。

可以使用 `genmsg` 对信息集中的信息进行自动编号。

有关更多信息，请参见 `genmsg(1)` 手册页。

要生成格式化信息目录文件，请使用 `gencat (1)` 实用程序。

有关可移植消息文件（`.po` 文件）的消息提取实用程序的信息，以及如何从 `.po` 文件生成消息对象文件（`.mo` 文件）的信息。

用户定义的和用户可扩展的代码转换

您可以使用 `geniconvtbl` 实用程序来创建用户定义的代码集转换器。

使用该实用程序，可以使用标准系统实用程序和接口（如 `iconv(1)` 和 `iconv(3C)`），完成用户定义的和用户可定制代码集的转换。该功能增强了应用程序处理不兼容的数据类型特别是从专用或传统应用程序生成的数据的能力。同时也支持对现有的 Solaris 代码集转换的修改。

`/usr/lib/iconv/geniconvtbl/srcs/` 目录中提供该实用程序的输入源文件实例。

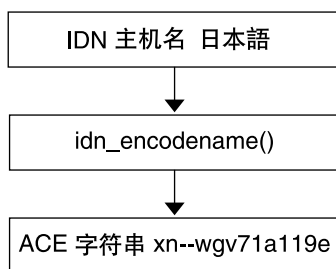
正确准备和放置用户定义的代码转换后，用户可以使用 32 位和 34 位 Solaris 操作系统的 `iconv(1)` 实用程序和 `iconv(3C)` 函数进行代码转换。

国际化域名 (IDN) 支持

利用国际化域名 (IDN) 功能，可以使用非英语的本机语言名称作为主机名和域名。要使用非英语的主机名和域名，根据 RFC 3490 的规定，请在将名称发送至解析器路由之前将这些名称转换为使用 ASCII 兼容编码 (ACE) 编码的名称。在系统管理应用程序不支持 IDN 的系统文件和应用程序中，系统管理员仍然需要使用 ACE 名称。

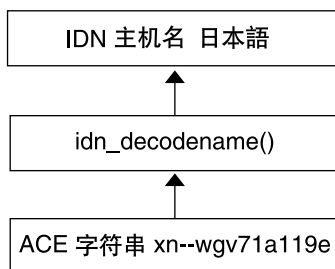
请参见 RFC 3490“应用程序中的国际化域名 (IDNA)”。

libidnkit(3EXT) 中的国际化域名 API 提供 UTF-8 或应用程序语言环境的字符集和 ACE 的方便转换。如果使用了 `idn_decodename2(3EXT)`，您还可以指定任意代码集名称作为输入参数的代码集。



用 ACE 字符串做为解析器例程 (例如 `getaddrinfo(3SOCKET)`) 的输入

图 2-1 IDN 至 ACE 转换



ACE 字符串从解析器例程 (如 `getnameinfo(3SOCKET)`) 返回

图 2-2 ACE 至 IDN 转换

下表显示您可以使用的双向 `iconv` 代码转换。

表 2-16 iconv 代码转换

源代码	目标代码
ACE	UTF-8
ACE-ALLOW-UNASSIGNED	UTF-8
UTF-8	ACE
UTF-8	ACE-ALLOW-UNASSIGNED

ACE 和 ACE-ALLOW-UNASSIGNED iconv 代码转换名称具有下列意义：

- ACE。
ACE 是可在 iconv 代码转换中使用的 fromcode 或 tocode 名称，指 RFC 3490 中定义的 ASCII 兼容编码。该转换使用 STD3 ASCII 规则。不允许使用未分配字符。ACE 通常用于将主机名或域名存储在或传递到计算机上。
- ACE-ALLOW-UNASSIGNED。
ACE-ALLOW-UNASSIGNED 与 ACE 执行相同操作，只是 ACE-ALLOW-UNASSIGNED 允许使用未分配的字符。ACE-ALLOW-UNASSIGNED 通常用于查询。

下例显示将输入的 `hostnames.txt` 文件从 ACE 转换为至 UTF-8。输出将变为标准输出。

```
system% iconv -f ACE -t UTF-8 hostnames.txt
```

专用 IDN 转换实用程序 `idnconv(1)` 提供具有不同选项的 IDN 转换。这些选项控制转换细节。

有关 IDN、转换例程和 iconv 代码转换的信息，请参见 `libidnkit(3LIB)`、`idn_decodename(3EXT)`、`idn_decodename2(3EXT)`、`idn_encodename(3EXT)` 和 `iconv_en_US.UTF-8(5)` 手册页。

第 3 章

Solaris 环境中的本地化

本章讨论当前 Solaris 环境中的本地化功能。本章包含以下主题：

- 第 51 页 “本地化的软件支持”
- 第 52 页 “支持的语言环境”
- 第 58 页 “语言环境的多键书写序列”
- 第 59 页 “Solaris 环境中的键盘支持”
- 第 71 页 “新增 Solaris 键盘软件支持”

本地化的软件支持

本节包含有关 Solaris 语言环境软件包、CD-ROM 光盘、本地化函数和脚本支持的信息。

Solaris 语言环境软件包摘要

所有当前 Solaris 语言环境软件包分为完全语言环境或部分语言环境两类。

部分语言环境是语言环境的启用程序。在系统上安装部分语言环境后，用户可以在目标语言环境中输入、显示、打印文本以及运行应用程序，而 Solaris 操作系统中的 OS/GUI 消息均为英语。所有部分语言环境软件包都可以从 Solaris Software CD 中获得。日语和亚洲部分语言环境根据语言进行封装。部分语言环境根据地区进行封装。

完全语言环境软件包包括软件消息的翻译、联机帮助文件、可选字体和特定于语言的功能。完全语言环境软件包为许多语言提供了全套语言功能。所有基于下列语言的语言环境均是完全语言环境：

- 德语
- 法语

- 西班牙语
- 瑞典语
- 意大利语
- 日语
- 朝鲜语
- 简体中文
- 繁体中文

完全语言环境软件包根据语言进行封装，并可以在 Language CD 中得到。

注意 – 为了使完全语言环境起作用，必须安装部分语言环境软件包（语言环境启用程序）。

在 Solaris 安装过程中，会提示您选择需要支持的地区。在安装完成后可用的语言环境支持取决于在该阶段所做的选择。部分语言环境从包含 Solaris 操作系统的 Solaris Software CD-ROM 安装，而完全语言环境则从 Language CD 安装。如果不需要完全语言环境支持，则可以在安装过程中跳过从 Languages CD 进行的安装。请注意，所安装的缺省语言环境为英语语言环境。

支持的语言环境

下表列出 Solaris 环境中支持的所有语言环境。语言环境名称符合国际命名标准。

表 3-1 亚洲语言环境

语言环境	用户界面	地区	代码集	语言支持
hi_IN.UTF-8	英语	印度	UTF-8	北印度语文（语）(UTF-8) Unicode 4.0
ja	日语	日本	eucJP ¹	日语 (EUC) JIS X 0201-1976 JIS X 0208-1990 JIS X 0212-1990
ja_JP.eucJP	日语	日本	eucJP	日语 (EUC) JIS X 0201-1976 JIS X 0208-1990

¹ eucJP 表示日语 EUC 代码集。ja_JP.eucJP 语言环境规范符合 UI_OSF 日语环境执行协议版本 1.1，并且 ja 语言环境符合早期 Solaris 发行版中的传统规范。

表 3-1 亚洲语言环境 (续)

语言环境	用户界面	地区	代码集	语言支持
				JIS X 0212-1990
ja_JP.PCK	日语	日本	PCK ²	日语 (PC Kanji)
				JIS X 0201-1976
				JIS X 0208-1990
ja_JP.UTF-8	日语	日本	UTF-8	日语 (UTF-8) Unicode 4.0
ko_KR.EUC	朝鲜语	韩国	1001	朝鲜语 (EUC) KS X 1001
ko_KR.UTF-8	朝鲜语	韩国	UTF-8	朝鲜语 (UTF-8) Unicode 4.0
th_TH.UTF-8	英语	泰国	UTF-8	泰文 (UTF-8) Unicode 4.0
th_TH.TIS620	英语	泰国	TIS620.2533	泰文 TIS620.2533
zh_CN.EUC	简体中文	中华人民共和国	gb2312 ³	简体中文 (EUC) GB2312-1980
zh_CN.GBK	简体中文	中华人民共和国	GBK ⁴	简体中文 (GBK)
zh_CN.GB18030	简体中文	中华人民共和国	GB18030-2000	简体中文 (GB18030-2000) GB18030-2000
zh_CN.UTF-8	简体中文	中华人民共和国	UTF-8	简体中文 (UTF-8) Unicode 4.0
zh_HK.BIG5HK	繁体中文	香港	Big5+HKSCS	繁体中文 (BIG5+HKSCS)
zh_HK.UTF-8	繁体中文	香港	UTF-8	繁体中文 (UTF-8) Unicode 4.0
zh_TW.EUC	繁体中文	中国台湾地区	cns11643	繁体中文 (EUC) CNS 11643-1992
zh_TW.BIG5	繁体中文	中国台湾地区	BIG5	繁体中文 (BIG5)
zh_TW.UTF-8	繁体中文	台湾	UTF-8	繁体中文 (UTF-8) Unicode 4.0

² PCK 也称作 Shift_JIS (SJIS)。

³ gb2312 表示简体中文 EUC 代码集，它包含 GB 1988-80 和 GB 2312-80。

⁴ GBK 表示 GB 扩展。这些扩展包含所有 GB 2312-80 字符和 ISO/IEC 10646-1 的所有统一汉字字符以及日语平假名和片假名字符。GBK 还包含许多中文、日语、朝鲜语字符集字符以及 ISO/IEC 10646-1 字符。

表 3-2 澳大拉西亚语言环境

语言环境	用户界面	地区	代码集	语言支持
en_AU.ISO8859-1	英语	澳大利亚	ISO8859-1	英语 (澳大利亚)
en_NZ.ISO8859-1	英语	新西兰	ISO8859-1	英语 (新西兰)

表 3-3 中美洲语言环境

语言环境	用户界面	地区	代码集	语言支持
es_CR.ISO8859-1	西班牙文	哥斯达黎加	ISO8859-1	西班牙文（哥斯达黎加）
es_GT.ISO8859-1	西班牙文	危地马拉	ISO8859-1	西班牙文（危地马拉）
es_NI.ISO8859-1	西班牙文	尼加拉瓜	ISO8859-1	西班牙文（尼加拉瓜）
es_PA.ISO8859-1	西班牙文	巴拿马	ISO8859-1	西班牙文（巴拿马）
es_SV.ISO8859-1	西班牙文	萨尔瓦多	ISO8859-1	西班牙文（萨尔瓦多）

表 3-4 中欧语言环境

语言环境	用户界面	地区	代码集	语言支持
cs_CZ.ISO8859-2	英语	捷克共和国	ISO8859-2	捷克文（捷克共和国）
de_AT.ISO8859-1	德文	奥地利	ISO8859-1	德文（奥地利）
de_AT.ISO8859-15	德文	奥地利	ISO8859-15	德文（奥地利、ISO8859-15 - 欧洲）
de_CH.ISO8859-1	德文	瑞士	ISO8859-1	德文（瑞士）
de_DE.UTF-8	德文	德国	UTF-8	德文（德国，Unicode 4.0）
de_DE.ISO8859-1	德文	德国	ISO8859-1	德文（德国）
de_DE.ISO8859-15	德文	德国	ISO8859-15	德文（德国、ISO8859-15 - 欧洲）
fr_CH.ISO8859-1	法文	瑞士	ISO8859-1	法文（瑞士）
hu_HU.ISO8859-2	英语	匈牙利	ISO8859-2	匈牙利文（匈牙利）
pl_PL.ISO8859-2	英语	波兰	ISO8859-2	波兰文（波兰）
pl_PL.UTF-8	英语	波兰	UTF-8	波兰文（波兰，Unicode 4.0）
sk_SK.ISO8859-2	英语	斯洛伐克	ISO8859-2	斯洛伐克文（斯洛伐克）

表 3-5 东欧语言环境

语言环境	用户界面	地区	代码集	语言支持
bg_BG.ISO8859-5	英语	保加利亚	ISO8859-5	保加利亚文（保加利亚）
et_EE.ISO8859-15	英语	爱沙尼亚	ISO8859-15	爱沙尼亚文（爱沙尼亚）
hr_HR.ISO8859-2	英语	克罗地亚	ISO8859-2	克罗地亚文（克罗地亚）

表 3-5 东欧语言环境 (续)

语言环境	用户界面	地区	代码集	语言支持
lt_LT.ISO8859-13	英语	立陶宛	ISO8859-13	立陶宛文 (立陶宛)
lv_LV.ISO8859-13	英语	拉脱维亚	ISO8859-13	拉脱维亚文 (拉脱维亚)
mk_MK.ISO8859-5	英语	马其顿	ISO8859-5	马其顿文 (马其顿)
ro_RO.ISO8859-2	英语	罗马尼亚	ISO8859-2	罗马尼亚文 (罗马尼亚)
ru_RU.KOI8-R	英语	俄罗斯	KOI8-R	俄文 (俄罗斯、KOI8-R)
ru_RU.ANSI1251	英语	俄罗斯	ansi-1251	俄文 (俄罗斯、ANSI 1251)
ru_RU.ISO8859-5	英语	俄罗斯	ISO8859-5	俄文 (俄罗斯)
ru_RU.UTF-8	英语	俄罗斯	UTF-8	俄文 (俄罗斯、Unicode 4.0)
sh_BA.ISO8859-2@bosnia	英语	波斯尼亚	ISO8859-2	波斯尼亚文 (波斯尼亚)
sl_SI.ISO8859-2	英语	斯洛文尼亚	ISO8859-2	斯洛文尼亚文 (斯洛文尼亚)
sq_AL.ISO8859-2	英语	阿尔巴尼亚	ISO8859-2	阿尔巴尼亚文 (阿尔巴尼亚)
sr_YU.ISO8859-5	英语	塞尔维亚	ISO8859-5	塞尔维亚文 (塞尔维亚)
tr_TR.ISO8859-9	英语	土耳其	ISO8859-9	土耳其文 (土耳其)
tr_TR.UTF-8	英语	土耳其	UTF-8	土耳其文 (土耳其、Unicode 4.0)

表 3-6 中东语言环境

语言环境	用户界面	地区	代码集	语言支持
He	英语	以色列	ISO8859-8	希伯来文 (以色列)

表 3-7 北非语言环境

语言环境	用户界面	地区	代码集	语言支持
ar_EG.UTF-8	英语	埃及	UTF-8	阿拉伯文（埃及）
Ar	英语	埃及	ISO8859-6	阿拉伯文（埃及）

表 3-8 北美语言环境

语言环境	用户界面	地区	代码集	语言支持
en_CA.ISO8859-1	英语	加拿大	ISO8859-1	英语（加拿大）
en_US.ISO8859-1	英语	美国	ISO8859-1	英语（美国）
en_US.ISO8859-15	英语	美国	ISO8859-15	英语（美国、ISO8859-15 - 欧洲）
en_US.UTF-8	英语	美国	UTF-8	英语（美国，Unicode 4.0）
fr_CA.ISO8859-1	法文	加拿大	ISO8859-1	法语（加拿大）
es_MX.ISO8859-1	西班牙文	墨西哥	ISO8859-1	西班牙文（墨西哥）

表 3-9 北欧语言环境

语言环境	用户界面	地区	代码集	语言支持
da_DK.ISO8859-1	英语	丹麦	ISO8859-1	丹麦文（丹麦）
da_DK.ISO8859-15	英语	丹麦	ISO8859-15	丹麦文（丹麦、ISO8859-15 - 欧洲）
fi_FI.ISO8859-1	英语	芬兰	ISO8859-1	芬兰文，Unicode 4.0
fi_FI.ISO8859-15	英语	芬兰	ISO8859-15	芬兰文（芬兰、ISO8859-15 - 欧洲）
fi_FI.UTF-8	英语	芬兰	UTF-8	芬兰文（芬兰）
is_IS.ISO8859-1	英语	冰岛	ISO8859-1	冰岛文（冰岛）
no_NO.ISO8859-1@bokmal	英语	挪威	ISO8859-1	挪威文（挪威 - 博克马尔）
no_NO.ISO8859-1@nyorsk	英语	挪威	ISO8859-1	挪威文（挪威 - 尼诺斯克）
sv_SE.ISO8859-1	瑞典语	瑞典	ISO8859-1	瑞典语（瑞典）
sv_SE.ISO8859-15	瑞典语	瑞典	ISO8859-15	瑞典语（瑞典、ISO8859-15 - 欧洲）
sv_SE.UTF-8	瑞典语	瑞典	UTF-8	瑞典语（瑞典，Unicode 4.0）

表 3-10 南美语言环境

语言环境	用户界面	地区	代码集	语言支持
es_AR.ISO8859-1	西班牙文	阿根廷	ISO8859-1	西班牙文 (阿根廷)
es_BO.ISO8859-1	西班牙文	玻利维亚	ISO8859-1	西班牙文 (玻利维亚)
es_CL.ISO8859-1	西班牙文	智利	ISO8859-1	西班牙文 (智利)
es_CO.ISO8859-1	西班牙文	哥伦比亚	ISO8859-1	西班牙文 (哥伦比亚)
es_EC.ISO8859-1	西班牙文	厄瓜多尔	ISO8859-1	西班牙文 (厄瓜多尔)
es_PE.ISO8859-1	西班牙文	秘鲁	ISO8859-1	西班牙文 (秘鲁)
es_PY.ISO8859-1	西班牙文	巴拉圭	ISO8859-1	西班牙文 (巴拉圭)
es_UY.ISO8859-1	西班牙文	乌拉圭	ISO8859-1	西班牙文 (乌拉圭)
es_VE.ISO8859-1	西班牙文	委内瑞拉	ISO8859-1	西班牙文 (委内瑞拉)
pt_BR.ISO8859-1	英语	巴西	ISO8859-1	葡萄牙文 (巴西)
pt_BR.UTF-8	英语	巴西	UTF-8	葡萄牙文 (巴西, Unicode 4.0)

表 3-11 南欧语言环境

语言环境	用户界面	地区	代码集	语言支持
ca_ES.ISO8859-1	英语	西班牙	ISO8859-1	加泰罗尼亚文 (西班牙)
ca_ES.ISO8859-15	英语	西班牙	ISO8859-15	加泰罗尼亚文 (西班牙、ISO8859-15 - 欧洲)
el_GR.ISO8859-7	英文	希腊	ISO8859-7	希腊文 (希腊)
es_ES.ISO8859-1	西班牙文	西班牙	ISO8859-1	西班牙文 (西班牙)
es_ES.ISO8859-15	西班牙文	西班牙	ISO8859-15	西班牙文 (西班牙、ISO8859-15 - 欧洲)
es_ES.UTF-8	西班牙文	西班牙	UTF-8	西班牙文 (西班牙, Unicode 4.0)
it_IT.ISO8859-1	意大利语	意大利	ISO8859-1	意大利语 (意大利)
it_IT.ISO8859-15	意大利语	意大利	ISO8859-15	意大利语 (意大利、ISO8859-15 - 欧洲)
it_IT.UTF-8	意大利语	意大利	UTF-8	意大利语 (意大利, Unicode 4.0)
pt_PT.ISO8859-1	英文	葡萄牙	ISO8859-1	葡萄牙文 (葡萄牙)

表 3-11 南欧语言环境 (续)

语言环境	用户界面	地区	代码集	语言支持
pt_PT.ISO8859-15	英文	葡萄牙	ISO8859-15	葡萄牙文(葡萄牙、ISO8859-15 - 欧洲)

表 3-12 西欧语言环境

语言环境	用户界面	地区	代码集	语言支持
en_GB.ISO8859-1	英文	大不列颠	ISO8859-1	英文(大不列颠)
en_IE.ISO8859-1	英文	爱尔兰	ISO8859-1	英文(爱尔兰)
fr_BE.ISO8859-1	法语	比利时 - 瓦龙	ISO8859-1	法语(比利时 - 瓦龙, Unicode 4.0)
fr_BE.UTF-8	法语	比利时 - 瓦龙	UTF-8	法语(比利时 - 瓦龙, Unicode 4.0)
fr_FR.ISO8859-1	法语	法国	ISO8859-1	法语(法国)
fr_FR.UTF-8	法语	法国	UTF-8	法语(法国, Unicode 4.0)
nl_BE.ISO8859-1	英文	比利时 - 佛兰德	ISO8859-1	荷兰文(比利时 - 佛兰德)
nl_NL.ISO8859-1	英文	荷兰	ISO8859-1	荷兰文(荷兰)

语言环境的多键书写序列

许多 Solaris 语言环境，特别是欧洲和 Unicode 语言环境允许使用所谓的“死键序列”（也称为书写键序列）输入各种字符。

书写键序列用于输入带有变音标记的字符和其他在键盘键帽上未标示出来的字符。

下表显示了几个书写键序列的实例。有关书写键序列的更完整信息，请参见第 112 页“[英语/欧洲语言输入模式](#)”。

表 3-13 使用书写键创建的发音字符

标记	书写键组合	实例
分音符		书写 A " -> 带分音符的 A
抑扬符	v	书写 Z v -> 带抑扬符的 Z
短音符	u	书写 G u -> 带短音符的 G

表 3-13 使用书写键创建的发音字符 (续)

标记	书写键组合	实例
Ogonek	a	书写 A a -> 带 Ogonek 的 A
变音符	,	书写 K , -> 带变音符的 K
注册标记	R O	书写 R O -> 注册标记
倒置叹号	!!	书写 !! -> 倒置叹号

注意 - 如果一个字符不是当前语言环境中的代码集的一部分，则书写键序列不能生成该字符。例如，由于带抑扬符的 Z 属于 ISO8859-1 代码集，所以不能在 en_US.ISO8859-1 语言环境中输入带抑扬符的 Z。

Solaris 环境中的键盘支持

为 SPARC 和 Intel Architecture (IA) 平台支持用于特定地区的不同布局的键盘。Solaris 操作系统支持下表中列出的地区键盘。

表 3-14 对地区键盘的支持

地区	国家 (地区)	Sun 键盘 (4/5/5c 型)	Sun 键盘 (6 型)	PC 键盘
亚洲	日本	X	X	X
	韩国	X	X	X
	台湾	X	X	X
欧洲	比利时	X	X	X
	捷克共和国	X		X
	丹麦	X	X	X
	芬兰		X	
	法国	X	X	X
	德国	X	X	X
	大不列颠	X	X	X
	希腊	X		X
	匈牙利	X		X

表 3-14 对地区键盘的支持 (续)

地区	国家 (地区)	Sun 键盘 (4/5/5c 型)	Sun 键盘 (6 型)	PC 键盘
	意大利	X	X	X
	拉脱维亚	X		X
	立陶宛	X		X
	荷兰	X	X	X
	挪威	X	X	X
	波兰	X		X
	葡萄牙	X	X	X
	俄罗斯	X	X	X
	西班牙	X	X	X
	瑞典	X	X	X
	瑞士 (法语)	X	X	X
	瑞士 (德语)	X	X	X
	土耳其	X	X	X
美洲	加拿大 (法语)	X	X	X
	拉丁美洲 (西班牙语)	X		
	美国	X	X	X
中东	阿拉伯语	X	X	

对于键盘布局符合国际标准的地区 (如中国), 则使用为美国提供的键盘布局支持输入该语言环境的字符。基础键盘映射是相同的。某些国家 (如日本、土耳其和瑞士) 具有多种键盘, 原因是使用多种语言或者存在多种键盘布局。

Sun 4、5 和 5c 型键盘通过 Mini DIN 8 针连接使用 Sun I/O 接口。Sun 6 型键盘具有两种版本的接口:

- 通过 Mini DIN 8 针连接 Sun I/O 接口
- USB

Sun 键盘类型印在每个 Sun 键盘的背面。

PC 键盘使用各种不同的接口, 如 PS/2 或 USB。

在 SPARC 系统上更改键盘

您可以在 Solaris 系统上使用 DIP 切换设置，更改大多数 Sun 4 型、5 型和 5c 型键盘的布局。可用于 DIP 切换设置的键盘类型、名称和相应的布局 ID 的列表位于 `/usr/openwin/share/etc/keytables/keytable.map` 文件中。

注意 – 不能更改 6 型键盘的布局，因为键盘后面没有 DIP 开关。某些 5 型和 5c 型键盘（如美国键盘、美国/UNIX 键盘和日语键盘）具有跳线，而没有 DIP 开关。除 `xmodmap(1)` 等实用程序外，SPARC 平台或 IA 平台不提供用于切换键盘的实用程序或工具。

下表列出 4 型、5 型和 5c 型键盘的布局 ID 值（1 = 切换开，0 = 切换关）。

表 3-15 4、5 和 5c 型键盘的布局

DIP 开关	键盘 (Keytable 文件)	二进制设置
0	美国 (US4.kt)	000000
1	美国 (US4.kt)	000001
2	比利时 (FranceBelg4.kt)	000010
3	加拿大 (Canada4.kt)	000011
4	丹麦 (Denmark4.kt)	000100
5	德国 (Germany4.kt)	000101
6	意大利 (Italy4.kt)	000110
7	荷兰 (Netherland4.kt)	000111
8	挪威 (Norway4.kt)	001000
9	葡萄牙 (Portugal4.kt)	001001
10 (0x0a)	拉丁美洲/西班牙语 (SpainLatAm4.kt)	001010
11 (0x0b)	瑞典 (SwedenFin4.kt)	001011
12 (0x0c)	瑞士/法语 (Switzer_Fr4.kt)	001100
13 (0x0d)	瑞士/德语 (Switzer_Ge4.kt)	001101
14 (0x0e)	大不列颠 (UK4.kt)	001110
16 (0x10)	韩国 (Korea4.kt)	010000
17 (0x11)	中国台湾地区 (Taiwan4.kt)	010001
23	俄语	100001

表 3-15 4、5 和 5c 型键盘的布局 (续)

DIP 开关	键盘 (Keytable 文件)	二进制设置
33 (0x21)	美国 (US5.kt)	100111
34 (0x22)	美国/UNIX (US_UNIX5.kt)	100010
35 (0x23)	法国 (France5.kt)	100011
36 (0x24)	丹麦 (Denmark5.kt)	100100
37 (0x25)	德国 (Germany5.kt)	100101
38 (0x26)	意大利 (Italy5.kt)	100110
39 (0x27)	荷兰 (Netherland5.kt)	100111
40 (0x28)	挪威 (Norway5.kt)	101000
41 (0x29)	葡萄牙 (Portugal5.kt)	101001
42 (0x2a)	西班牙 (Spain5.kt)	101010
43 (0x2b)	瑞典 (Sweden5.kt)	101011
44 (0x2c)	瑞士/法语 (Switzer_Fr5.kt)	101101
45 (0x2d)	瑞士/德语 (Switzer_Ge5.kt)	101110
46 (0x2e)	大不列颠 (UK5.kt)	101111
47 (0x2f)	韩国 (Korea5.kt)	101111
48 (0x30)	中国台湾地区 (Taiwan5.kt)	110000
49 (0x31)	日本 (Japan5.kt)	110001
50 (0x32), 参见 63 (0x3f)	加拿大/法语 (Canada_Fr5.kt)	110010
51 (0x33)	匈牙利 (Hungary5.kt)	110011
52 (0x34)	波兰 (Poland5.kt)	110100
53 (0x35)	捷克语 (Czech5.kt)	110101
54 (0x36)	俄罗斯 (Russia5.kt)	110110
55 (0x37)	拉脱维亚 (Latvia5.kt)	110111
56 (0x38), 参见 62 (0x3e)	土耳其-Q5 (TurkeyQ5.kt)	111000
57 (0x39)	希腊 (Greece5.kt)	111001
58 (0x3a)	阿拉伯语 (Arabic5.kt)	111011
59 (0x3b)	立陶宛 (Lithuania5.kt)	111010

表 3-15 4、5 和 5c 型键盘的布局 (续)

DIP 开关	键盘 (Keytable 文件)	二进制设置
60 (0x3c)	比利时 (Belgian5.kt)	111100
62 (0x3e)	加拿大/法语 (Canada_Fr5_TBITS5.kt)	111111
	加拿大法语	
	波兰语程序员	
	爱沙尼亚	

带有 4 的 Keytable 文件名用于 4 型键盘。带有 5 的 Keytable 文件名用于 5 型键盘。

▼ 如何将键盘布局更改为捷克语布局

1. 从该表或从 `/usr/openwin/share/etc/keytables/keytable.mp` 文件中确定正确的 DIP 开关 ID (或布局 ID)。keytable.mp 文件中的布局 ID 值是一个十进制值。
对于捷克语，布局 ID 为十进制的 53 (即十六进制的 0x35)。
2. 将布局 ID 转换为二进制，或者使用上表中正确的二进制设置值。对于基转换，可能会用到计算器公用程序，如 `dtcalc(1)`。
例如，捷克语键盘的正确二进制值是 110101。
3. 关机并切断系统电源。
4. 使用第 2 步中的二进制值来更改键盘背面的 DIP 切换设置。
第一个 DIP 开关位于您的左侧。对于 1，将开关向上切换；对于 0，将开关向下切换。
捷克语键盘的二进制值 110101 对应于：上上下下上上
5. 打开电源并引导系统以备使用。

注意 - 与 4 型键盘不同，5 型和 5c 型键盘只有五个 DIP 开关。对于 5 型和 5c 型键盘，请忽略第一个二进制数字。例如，对于捷克语 5c 型键盘，正确的 DIP 切换设置仅使用最后五位数字 10101，即为“上下上下上”。

在 Intel 系统上更改键盘

在 Intel 体系结构系统上，在安装的 `kdmconfig(1M)` 过程中选择键盘。要在安装后更改该设置，请退出 GUI 桌面环境并转到命令行模式下。作为超级用户，请键入 `kdmconfig` 来运行程序。遵循说明来获取所需的键盘布局。

键盘布局图解

下图显示阿拉伯语键盘。

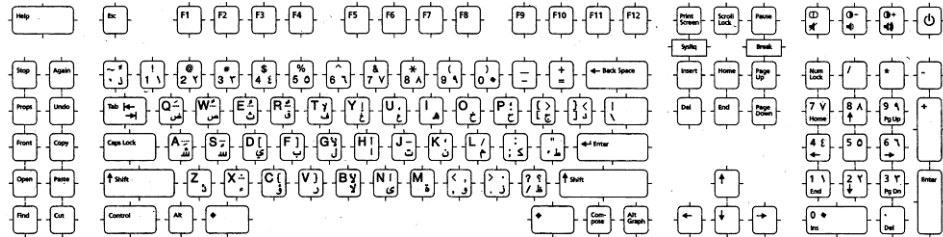


图 3-1 阿拉伯语键盘

下图显示了比利时语键盘。

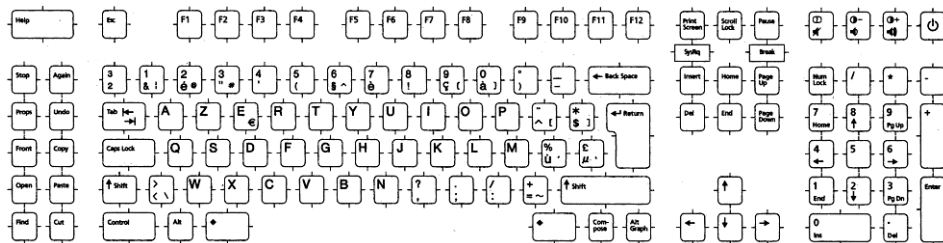


图 3-2 比利时语键盘

下图显示了西里尔语键盘。

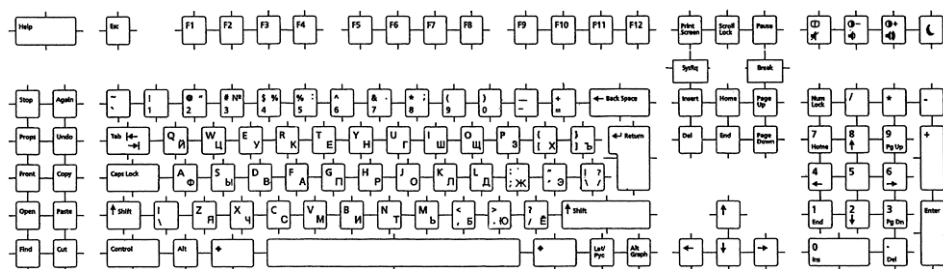


图 3-3 西里尔语（俄语）键盘

下图显示了德语键盘。

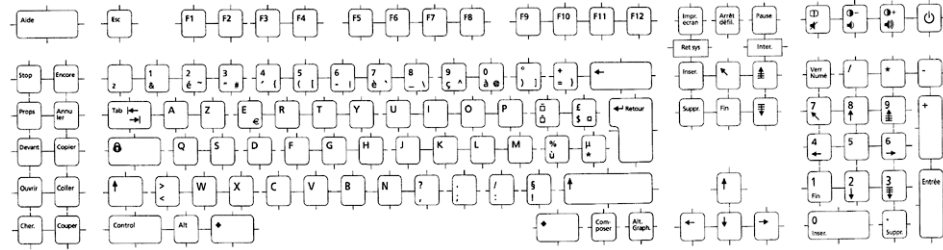


图 3-7 德语键盘

下图显示了意大利语键盘。

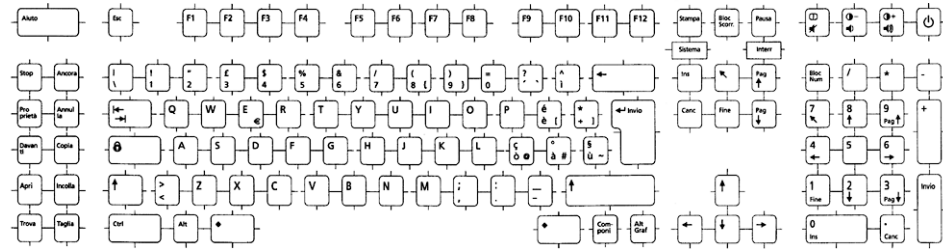


图 3-8 意大利语键盘

下图显示了日语键盘。

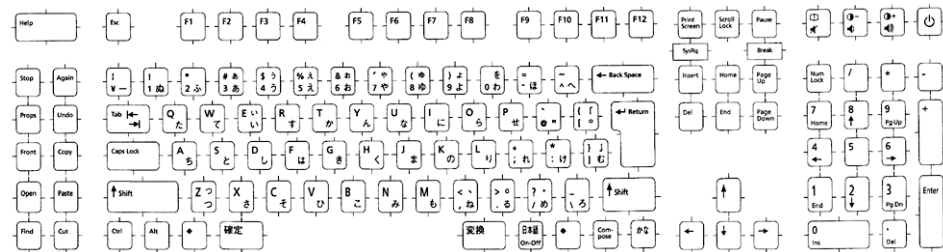


图 3-9 日语键盘

下图显示了葡萄牙语键盘。

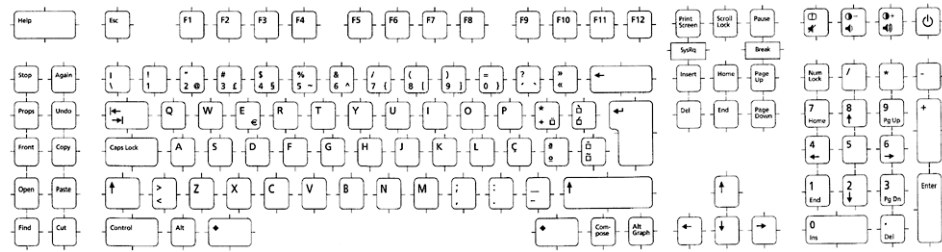


图 3-13 葡萄牙语键盘

下图显示了西班牙语键盘。

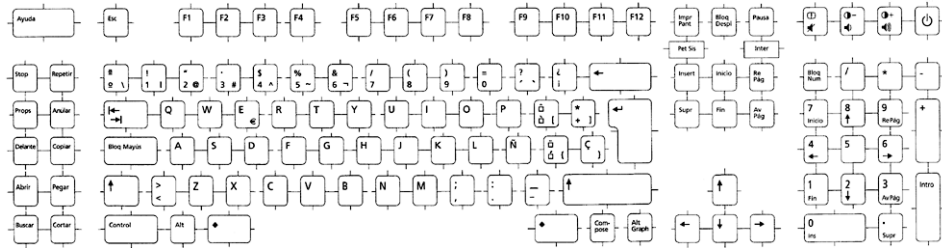


图 3-14 西班牙语键盘

下图显示了瑞典语键盘。

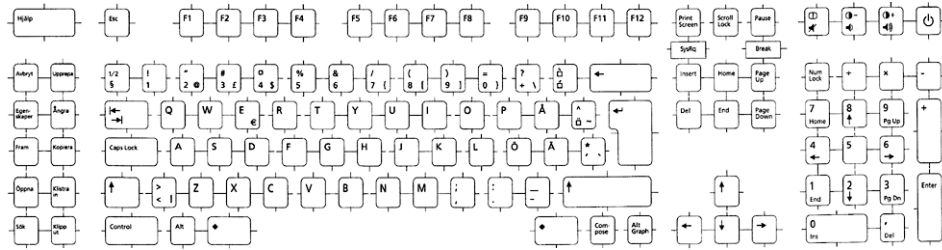


图 3-15 瑞典语键盘

下图显示了瑞士（法语）键盘。

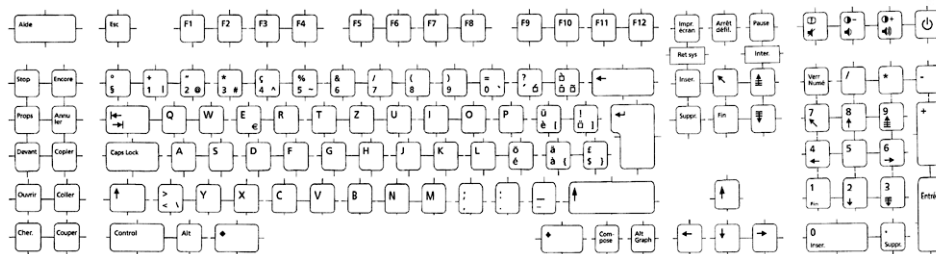


图 3-16 瑞士（法语）键盘

下图显示了瑞士（德语）键盘。

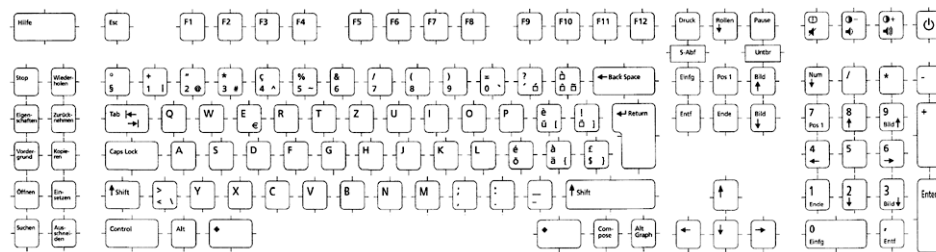


图 3-17 瑞士（德语）键盘

下图显示了繁体中文键盘。

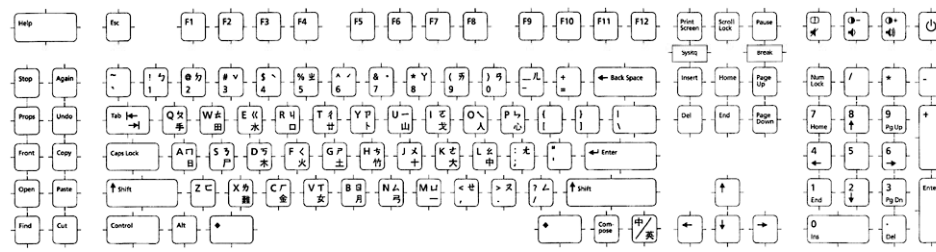


图 3-18 繁体中文键盘

下图显示了土耳其语 F 型键盘。

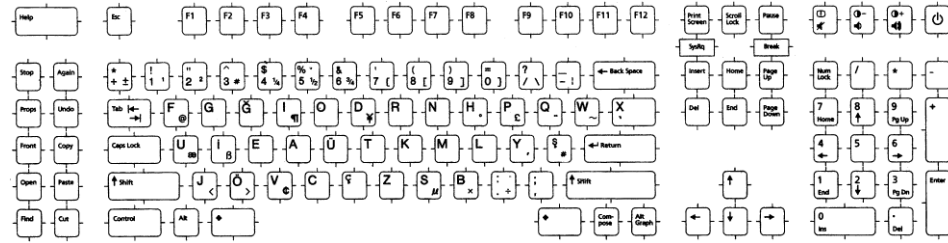


图 3-19 土耳其语 F 型键盘

下图显示了土耳其语 Q 型键盘。

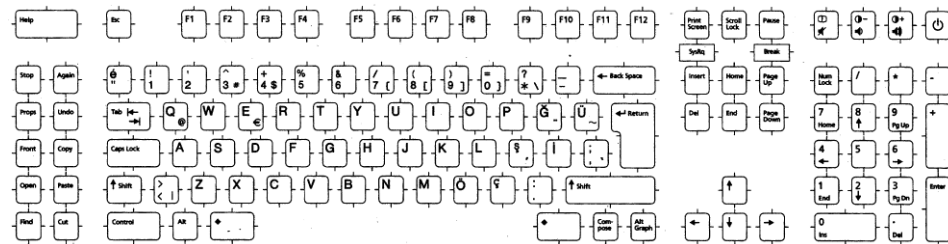


图 3-20 土耳其语 Q 型键盘

下图显示了英国键盘。

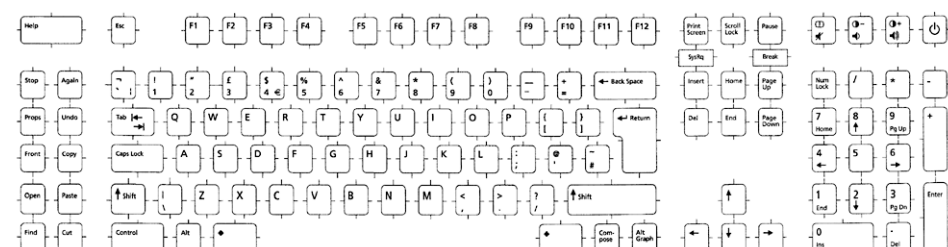


图 3-21 英国键盘

下图显示了美国键盘。

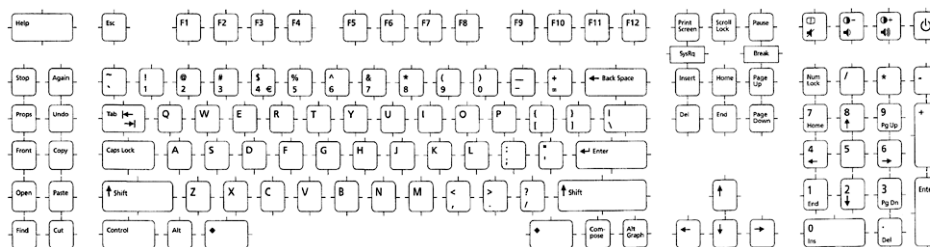


图 3-22 美国键盘

下图显示了美国/UNIX 键盘。

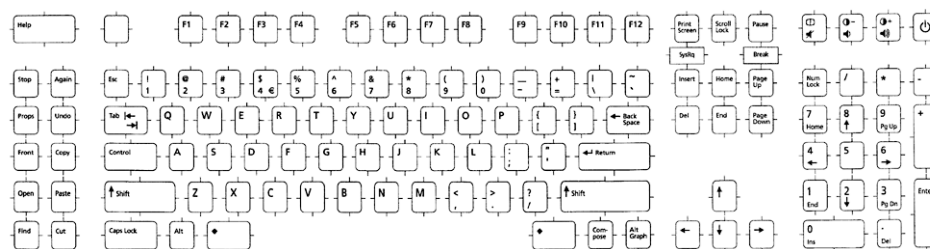


图 3-23 美国/UNIX 键盘

新增 Solaris 键盘软件支持

本发行版中提供下列其他键盘的软件支持。

- 俄罗斯 6 型 USB 键盘
- 爱沙尼亚 6 型 USB 键盘
- 加拿大法语 6 型 USB 键盘
- 波兰语程序员 5 型键盘

该软件使俄罗斯、加拿大、爱沙尼亚和波兰的用户能够修改标准美国键盘布局以满足不同的语言需要。目前，这些其他键盘类型还没有可用的硬件。要利用此新增的键盘软件，请执行本节中的过程中的步骤。

▼ 如何访问爱沙尼亚 6 型 USB 键盘支持

1. 将 `/usr/openwin/share/etc/keytables/keytable.map` 文件中的 `US6.kt` 项更改为 `Estonia6.kt`。

已修改的项如下所示：

```
6          0          Estonia6.kt
```

2. 将下列项之一添加到 `/usr/openwin/share/lib/locale/iso_8859_15/Compose` 文件。

已修改的项如下所示：

```
<scaron>   : "/xa8"   scaron  
<scaron>   : "/xa6"   scaron  
<scaron>   : "/270"   scaron  
<scaron>   : "/264"   scaron
```

3. 重新引导系统以实现更改。

▼ 如何访问加拿大法语 6 型 USB 键盘支持

1. 将 `/usr/openwin/share/etc/keytables/keytable.map` 文件中的 `US6.kt` 项更改为 `Canada6.kt`。

已修改的项如下所示：

```
6          0          Canada6.kt
```

2. 重新引导系统以实现更改。

▼ 如何访问波兰语程序员 5 型键盘支持

1. 将 `/usr/openwin/share/etc/keytables/keytable.map` 文件中的 `Poland5.kt` 项更改为 `Poland5_pr.kt`。

已修改的项如下所示：

```
6          0          Poland5_pr.kt
```

2. 重新引导系统以实现更改。

第 4 章

支持的亚洲语言环境

本章提供有关日语、印度语和泰语的本地化相关信息的信息。本章包含以下部分：

- 第 73 页 “日语本地化”
- 第 77 页 “印度语本地化”
- 第 104 页 “泰语本地化”

日语本地化

这部分描述日语语言环境特定的信息。

日语语言环境

在当前 Solaris 环境中，可使用四种支持不同字符编码的日语语言环境。ja 和 ja_JP.eucJP 语言环境基于日语 EUC。ja_JP.eucJP 语言环境符合 UI-OSF 日语环境实现协议 1.1 版，ja 语言环境符合早期 Solaris 版本中的传统规范。ja_JP.PCK 语言环境基于 PC 日文汉字代码（称为 Shift_JIS），而 ja_JP.UTF-8 基于 UTF-8。

有关日语 EUC 和该字符集之间的映射，请参见 eucJP(5) 手册页。有关显示 PC 日文汉字代码和该字符集的映射，请参见 PCK(5) 手册页。

日语字符集

受支持的日语字符集包括：

- JIS X 0201-1976
- JIS X 0208-1990

- JIS X 0212-1990
- JIS X 0213-2000（仅限于 Unicode 4.0 中定义的字符）

JIS X 0212-1990 在 `ja_JP.PCK` 语言环境中不受支持。JIS X 0213-2000 仅在 `ja_JP.UTF-8` 语言环境中受到支持。并不是在 JIS X 0213-2000 中定义的所有字符都可用。只有在 Unicode 4.0 字符集中定义的那些字符可用。

还支持供应商定义的字符 (VDC) 和用户定义的字符 (UDC)。VDC 占用 JIS X 0208-1990 或 JIS X 0212-1990 的未使用（保留）代码点。UDC 与 VDC 占用相同的代码点，但分配给 VDC 的那些代码点除外。

日语字体

支持三种日语字体格式。位图、TrueType 和 Type1。日语 Type1 字体只包括用于打印的 JIS X 0212。Type1 字体还由 UDC 使用。

下表对日语位图字体进行了描述。

表 4-1 日语位图字体

完整系列名	子系列	格式	供应商	编码
sun gothic	R、B	PCF(12,14,16,20,24)		JIS X 0208-1983、 JIS X 0201-1976
sun minchou	R	PCF(12,14,16,20,24)		JIS X 0208-1983、 JIS X 0201-1976
ricoh hg gothic b	R	PCF(10,12,14,16,18,20,24)	RICOH	JIS X 0208-1983, JIS X 0201-1976
ricoh hg mincho l	R	PCF(10,12,14,16,18,20,24)	RICOH	JIS X 0208-1983, JIS X 0201-1976
ricoh gothic	R	PCF(10,12,14,16,18,20,24)	RICOH	JIS X 0212-1990, JIS X 0213-2000
ricoh mincho	R	PCF(10,12,14,16,18,20,24)	RICOH	JIS X 0212-1990, JIS X 0213-2000
ricoh heiseimin	R	PCF(12,14,16,18,20,24)	RICOH	JIS X 0212-1990

下表对日语 TrueType 字体进行了描述。

表 4-2 日语 TrueType 字体

完整系列名	子系列	格式	供应商	编码
ricoh hg gothic b	固定	TrueType	RICOH	JIS X 0208-1983、JIS X 0201-1976
ricoh hg mincho l	固定	TrueType	RICOH	JIS X 0208-1983、JIS X 0201-1976
ricoh hg gothicb sun	固定、成比例	TrueType	RICOH	JIS X 0201-176、JIS X 0208-1983、JIS X 0213-2000
ricoh hg minchol sun	固定、成比例	TrueType	RICOH	JIS X 0201-1976、JIS X 0208-1983、JIS X 0213-2000
ricoh heiseimin	固定	TrueType	RICOH	JIS X 0212-1990

日语输入系统

ATOK12 是当前 Solaris 环境中的缺省日语输入系统。安装了日语语言环境后，ATOK12 就可用于所有日语语言环境和所有 UTF-8 语言环境。Wnn6 日语输入系统也可用于所有日语语言环境。您可以在桌面菜单中切换输入系统。对于日语 Solaris 1.x BCP 支持，可使用 kkcvt 日语输入系统。

以下过程介绍如何使用 ATOK12 输入法输入日语文本。

▼ 如何使用 ATOK 输入法

1. 按 **Ctrl-空格键**可打开输入转换。
2. 键入要转换的文本的假名字符。
例如，可以键入对应日文汉字 *henkan* 的假名。
3. 按空格键以显示可用于假名拼写的日文汉字转换候选字。
4. 键入要选择的转换候选字的编号。
5. 按“返回”可提交完成的日文汉字假名拼写。
另外，您可以按向下箭头键，仅提交所选字符。
6. 按 **Ctrl-空格键**可关闭输入转换。

日语终端的终端设置

要在基于字符的终端 (TTY) 上使用日语语言环境，必须使用终端设置以便使行编辑正常工作。

- 如果您的终端是 CDE 终端仿真器 (dtterm)，请在任何日语语言环境 (ja、ja_JP.PCK 或 ja_JP.UTF-8) 中使用带参数 -defeucw 的 stty(1)。例如，在 ja 语言环境中应键入：

```
% setenv LANG ja
% stty defeucw
```

- 如果您的终端不是 CDE 终端仿真器，但终端的代码集与当前语言环境的代码集相同，请使用带参数 -defeucw 的 stty(1)。
- 如果您的终端的代码集与当前语言环境的代码集不匹配，请使用 setterm(1) 以启用代码转换。例如，如果您正在使用 ja 语言环境，但您的终端要求 PCK (Shift_JIS 代码)，则应键入：

```
% setenv LANG ja
% setterm -x PCK
```

有关详细信息，请参见 setterm(3CURSES) 手册页。

日语 iconv 模块

使用 iconv(1) 和 iconv(3) 可支持多个日语代码集转换。有关详细信息，请参见 iconv_ja(5) 手册页。

用户定义的字符支持

用户定义的字符实用程序 sdtudctool 处理轮廓 (Type1) 和位图 (PCF) 字体。某些实用程序也可用于移植由早期发行版中的旧实用程序创建的 UDC 字体，如 fontedit、type3creator 和 fontmanager。

部分语言环境和完全语言环境之间的差别

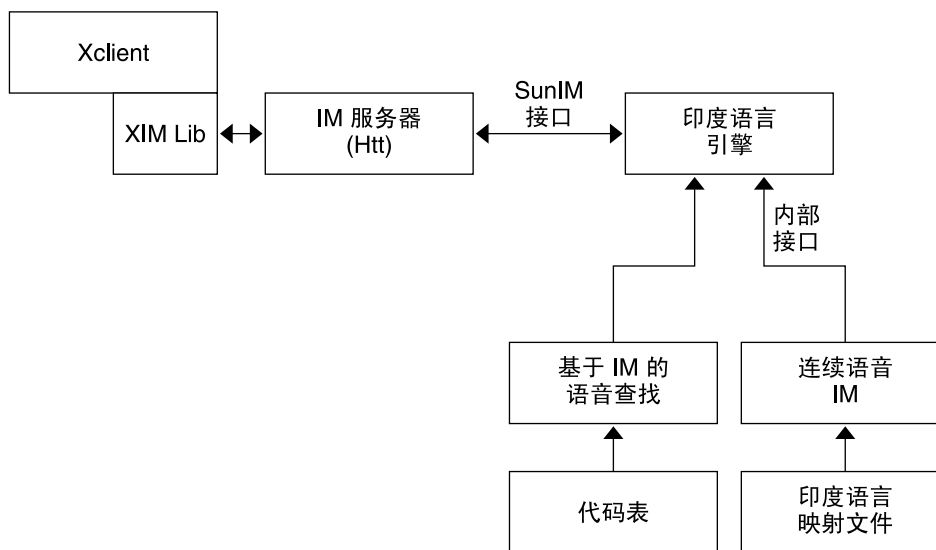
下列组件只在带有 Languages CD 的日语完全语言环境中可用：

- 已翻译的信息、帮助和手册页
- Wnn6 日语输入系统
- 日语 Solaris 1.x BCP 支持
- Mincho (min*) 字样位图字体
- 用于打印的 JIS X 0212 Type1 字体
- 日语特定的哑打印机和 postprint 支持
- 传统的日语实用程序，如 kanji(1)

印度语本地化

对于 UTF-8 语言环境中支持的所有印度语语言，基于拼音查找的输入法 (Shabdalipi) 和连续拼音输入法都可用。使用这些输入法和虚拟键盘，可以在所有 CDE 应用程序中输入印度语文本。

以下数据流说明了印度语输入过程的工作方式。



▼ 如何使用印度语输入法

1. 单击输入状态区可显示输入法选择菜单。
2. 从菜单中选择输入法。
另外，您可以按 F6 键，以便从可用的输入法中进行选择。
还可以键入 Compose-hi 键序列，选择您以前使用的输入法。
3. 按 F5 键可选择要使用的印度语脚本。
 - a. 对于基于键盘（印度语 INSCRIPT 键盘）的输入法，请使用第 78 页“印度语键盘”中显示的键盘图像。
 - b. 对于基于拼音查找的输入法，请键入与目标脚本中的字符对应的第一个英文拼音等效字符。

了解映射

第 81 页“基于连续拼音的输入法的映射”中的图像显示了英文标记与其在所支持的各个目标脚本中的等效代码点之间的映射。CONSONANT 类别表示英文标记与脚本的辅音字母之间的映射。VOWEL 类别表示英文标记与脚本的元音字母之间的映射。OTHER 类别包含不具有辅音字母和元音字母（其格式不随两边字符的变化而变化）特性的字符的映射。

关键字 CONSONANT、VOWEL 和 OTHER 还表示这些字符是 Unicode 标准的一部分。SPECIAL CONSONANT、SPECIAL VOWEL 或 SPECIAL OTHER 表示：虽然这些字符原则上显示辅音字母、元音字母或其他字符的特性，但它们未正式包含在 Unicode 标准中且与字体相关。它们是在 Unicode 私有用户区中分配的代码点值。它们在 Solaris UTF-8 语言环境中受到支持，而映射在不同平台上则可能不工作。

这些映射文件与系统中的映射文件不同，而是进行过一些编辑，删除了对本文讨论内容无关紧要的关键字。

在 VOWELS 和 SPECIAL VOWELS 段中，根据上下文，为同一英文标记显示独立格式和相关格式。请参见第 103 页“连续拼音输入法的工作原理”。

malayalam 脚本包含特殊的 'CHILLU' 段，该段实际上是 SPECIAL OTHER 类别。

基于连续拼音的输入法的映射

下图显示英文至目标印度语脚本中的等效拼音字符的现有映射。使用这些图作为参考，直到您了解所使用脚本的所有映射。此处给出的映射非常直观，所以您应当不需要看图即可输入大多数字符。

注意 – 这些映射中包含诸如 ‘/’ 和 ‘|’ 等特殊字符，作为使用 ‘\’ 字符换码的映射的一部分。如果不进行换码，那么如果多个标记表示同一个 UTF-8 字符，则 ‘|’ 字符将作为分隔符。

图 4-1、图 4-2 和图 4-3 显示辅音字母、元音字母和其他字符的英文至孟加拉语的映射。

辅音字母

k	ক	Dh	ঢ	r	র
kh	খ	N	ণ	l	ল
g	গ	th	ত	sh	শ
gh	ঘ	thh	থ	S	ষ
nng	ঙ	d	দ	s	স
ch	চ	dh	ধ	h	হ
chh	ছ	n	ন	rr\.	ড়
j	জ	p	প	rh\.	ঢ়
jh	ঝ	f ph	ফ	y\.	য়
ny yn	ঞ	b	ব	v	ব্ব
T	ট	bh	ভ	V	ব্ব্ব
Th	ঠ	m	ম		
D	ড	y	য		

图 4-1 孟加拉语辅音字母映射

元音字母	相关形式	独立形式
a	অ	।
aa	আ	।
i	ই	ি
ee ii	ঈ	ী
u	উ	ু
oo uu	ঊ	ূ
r\^	ঋ	ৃ
rr\^	ঌ	্ল
n\^	ঐ	ে
nn\^	ঔ	ে
e	এ	ে
ai	ঐ	ে
o	ও	ে
au	ঔ	ে

图 4-2 孟加拉语元音字母映射

其他字符	
UM	ঊ
\.N	ঋ
M	ঌ
H	঍
OU	ঔ
Rs	৳
Rs \ .	৳.

图 4-3 其他孟加拉语字符映射

图 4-4、图 4-5 和图 4-6 显示辅音字母、元音字母和其他字符的英文至古吉拉特语的映射。

辅音字母

k	ક	D	ડ	m	મ
kh	ખ	Dh	ઢ	y	ય
g	ગ	N	ણ	r	ર
gh	ઘ	th	ત	l	લ
ng	ંગ	thh	થ	zh	જ
c	ચ	d	દ	w v	વ
ch	છ	dh	ધ	S	શ
j	જ	n	ન	s	ષ
jh	ઝ	p	પ	sh	સ
ny	ન્ય	ph	ફ	h	હ
T	ટ	b	બ		
Th	ઠ	bh	ભ		

图 4-4 古吉拉特语辅音字母映射

元音字母	相关形式	独立形式
a	ਅ	
aa	ਅੁ	ੴ
i	ਏ	ੴ
ee	ਏ	ੴ
u	ਊ	ੴ
oo uu	ਊ	ੴ
r\^	ੴ	ੴ
e	ਏ	ੴ
E	ਏ	ੴ
ai	ਏ	ੴ
o	ਐ	ੴ
O	ਐ	ੴ
au	ਐ	ੴ

图 4-5 古吉拉特语元音字母映射

其他字符	◌̣
NG	◌̣
M	◌̣
H	◌̣
OM	ૐ
RR	૨૨
kt	◌̣
av	૬

图 4-6 其他古吉拉特语字符映射

图 4-7、图 4-8 和图 4-9 显示辅音字母、元音字母和其他字符的英文至果鲁穆奇语的映射。

辅音字母

k	ਕ	Dh	ਢ	r	ਰ
kh	ਖ	N	ਣ	l	ਲ
g	ਗ	t	ਤ	ll	ਲ਼
gh	ਘ	th	ਥ	v	ਵ
ny	ਙ	d	ਦ	sh	ਸ਼
ch	ਚ	dh	ਧ	s	ਸ
chh	ਛ	n	ਨ	h	ਹ
j	ਜ	p	ਪ	khh	ਖ਼
jh	ਝ	ph	ਫ	ghh	ਗ਼
nj	ਞ	b	ਬ	z	ਜ਼
T	ਟ	bh	ਭ	rr	ਰ਼
Th	ਠ	m	ਮ	f	ਫ਼
D	ਡ	y	ਯ		

图 4-7 果鲁穆奇语辅音字母映射

元音字母	相关形式	独立形式
a	ਅ	
aa	ਆ	ੜ
i	ਇ	ਫ
ee ii	ਈ	ੀ
u	ਉ	ੁ
oo uu	ਊ	
E	ਏ	ੲ
ai	ਐ	ੳ
O	ਓ	ੴ
au	ਔ	ੵ

图 4-8 果鲁穆奇语元音字母映射

其他字符	
um	ੰ
\.N	ੰ
UH	ੰ
AD	ੁ
IR	ੲ
UR	ਓ
OM	ੴ

图 4-9 其他果鲁穆奇语字符映射

图 4-10、图 4-11 和图 4-12 中显示辅音字母、元音字母和其他字符的英文至印度语映射。

辅音字母

k	क	t	त	L	ळ
kh	ख	th	थ	\.L	ळ
g	ग	d	द	v	व
gh	घ	dh	ध	S	श
ng	ङ	n	न	sh	ष
c	च	\.n	न	s	स
ch	छ	p	प	h	ह
j	ज	f ph	फ	q	क्व
jh	झ	b	ब	\.kh	ख
ny	ञ	bh	भ	\.gh	ग
T	ट	m	म	\.j	ज
Th	ठ	y	य	\.D	ड
D	ड	r	र	\.Dh	ढ
Dh	ढ	R	ऋ	\.f \.ph	ऋ
N	ण	l	ल	\.y	य

图 4-10 印度语辅音字母映射

元音字母	相关形式	独立形式
a	अ	
aa	आ	।
i	इ	ि
ee	ई	ी
u	उ	ु
oo	ऊ	ू
r\^	ऋ	ॠ
rr\^	ॠ	ॡ
l\^	ऌ	ॣ
ll\^	ॣ	।
EE	ए	े
E	ए	ै
e	ए	ॆ
ai	ऐ	े
OO	ऑ	ॊ
O	ओ	ो
o	ओ	ौ
au	औ	्

图 4-11 印度语元音字母映射

其他字符

OM	ॐ	U\~	ॠ
\.C	.	A\^	ॡ
M	.	A\~	ॢ
H	॥	\	ॣ
\.N	.	\ \	।
V\^	॥	EH	॥
U\^	॥		॥

图 4-12 其他印度语字符映射

图 4-13、图 4-14 和图 4-15 中给出辅音字母、元音字母和其他字符的英文至埃纳德语的映射。

辅音字母

k Kh K	က	Dh	ဒ	y	ယ
g	ဂ	N	န	R r	ရ
G gh	ဧ	t	တ	rx rh	ဧ
\~G	ဃ	th	ထ	l	လ
c ch	ဇ	d	ဒ	ll	လ
C CH	ဇ	dh	ဒ	w v	ဗ
j	ည	n	န	S sh	ဆ
J jh	ည	p	ပ	Sh	ဆ
\~J	ဋ	p ph	ပ	s	ဆ
T	တ	b	ဗ	\~h h	ဟ
Th	ထ	B bh	ဗ	f	ဖ
D	ဒ	m	မ		

图 4-13 埃纳德语辅音字母映射

元音字母	相关形式	独立形式
a	Ḃ	
aa	Ḃ	Ḃ
i	Ḃ	Ḃ
ee	Ḃ	Ḃ
u	Ḃ	Ḃ
U oo	Ḃ	Ḃ
r\^	Ḃ	Ḃ
R\^	Ḃ	Ḃ
e	Ḃ	Ḃ
E	Ḃ	Ḃ
ai	Ḃ	Ḃ
o	Ḃ	Ḃ
O	Ḃ	Ḃ
au ou	Ḃ	Ḃ

图 4-14 埃纳德语元音字母映射

其他字符	
M	೦
H	ಃ
OU	ೳ
LM	ೳ
RR \ ^	ಋ
\ ~N	ೠ

图 4-15 其他埃纳德语字符映射

图 4-16、图 4-17 和图 4-18 给出辅音字母、元音字母和其他字符的英文至马拉雅拉姆语的映射。

辅音字母

k	ക	th	ത	zh	ഴ
kh	ഖ	thh	ഥ	w v	വ
g	ഗ	d	ദ	s	ശ
gh	ഘ	dh	ധ	sh	ഷ
ng	ങ	n	ന	s	സ
ch	ച	p	പ	h	ഹ
chh	ഛ	f ph	ഫ		
j	ജ	b	ബ	特殊辅音字母	
jh	ജ്ഞ	bh	ഭ	nt	ന്ത
nj	ഞ	m	മ	nth	ന്ത
T	ട	y	യ	nnj	ഞ്ഞ
Th	ത	r	ര	nk	ങ്ക
D	ഡ	R	റ	nng	ങ്ങ
Dh	ഢ	l	ല	t	റ്റ
N	ണ	L	ള		

图 4-16 马拉雅拉姆语辅音字母映射

元音字母	相关形式	独立形式
a	അ	
A aa	ആ	ഓ
i	ഇ	ീ
ee	ഇയ്യ	ീ
u	ഉ	ൂ
oo	ഉയ്യ	ൂ
r^	ഋ	ൃ
e	എ	ൈ
E	ഏ	േ
ai	ഐ	ൈ
o	ഒ	ൊ
O	ഓ	ോ
au	ഔ	ൌ
特殊元音字母		
ou	ഔ	ൗ

图 4-17 马拉雅拉姆语元音字母映射

其他字符		CHILLU
M	o	n \ ~ ന്
H	ഃ	N \ ~ ണ്
rr \ ^	ഋ	l \ ~ ൽ
U	ൺ	L \ ~ ശ്
UU	൹	r \ ~ ര്

图 4-18 其他马拉雅拉姆语字符映射

图 4-19 和图 4-20 给出辅音字母和元音字母的英文至 泰米尔语的映射。

辅音字母

k g K G	க	R	ற
nG ng	ங	TR	ற்ற
ch CH	ச	DR	ற
j J	ஜ	l	ல
gn Gn	ஞ	L	ள
t d	ட	zh ZH	ழ
N	ண	w v W V	வ
th dh TH DH	த	S	ஷ
n	ந	s	ஸ
n \ ^	ன	h H	ஹ
p b P B	ப	ndh	ந்த
m M	ம	nth	ந்த
y Y	ய	nj NJ	ஞ்ச
r	ர	f ph F PH	ஃப

图 4-19 泰米尔语辅音字母映射

元音字母	相关形式	独立形式
a	அ	
A aa	ஆ	ஈ
i	இ	ஊ
I ii ee	ஈ	ஊ
u	உ	ஊ
oo U	ஊ	ஊ
e	எ	ஊ
E ae	ஏ	ஊ
ai	ஐ	ஊ
o	ஓ	ஊ
O oa oe	ஓ	ஊ
ow ou au	ஔ	ஊ
其他字符		
H	ஹ	

图 4-20 泰米尔语元音字母映射

图 4-21、图 4-22 和图 4-23 给出辅音字母、元音字母和其他字符的英文至 Telugu 的映射。

辅音字母

k	క	D	డ	m	మ
K Kk kh	ఖ	Dh	ఢ	y	య
g	గ	N nh	ణ	r	ర
G Gh gh	ఘ	t	ట	rr	ఱ
\~m	ఞ	th	ఠ	l	ల
ch c	ఛ	d	ద	L	ఱ
C Ch	ఞ	dh	ఢ	w W V v	వ
j	జ	n	న	S	ష
J Jh jh	ఙ	p	ప	sh	శ
\~n	ఞ	F f P Ph ph	ఫ	s	స
T	త	b	బ	h	హ
Th	ఠ	Bh B bh	భ		

图 4-21 Telugu 辅音字母映射

元音字母	相关形式	独立形式
a	అ	
A aa	ఆ	౧
i	ఇ	౨
I ia ee ii	ఈ	౩
u	ఉ	౪
ua U oo uu	ఊ	౫
R	ఋ	౬
Ru	ౠ	౭
E ae ea	ఎ	౮
ai	ఐ	౯
o	ఒ	౧౦
oa oe O	ఓ	౧౧
ou au	ఔ	౧౨

图 4-22 Telugu 元音字母映射

其他字符	
\ . N	ఁ
M	ఠ
H	ః
OU	ఌ
LM	఍
RR	ఙ్
Nu	ఞ

图 4-23 其他 Telugu 字符映射

连续拼音输入法的工作原理

对于每个印度语脚本，一个 ‘virama’ 或等效标记与一个辅音字母组合，成为一个辅音字母的一半形式（或可组合形式）。只要键入与辅音字母对应的多键组合，就会输出辅音字母 + virama 的形式，这表明可以开始组合字符。

先输入的辅音字母将假定其另一半的形式，并且在其后输入元音字母时成为满音节或其变化形式。

两个连续辅音字母仍然是可组合的一半形式。布局引擎可将一半形式转换为单个组合字符，或者保持在句法上对所有语言有效的独立形式。

一个单词的任何起始元音字母或者后面跟有另一元音字母的任何元音字母都为独立形式。后面紧随着辅音字母的元音字母为相关格式。

在任何上下文中都不改变形状的字符称为其他字符。这些字符既不是辅音字母也不是元音字母。

不组成字符的数字和其他标点符号标记被一一映射。

使用以上原则，解析器被编写为将输入解析为不同类别并输出特定于语言的 Unicode 代码点。连续拼音输入法引擎不处理布局或呈现，相关处理由系统中的其他模块完成。

泰语本地化

当前 Solaris 环境支持三种泰语输入级别和四种泰语键盘布局。

泰语输入法

本发行版中支持下列泰语输入法。Thai IT 标准中指定对这些输入法进行字符序列检查。

1. 传递级别，不进行任何输入检查
2. 基本的输入检查级别
3. 严格的输入检查级别

对传递级别不进行序列检查，该级别是本发行版中的缺省级别，这一点和以前的 Solaris 发行版一样。

您可以使用 F2 功能键在各个输入级别之间切换。

泰语键盘布局

为泰语输入法支持四种不同的键盘布局。

- Kedmanee (TIS820-2531) 键盘布局。Kedmanee 布局是为打字机设计的，不是为计算机键盘设计的。打字机键盘的有限键数表明，布局中部分特殊泰语字符不可用。TIS820-2531 已采用 Kedmanee 布局以使用计算机键盘。



- TIS820-2538 键盘布局。该增强 Kedmanee 布局是 TIS820-2531 布局的更新版本，包括原来的 Kedmanee 布局中不可用的部分特殊泰语字符。当前，TIS820-2538 是 Thai Industrial Standard Institute 颁发的唯一一个泰语键盘布局标准。



- Pattajoti 键盘布局。还为打字机设计了 Pattajoti 布局，但改进了指压分布。



- 可配置键盘布局。泰语输入法的用户定义键盘布局。

泰语输入法辅助窗口

泰语输入法辅助窗口支持下列功能和实用程序：

- 输入级别切换。您可以单击辅助调色板中的输入级别按钮，以选择传递、基本或严格做为输入级别。
- 泰语虚拟键盘。您可以单击键盘按钮来显示泰语虚拟键盘，以用于输入泰语字符。

第 5 章

UTF-8 语言环境支持概述

本章对 UTF-8 语言环境支持进行概述。本章包含以下主题：

- 第 107 页 “Unicode 概述”
- 第 108 页 “Unicode 语言环境：en_US.UTF-8 支持”
- 第 110 页 “关于桌面输入法”
- 第 139 页 “系统环境”
- 第 143 页 “代码转换”
- 第 144 页 “DtMail 支持”
- 第 146 页 “程序设计环境”

Unicode 概述

Unicode 是通用字符编码标准，用于表示文本以供计算机处理。Unicode 与国际标准 ISO/IEC 10646-1:2000 和 ISO/IEC 10646-2:2001 完全兼容，并且与 ISO/IEC 10646 包含完全相同的字符和编码点。Unicode 标准提供有关相关字符及其用途的其他信息。符合 Unicode 的所有实现程序也同时符合 ISO/IEC 10646。

Unicode 提供了一种对多语种纯文本进行一致编码的方法，便于国际文本文件的交换。处理多语种文本的计算机用户、商务人员、语言学家、研究人员、科学家等会发现 Unicode 标准可大大简化他们的工作。Unicode 标准对于经常使用数学符号和其他技术字符的数学家和技术人员也很有价值。

Unicode 能支持的最大可能代码点数是 1,114,112，位于 17 个 16 位平面中。每一个平面都可支持 65,536 个不同的代码点。

在 Unicode 可支持的一百多万代码点中，版本 4.0 当前在平面 0、1、2 和 14 中定义 96382 个字符。平面 15 和 16 用于专用字符，这类字符也称为用户定义的字符。平面 15 和 16 总共可支持 131,068 个用户定义的字符。

Unicode 可以使用下列任意字符编码方案：

- UTF-8
- UTF-16
- UTF-32

UTF-8 是 Unicode 的一种可变长度编码形式，它透明地保留了 ASCII 字符代码值。该形式在 Solaris Unicode 语言环境中用作文件代码。

UTF-16 是 Unicode 的一种 16 位编码形式。在 UTF-16 中，多达 65,535 个字符被编码为单个 16 位值。映射在 65,535 到 1,114,111 的字符被编码为成对的 16 位值（代理）。

UTF-32 是 Unicode 的一种固定长度的 21 位编码形式，通常用在 32 位容器或数据类型中。该形式在 Solaris Unicode 语言环境中用作进程代码（宽字符代码）。

有关 Unicode 标准和 ISO/IEC 10646 及其各种表示形式的详细信息，请参阅下列来源：

- Unicode Consortium 提供的《Unicode 标准，版本 4.0》
- ISO/IEC 10646-1:2000、信息技术-通用多八进制字符集 (UCS) - 第 1 部分：体系结构和基本多语种平面
- ISO/IEC 10646-2：信息技术-通用多八进制字符集 (UCS) - 第 2 部分：脚本和符号的次要多语种平面、CJK 象形文字的辅助平面、特殊用途平面
- Unicode Consortium web 站点：<http://www.unicode.org/>。

Unicode 语言环境：en_US.UTF-8 支持

Unicode/UTF-8 语言环境支持 Unicode 4.0。en_US.UTF-8 语言环境使用 UTF-8 作为其字符集来提供多脚本处理支持。该语言环境处理多个脚本中的输入和输出文本，是 Solaris 操作系统中第一个具有此功能的语言环境。其他 UTF-8 语言环境的功能与 en_us.UTF-8 的功能相似。后面对 en_US.UTF-8 的探讨同样适用于这些语言环境。

注意 – UTF-8 是由 X/Open-Uniform Joint Internationalization Working Group (XoJIG) 在 1992 年制定的 Unicode/ISO/IEC 10646-1 的文件系统安全通用字符集转换格式，并在 1996 年作为 ISO/IEC 10646-1:1993 的第二修正案被 ISO 和 IEC 批准。该标准已由 Unicode Consortium、国际标准化组织和国际电工技术委员会采用而成为 Unicode 4.0 和 ISO/IEC 10646-1 的一部分。

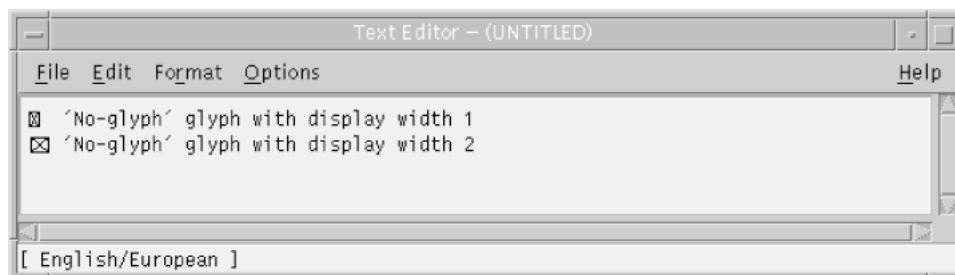
Solaris 环境中的 Unicode 语言环境支持对在 Unicode 4.0 与 ISO/IEC 10646-1 和 10646-2 中定义的每一个代码点值的处理。所支持的脚本不仅包括全欧洲语言和亚洲语言，还包括一些复杂的文本布局，如阿拉伯语、希伯来语、印度语和泰语。

注意 – 某些 Unicode 语言环境，特别是亚洲语言环境，包括更多的 Kanji 或 Hanzi 字型。

由于字体资源有限，当前 Solaris Unicode 语言环境仅包括来自下列字符集的字符字型。

- ISO 8859-1 (大多数西欧语言, 如英语、法语、西班牙语和德语)
- ISO 8859-2 (大多数中欧语言, 如捷克语、波兰语和匈牙利语)
- ISO 8859-4 (斯堪的纳维亚和波罗的海语)
- ISO 8859-5 (俄语)
- ISO 8859-6 (阿拉伯语, 包括许多具有更多表示形式的字符字型)
- ISO 8859-7 (希腊语)
- ISO 8859-8 (希伯来语)
- ISO 8859-9 (土耳其语)
- TIS 620.2533 (泰语, 包括许多具有更多表示形式的字符字型)
- ISO 8859-15 (大多数带有欧元符号的西欧语言)
- GB 2312-1980 (简体中文)
- JIS X 0201-1976, JIS X 0208-1990 (日语)
- KSC 5601-1992 附件 3 (朝鲜语)
- GB 18030 (简体中文)
- HKSCS (繁体中文, 中国香港特别行政区)
- Big5 (繁体中文, 中国台湾地区)
- IS 13194.1991, 也称为 ISCII (印度语, 包括许多具有更多表示形式的字符字型)

如果您试图查看 en_US.UTF-8 语言环境中没有其相应字型的字符, 则该语言环境将显示 no-glyph 字型, 如下图所示:



语言环境可以在安装时选择, 并可指定为系统的缺省语言环境。

对于 64 位和 32 位的 Solaris 系统, 提供了同一级别的 en_US.UTF-8 语言环境支持。

注意 - Motif 和 CDE 桌面应用程序和库支持 en_US.UTF-8 语言环境。但 XView™ 和 OLIT 库不支持 en_US.UTF-8 语言环境。

关于桌面输入法

CDE 提供使用 Xm Toolkit 为国际化的应用程序输入本地化输入的能力。启用了 XmText [Field] 小部件以便与各个语言环境中的输入法相连接。输入法被国际化，因为某些语言环境以从右向左、从上到下等方向写入它们的文本。在同一个应用程序中，您可以使用应用多种字体的不同输入法。

预编辑区显示正被预编辑的字符串。可以通过四种模式写入文本：

- OffTheSpot
- OverTheSpot (缺省)
- 根
- 无

在 OffTheSpot 模式下，此位置在状态区右边主窗口的正下方。在 OverTheSpot 模式下，预编辑区位于光标位置处。在根模式下，预编辑区和状态区与客户程序窗口分离。

有关详细信息，请参见 VendorShell(3X) 手册页上的 XmNpreeditType 资源描述。

注意 – 在当前 Solaris 环境中，对简体/繁体中文、日语和朝鲜语可采用本地亚洲语言输入法。这些方法是对 Unicode 语言环境的当前多脚本输入法的补充。

第 111 页“访问输入法”包括对所选输入法、如何使用这些输入法以及如何在这类输入法之间切换的描述。

语言选择和输入模式

Solaris Unicode 语言环境支持多种脚本。每个 Unicode 语言环境共有 14 种输入模式。

- 英语/欧洲语言
- 西里尔文
- 希腊文
- 阿拉伯语
- 希伯来语
- 泰语
- 日语
- 朝鲜语
- 简体中文
- 繁体中文
- 繁体中文 (香港特别行政区)
- 印度语
- Unicode 十六进制和八进制代码输入法
- 查表输入法

访问输入法

您可以使用书写键组合或输入模式选择窗口来切换到某个特定的输入模式。要访问输入模式选择窗口，请在应用程序窗口左下角的状态区中单击。输入模式选择窗口显示在下图中。



图 5-1 输入模式选择窗口

输入模式切换键序列

您可以使用表 5-1 中列出的键序列将当前输入模式更改为新的输入模式。使用这些键序列的唯一限制是，如果您当前正使用某个亚洲语言输入模式，则需要同时按下 **Ctrl** 键和空格键切换回英语/欧洲语言输入模式。一旦处于英语/欧洲语言输入模式，您就可以使用这些键序列随意切换到其他任何一种输入模式中。

下列键序列显示如何从英语/欧洲语言输入模式切换到西里尔文输入模式：

1. 按下书写键。
2. 按下并释放 **C** 键。
3. 按下 **C** 键。

表 5-1 输入模式切换键序列

键序列	输入模式
Ctrl-空格键	英语/欧洲语言
Compose c c	西里尔文
Compose g g	希腊文
Compose a r	阿拉伯语
Compose h h	希伯来语
Compose t t	泰语
Compose h i	印度语
Compose i n	印度语
Compose j a	日语
Compose k o	朝鲜语
Compose s c	简体中文
Compose t c	繁体中文
Compose h k	繁体中文 (香港特别行政区)
Compose u o	Unicode 八进制代码输入法
Compose u h	Unicode 十六进制代码输入法
Compose l l	查表输入法

英语/欧洲语言输入模式

英语/欧洲语言输入模式包括英语字母和带有变音标记的字符（例如，á、è、î、õ 和 ü）以及欧洲脚本中的一些字符（如 ÿ、ſ、ı）。

该输入模式是所有应用程序的缺省模式。输入模式显示在 GUI 应用程序窗口的左下角。

要插入带有变音标记的字符或 Latin-1、Latin-2、Latin-4、Latin-5 和 Latin-9 中的特殊字符，必须键入一个书写键序列，如下例所述。

要显示 Ä 字符：

1. 按下并释放书写键。
2. 同时按下 Shift 和 A 键。释放 Shift 和 A 键。
3. 按下并释放 " 键。

要显示 the ÿ 字符：

1. 按下并释放书写键。
2. 按下并释放 ? 键。
3. 按下并释放 ? 键。

如果键盘上没有书写键，您可以模仿书写键的操作，方法是同时按下 **Ctrl** 键和 **Shift** 键。

要从该语言环境中输入欧元货币符号（Unicode 值为 U+20AC），则可以使用下列任何一个输入序列：

- 同时按下 **AltGraph** 和 **E**
- 同时按下 **AltGraph** 和 **4**
- 同时按下 **AltGraph** 和 **5**

对于这些输入序列，您可以同时按下这两个键。如果键盘上没有 **AltGraph** 键，可以使用某些备用的欧元符号输入序列（如 **Compose e =** 或 **Compose c =**）。

下面各表显示 Solaris 操作系统中最常用的 Latin-1、Latin-2、Latin-3、Latin-4、Latin-5 和 Latin-9 脚本输入的书写序列。

下表列出常用的 Latin-1 书写键序列。

表 5-2 常用的 Latin-1 书写键序列

按下 Compose ，然后按下并释放	然后按下并释放	结果
空格键	空格键	无间断间隔
s	1	上标 1
s	2	上标 2
s	3	上标 3
!	!	倒置叹号
x	o	货币符号 ¢
p	!	段落符号 ¶
/	u	mu u
'	"	锐音符 ´
`	, (逗号)	下加符 ¸
-	"	分音符 ¨
-	^	长音符 ˘
o	o	度 °
x	x	乘号 ×

表 5-2 常用的 Latin-1 书写键序列 (续)

按下 Compose，然后按下并释放	然后按下并释放	结果
+	-	正负号 ±
-	-	软连字符 -
-	:	除号 ÷
-	a	序号 (阴性) ^a
-	o	序号 (阳性) ^o
-	, (逗号)	非标记 ¬
.	.	中间点 ·
1	2	普通分数 ½
1	4	普通分数 ¼
3	4	普通分数 ¾
<	<	左双尖引号 «
>	>	右双尖引号 »
?	?	倒置问号 ¿
A	' (反引号)	带抑音符的 A À
A	' (单引号)	带锐音符的 A Á
A	nls=gb2312*	带上圆圈的 A Å
A		带分音符的 A Ä
A	^	带扬抑符的 A Â
A	~	带波浪号的 A Ã
A	E	双元音 AE Æ
C	, (逗号)	带下加符的 C Ç
C	o	版权所有标记 ©
D	-	大写 eth ð
E	' (反引号)	带抑音符的 E È
E	'	带锐音符的 E É
E		带分音符的 E Ê
E	^	带扬抑符的 E Ê
I	' (反引号)	带抑音符的 I Ì

表 5-2 常用的 Latin-1 书写键序列 (续)

按下 Compose，然后按下并释放	然后按下并释放	结果
I	'	带锐音符的 Í
I	"	带分音符的 Î
I	^	带抑扬符的 Î
L	-	英镑标记 £
N	~	带波浪号的 Ñ
O	` (反引号)	带抑音符的 Ò
O	'	带锐音符的 Ó
O	/	带斜杠的 Ø
O	"	带分音符的 Ö
O	^	带扬抑符的 Ô
O	~	带波浪号的 Õ
R	o	注册商标标记 ®
T	H	拉丁文字母 Thorn þ
U	` (反引号)	带抑音符的 Ù
U	'	带锐音符的 Ú
U		带分音符的 Û
U	^	带扬抑符的 Û
Y	'	带锐音符的 Ý
Y	-	日元标记 ¥
a	` (反引号)	带抑音符的 à
a	'	带锐音符的 á
a	nls=gb2312*	带上圆圈的 å
a		带分音符的 ä
a	~	带波浪号的 ã
a	^	带扬抑符的 â
a	e	双元音 æ
c	, (逗号)	带下加符的 ç
c	/	分标记 ¢

表 5-2 常用的 Latin-1 书写键序列 (续)

按下 Compose，然后按下并释放	然后按下并释放	结果
c	o	版权所有标记 ©
d	-	eth ð
e	' (反引号)	带抑音符的 e è
e	'	带锐音符的 e é
e		带分音符的 e ë
e	^	带扬抑符的 e ê
i	' (反引号)	带抑音符的 i ì
i	'	带锐音符的 i í
i		带分音符的 i ï
i	^	带扬抑符的 i î
n	~	带波浪号的 n ñ
o	' (反引号)	带抑音符的 o ò
o	'	带锐音符的 o ó
o	/	带斜杠的 o ø
o		带分音符的 o ö
o	^	带扬抑符的 o ô
o	~	带波浪号的 o õ
s	s	德语双 s ß，也称为清音 S
t	h	拉丁文字母 Thorn þ
u	' (反引号)	带抑音符的 u ù
u	'	带锐音符的 u ú
u		带分音符的 u ü
u	^	带扬抑符的 u û
y	'	带锐音符的 y ý
y		带分音符的 y ÿ
		破折条 ¡

下表列出常用的 Latin-2 书写键序列。

表 5-3 常用的 Latin-2 书写键序列

按下 Compose ，然后按下并释放	按下并释放	结果
k	k	拉丁文字母 kra
A	_	带长音符的 A
E	_	带长音符的 E
E	.	带上点的 E
G	,	带下加符的 G
I	_	带长音符的 I
I	~	带波浪号的 I
I	a	I Ogonek
K	,	带下加符的 K
L	,	带下加符的 L
N	,	带下加符的 N
O	_	带长音符的 O
R	,	带下加符的 R
T		带斜线的 T
U	~	带波浪号的 U
U	a	U Ogonek
U	_	带长音符的 U
N	N	拉丁文字母 Eng
a	_	带长音符的 a
e	_	带长音符的 e
e	.	带上点的 e
g	,	带下加符的 g
i	_	带长音符的 i
i	~	带波浪号的 i
i	a	i Ogonek
k	,	带下加符的 k
l	,	带下加符的 l
n	,	带下加符的 n

表 5-3 常用的 Latin-2 书写键序列 (续)

按下 Compose，然后按下并释放	按下并释放	结果
o	—	带长音符的 o
r	,	带下加符的 r
t		带斜线的 t
u	~	带波浪号的 u
u	a	u Ogonek
u	—	带长音符的 u
n	n	eng

下表列出常用的 Latin-3 书写键序列。

表 5-4 常用的 Latin-3 书写键序列

按下 Compose，然后按下并释放	按下并释放	结果
C	>	带抑扬符的 C
C	.	带上点的 C
G	>	带抑扬符的 G
G	.	带上点的 G
H	>	带抑扬符的 H
J	>	带抑扬符的 j
S	>	带抑扬符的 S
U	u	带短音符的 U
c	>	带抑扬符的 c
c	.	带上点的 c
g	>	带抑扬符的 g
g	.	带上点的 g
h	>	带抑扬符的 h
j	>	带抑扬符的 j
s	>	带抑扬符的 s
u	u	带短音符的 u

下表列出常用的 Latin-4 书写键序列。

表 5-5 常用的 Latin-4 书写键序列

按下 Compose，然后按下并释放	按下并释放	结果
k	k	拉丁文字母 kra
A	_	带长音符的 A
E	_	带长音符的 E
E	。	带上点的 E
G	、	带下加符的 G
I	_	带长音符的 I
I	~	带波浪号的 I
I	a	I Ogonek
K	、	带下加符的 K
L	、	带下加符的 L
N	、	带下加符的 N
O	_	带长音符的 O
R	、	带下加符的 R
T		带斜线的 T
U	~	带波浪号的 U
U	a	U Ogonek
U	_	带长音符的 U
N	N	拉丁文字母 Eng
a	_	带长音符的 a
e	_	带长音符的 e
e	。	带上点的 e
g	、	带下加符的 g
i	_	带长音符的 i
i	~	带波浪号的 i
i	a	i Ogonek
k	、	带下加符的 k

表 5-5 常用的 Latin-4 书写键序列 (续)

按下 Compose ，然后按下并释放	按下并释放	结果
l	˘	带下加符的 l
n	˘	带下加符的 n
o	–	带长音符的 o
r	˘	带下加符的 r
t		带斜线的 t
u	~	带波浪号的 u
u	ˆ	u Ogonek
u	–	带长音符的 u
n	n	eng

下表列出常用的 Latin-5 书写键序列。

表 5-6 常用的 Latin-5 书写键序列

按下 Compose ，然后按下并释放	按下并释放	结果
G	u	带短音符的 G
I	.	带上点的 I
g	u	带短音符的 g
i	.	无点的 i

下表列出常用的 Latin-9 书写键序列。

表 5-7 常用的 Latin-9 书写键序列

按下 Compose ，然后按下并释放	按下并释放	结果
o	e	连字 oe
O	E	连字 OE
Y	“	带分音符的 Y

如果您使用的键盘带有重音死键，请使用下列书写键序列。“dead_acute”以及 X11 注册键系统中的此类键名，如 X_dead_acute 等等在 /usr/openwin/include/X11/keysymdef.h 中给出。SunFA_Circum 以及 Sun 定义的 X11 键系统名称中的此类键名（如 SunXK_FA_Circum）在 /usr/openwin/include/X11/Sunkeysym.h 中给出。

表 5-8 基于重音死键的书写键序列

按下并释放	按下并释放	结果
dead_grave	空格	重音符
dead_acute	省略符	锐音符
dead_acute	空格	省略符
dead_diaeresis	双引号	分音符
dead_diaeresis	空格	分音符
dead_circumflex	空格	抑扬符
dead_circumflex	斜线	垂直线
dead_circumflex	0	程度标记
dead_circumflex	1	上标一
dead_circumflex	2	上标二
dead_circumflex	3	上标三
dead_circumflex	句点	中点
dead_circumflex	叹号	破折条
dead_circumflex	减号	长音符号
dead_circumflex	下划线	长音符号
dead_cedilla	逗号	下加符
dead_cedilla	减号	Not 标记
dead_tilde	空格	波浪号
dead_grave	A	带抑音符的 A
dead_acute	A	带锐音符的 A
dead_circumflex	A	带抑扬符的 A
dead_tilde	A	带波浪号的 A
dead_diaeresis	A	带分音符的 A
dead_grave	a	带抑音符的 a
dead_acute	a	带锐音符的 a
dead_circumflex	a	带抑扬符的 a
dead_tilde	a	带波浪号的 a
dead_diaeresis	a	带分音符的 a

表 5-8 基于重音死键的书写键序列 (续)

按下并释放	按下并释放	结果
dead_cedilla	C	带下加符的 C
dead_cedilla	c	带下加符的 c
dead_grave	E	带抑音符的 E
dead_acute	E	带锐音符的 E
dead_circumflex	E	带抑扬符的 E
dead_diaeresis	E	带分音符的 E
dead_grave	e	带抑音符的 e
dead_acute	e	带锐音符的 e
dead_circumflex	e	带抑扬符的 e
dead_diaeresis	e	带分音符的 e
dead_grave	I	带抑音符的 I
dead_acute	I	带锐音符的 I
dead_circumflex	I	带抑扬符的 I
dead_diaeresis	I	带分音符的 I
dead_grave	i	带抑音符的 i
dead_acute	i	带锐音符的 i
dead_circumflex	i	带抑扬符的 i
dead_diaeresis	i	带分音符的 i
dead_tilde	N	带波浪号的 N
dead_tilde	n	带波浪号的 n
dead_grave	O	带抑音符的 O
dead_acute	O	带锐音符的 O
dead_circumflex	O	带抑扬符的 O
dead_tilde	O	带波浪号的 O
dead_diaeresis	O	带分音符的 O
dead_grave	o	带抑音符的 o
dead_acute	o	带锐音符的 o
dead_circumflex	o	带抑扬符的 o
dead_tilde	o	带波浪号的 o

表 5-8 基于重音死键的书写键序列 (续)

按下并释放	按下并释放	结果
dead_diaeresis	o	带分音符的 o
dead_cedilla	S	带下加符的 S
dead_cedilla	s	带下加符的 s
dead_grave	U	带抑音符的 U
dead_acute	U	带锐音符的 U
dead_circumflex	U	带抑扬符的 U
dead_diaeresis	U	带分音符的 U
dead_grave	u	带抑音符的 u
dead_acute	u	带锐音符的 u
dead_circumflex	u	带抑扬符的 u
dead_diaeresis	u	带分音符的 u
dead_acute	Y	带锐音符的 Y
dead_acute	y	带锐音符的 y
dead_diaeresis	y	带分音符的 y
SunFA_Grave	空格	重音符
SunFA_Grave	A	带抑音符的 A
SunFA_Grave	a	带抑音符的 a
SunFA_Grave	E	带抑音符的 E
SunFA_Grave	e	带抑音符的 e
SunFA_Grave	I	带抑音符的 I
SunFA_Grave	i	带抑音符的 i
SunFA_Grave	O	带抑音符的 O
SunFA_Grave	o	带抑音符的 o
SunFA_Grave	U	带抑音符的 U
SunFA_Grave	u	带抑音符的 u
SunFA_Acute	省略符	锐音符
SunFA_Acute	空格	省略符
SunFA_Acute	A	带锐音符的 A
SunFA_Acute	a	带锐音符的 a

表 5-8 基于重音死键的书写键序列 (续)

按下并释放	按下并释放	结果
SunFA_Acute	C	带锐音符的 C
SunFA_Acute	c	带锐音符的 c
SunFA_Acute	E	带锐音符的 E
SunFA_Acute	e	带锐音符的 e
SunFA_Acute	I	带锐音符的 I
SunFA_Acute	i	带锐音符的 i
SunFA_Acute	L	带锐音符的 L
SunFA_Acute	l	带锐音符的 l
SunFA_Acute	N	带锐音符的 N
SunFA_Acute	n	带锐音符的 n
SunFA_Acute	O	带锐音符的 O
SunFA_Acute	o	带锐音符的 o
SunFA_Acute	R	带锐音符的 R
SunFA_Acute	r	带锐音符的 r
SunFA_Acute	S	带锐音符的 S
SunFA_Acute	s	带锐音符的 s
SunFA_Acute	U	带锐音符的 U
SunFA_Acute	u	带锐音符的 u
SunFA_Acute	Y	带锐音符的 Y
SunFA_Acute	y	带锐音符的 y
SunFA_Acute	Z	带锐音符的 Z
SunFA_Acute	z	带锐音符的 z
SunFA_Cedilla	逗号	下加符
SunFA_Cedilla	减号	Not 标记
SunFA_Cedilla	C	带下加符的 C
SunFA_Cedilla	c	带下加符的 c
SunFA_Cedilla	G	带下加符的 G
SunFA_Cedilla	g	带下加符的 g
SunFA_Cedilla	K	带下加符的 K

表 5-8 基于重音死键的书写键序列 (续)

按下并释放	按下并释放	结果
SunFA_Cedilla	k	带下加符的 k
SunFA_Cedilla	L	带下加符的 L
SunFA_Cedilla	l	带下加符的 l
SunFA_Cedilla	N	带下加符的 N
SunFA_Cedilla	n	带下加符的 n
SunFA_Cedilla	R	带下加符的 R
SunFA_Cedilla	r	带下加符的 r
SunFA_Cedilla	S	带下加符的 S
SunFA_Cedilla	s	带下加符的 s
SunFA_Cedilla	T	带下加符的 T
SunFA_Cedilla	t	带下加符的 t
SunFA_Circum	空格	抑扬符
SunFA_Circum	0	程度标记
SunFA_Circum	1	上标一
SunFA_Circum	2	上标二
SunFA_Circum	3	上标三
SunFA_Circum	叹号	破折条
SunFA_Circum	减号	长音符号
SunFA_Circum	下划线	长音符号
SunFA_Circum	句点	中点
SunFA_Circum	斜线	垂直线
SunFA_Circum	A	带抑扬符的 A
SunFA_Circum	a	带抑扬符的 a
SunFA_Circum	C	带抑扬符的 C
SunFA_Circum	c	带抑扬符的 c
SunFA_Circum	E	带抑扬符的 E
SunFA_Circum	e	带抑扬符的 e
SunFA_Circum	G	带抑扬符的 G
SunFA_Circum	g	带抑扬符的 g

表 5-8 基于重音死键的书写键序列 (续)

按下并释放	按下并释放	结果
SunFA_Circum	H	带抑扬符的 H
SunFA_Circum	h	带抑扬符的 h
SunFA_Circum	I	带抑扬符的 I
SunFA_Circum	i	带抑扬符的 i
SunFA_Circum	J	带抑扬符的 J
SunFA_Circum	j	带抑扬符的 j
SunFA_Circum	O	带抑扬符的 O
SunFA_Circum	o	带抑扬符的 o
SunFA_Circum	S	带抑扬符的 S
SunFA_Circum	s	带抑扬符的 s
SunFA_Circum	U	带抑扬符的 U
SunFA_Circum	u	带抑扬符的 u
SunFA_Diaeresis	双引号	分音符
SunFA_Diaeresis	空格	分音符
SunFA_Diaeresis	A	带分音符的 A
SunFA_Diaeresis	a	带分音符的 a
SunFA_Diaeresis	E	带分音符的 E
SunFA_Diaeresis	e	带分音符的 e
SunFA_Diaeresis	I	带分音符的 I
SunFA_Diaeresis	i	带分音符的 i
SunFA_Diaeresis	O	带分音符的 O
SunFA_Diaeresis	o	带分音符的 o
SunFA_Diaeresis	U	带分音符的 U
SunFA_Diaeresis	u	带分音符的 u
SunFA_Diaeresis	y	带分音符的 y
SunFA_Diaeresis	Y	带分音符的 Y
SunFA_Tilde	空格	波浪号
SunFA_Tilde	A	带波浪号的 A
SunFA_Tilde	a	带波浪号的 a

表 5-8 基于重音死键的书写键序列 (续)

按下并释放	按下并释放	结果
SunFA_Tilde	N	带波浪号的 N
SunFA_Tilde	n	带波浪号的 n
SunFA_Tilde	O	带波浪号的 O
SunFA_Tilde	o	带波浪号的 o

阿拉伯语输入模式

要切换到阿拉伯语输入模式，请按下 **Compose a r**，或者从输入模式选择窗口中选择阿拉伯语。有关访问输入模式选择窗口的信息，请参见第 111 页“访问输入法”。

下图显示阿拉伯语键盘布局。

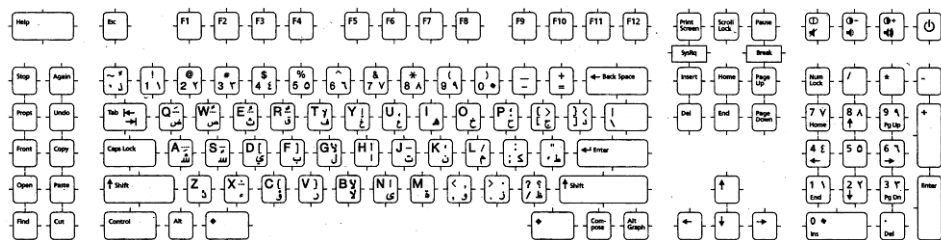


图 5-2 阿拉伯语键盘

西里尔语输入模式

要切换到西里尔语输入模式，请按下 **Compose c c**，或者从输入模式选择窗口中选择西里尔语。有关访问输入模式选择窗口的信息，请参见第 111 页“访问输入法”。

下图显示西里尔语（俄语）键盘布局。

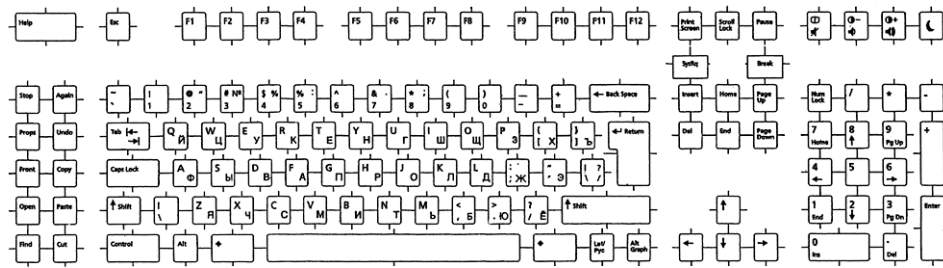


图 5-3 西里尔语（俄语）键盘

在切换到西里尔语输入模式后，您不能输入英语或欧洲语言文本。要切换回英语/欧洲语言输入模式，请同时键入 **Ctrl-空格键**，或者通过单击状态区从输入模式选择窗口中选择英语/欧洲语言输入模式。请参见第 111 页“访问输入法”。

您也可以通过键入相应的输入模式切换键序列切换到其他输入模式中。

希腊语输入模式

要切换到希腊语输入模式，请按下 **Compose g g**，或者从输入模式选择窗口中选择希腊语。有关访问输入模式选择窗口的信息，请参见第 111 页“访问输入法”。

在切换到希腊文输入模式后，您不能输入英语或欧洲语言文本。要切换回英语/欧洲语言输入模式，请同时键入 **Ctrl** 键和空格键，或者通过单击状态区从输入模式选择窗口中选择英语/欧洲语言输入模式。下图显示希腊语欧式键盘布局。

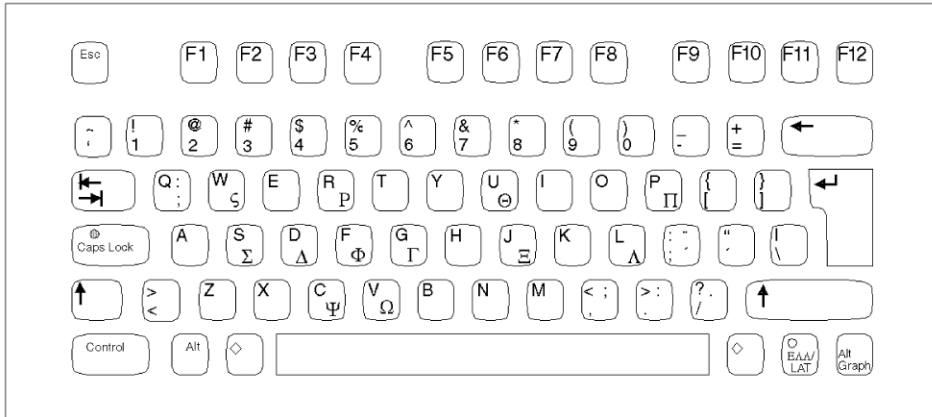


图 5-4 希腊语欧式键盘

下图显示希腊语 UNIX 键盘。

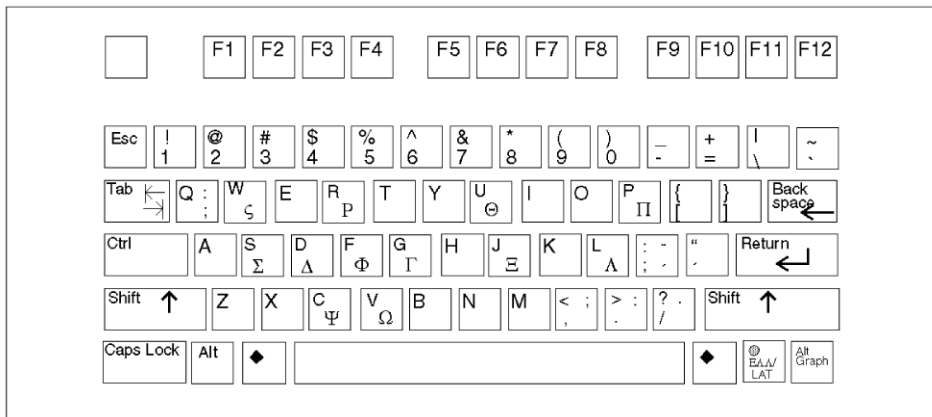


图 5-5 希腊语 UNIX 键盘

希腊语输入模式中支持下列书写键序列。部分书写键序列以重音死键开头。缩写“ordfemenine”代表阴性序数指示键。

表 5-9 希腊语输入模式的书写键序列

按下并释放	按下并释放	结果
分号	a	带 tonos 的小写 Greek_alpha

表 5-9 希腊语输入模式的书写键序列 (续)

按下并释放	按下并释放	结果
分号	e	带 tonos 的小写 Greek_epsilon
分号	h	带 tonos 的小写 Greek_eta
分号	i	带 tonos 的小写 Greek_iota
分号	o	带 tonos 的小写 Greek_omicron
分号	y	带 tonos 的小写 Greek_upsilon
分号	v	带 tonos 的小写 Greek_omega
分号	A	带 tonos 的大写 Greek_alpha
分号	E	带 tonos 的大写 Greek_epsilon
分号	H	带 tonos 的大写 Greek_eta
分号	I	带 tonos 的大写 Greek_iota
分号	O	带 tonos 的大写 Greek_omicron
分号	Y	带 tonos 的大写 Greek_upsilon
分号	V	带 tonos 的大写 Greek_omega
dead_acute	Greek_alpha	带 tonos 的小写 Greek_alpha
dead_acute	Greek_epsilon	带 tonos 的小写 Greek_epsilon
dead_acute	Greek_eta	带 tonos 的小写 Greek_eta
dead_acute	Greek_iota	带 tonos 的小写 Greek_iota
dead_acute	Greek_omicron	带 tonos 的小写 Greek_omicron
dead_acute	Greek_upsilon	带 tonos 的小写 Greek_upsilon
dead_acute	Greek_omega	带 tonos 的小写 Greek_omega
dead_acute	Greek_ALPHA	带 tonos 的大写 Greek_alpha
dead_acute	Greek_EPSILON	带 tonos 的大写 Greek_epsilon
dead_acute	Greek_ETA	带 tonos 的大写 Greek_eta
dead_acute	Greek_IOTA	带 tonos 的大写 Greek_iota

表 5-9 希腊语输入模式的书写键序列 (续)

按下并释放	按下并释放	结果
dead_acute	Greek_OMICRON	带 tonos 的大写 Greek_omicron
dead_acute	Greek_UPSILON	带 tonos 的大写 Greek_upsilon
dead_acute	Greek_OMEGA	带 tonos 的大写 Greek_omega
dead_acute	a	带 tonos 的小写 Greek_alpha
dead_acute	e	带 tonos 的小写 Greek_epsilon
dead_acute	h	带 tonos 的小写 Greek_eta
dead_acute	i	带 tonos 的小写 Greek_iota
dead_acute	o	带 tonos 的小写 Greek_omicron
dead_acute	y	带 tonos 的小写 Greek_upsilon
dead_acute	v	带 tonos 的小写 Greek_omega
dead_acute	A	带 tonos 的大写 Greek_alpha
dead_acute	E	带 tonos 的大写 Greek_epsilon
dead_acute	H	带 tonos 的大写 Greek_eta
dead_acute	I	带 tonos 的大写 Greek_iota
dead_acute	O	带 tonos 的大写 Greek_omicron
dead_acute	Y	带 tonos 的大写 Greek_upsilon
dead_acute	V	带 tonos 的大写 Greek_omega
冒号	i	带 dialytika 的小写 Greek_iota
冒号	y	带 dialytika 的小写 Greek_upsilon
冒号	I	带 dialytika 的大写 Greek_iota
冒号	Y	带 dialytika 的大写 Greek_upsilon
dead_diaeresis	i	带 dialytika 的小写 Greek_iota
dead_diaeresis	y	带 dialytika 的小写 Greek_upsilon

表 5-9 希腊语输入模式的书写键序列 (续)

按下并释放	按下并释放	结果
dead_diaeresis	I	带 dialytika 的大写 Greek_iota
dead_diaeresis	Y	带 dialytika 的大写 Greek_upsilon
dead_diaeresis	Greek_iota	带 dialytika 的小写 Greek_iota
dead_diaeresis	Greek_upsilon	带 dialytika 的小写 Greek_upsilon
dead_diaeresis	Greek_IOTA	带 dialytika 的大写 Greek_iota
dead_diaeresis	Greek_UPSILON	带 dialytika 的大写 Greek_upsilon
分号	分号	希腊语 tonos
冒号	冒号	分音符/dialytika
ordfeminine	0	加号-减号
ordfeminine	1	段落标记
ordfeminine	2	上标二
ordfeminine	3	上标三
ordfeminine	5	破折条
ordfeminine	6	版权所有标记
ordfeminine	7	Not 标记
ordfeminine	8	软连字符
ordfeminine	9	程度标记
ordfeminine	连字符	普通分数二分之一
ordfeminine	反斜线	英镑标记
ordfeminine	左括号	修饰符反逗号
ordfeminine	右括号	修饰符字母省略符
ordfeminine	左括号	左双尖引号
ordfeminine	右括号	右双尖引号
SunFA_Acute	a	带 tonos 的小写 Greek_alpha
SunFA_Acute	e	带 tonos 的小写 Greek_epsilon
SunFA_Acute	h	带 tonos 的小写 Greek_eta
SunFA_Acute	i	带 tonos 的小写 Greek_iota

表 5-9 希腊语输入模式的书写键序列 (续)

按下并释放	按下并释放	结果
SunFA_Acute	o	带 tonos 的小写 Greek_omicron
SunFA_Acute	y	带 tonos 的小写 Greek_upsilon
SunFA_Acute	v	带 tonos 的 Greek_omega
SunFA_Acute	A	带 tonos 的大写 Greek_alpha
SunFA_Acute	E	带 tonos 的大写 Greek_epsilon
SunFA_Acute	H	带 tonos 的大写 Greek_eta
SunFA_Acute	O	带 tonos 的大写 Greek_omicron
SunFA_Acute	I	带 tonos 的大写 Greek_iota
SunFA_Acute	Y	带 tonos 的大写 Greek_upsilon
SunFA_Acute	V	带 tonos 的大写 Greek_omega
SunFA_Acute	Greek_alpha	带 tonos 的小写 Greek_alpha
SunFA_Acute	Greek_epsilon	带 tonos 的小写 Greek_epsilon
SunFA_Acute	Greek_eta	带 tonos 的小写 Greek_eta
SunFA_Acute	Greek_iota	带 tonos 的小写 Greek_iota
SunFA_Acute	Greek_omega	带 tonos 的小写 Greek_omega
SunFA_Acute	Greek_omicron	带 tonos 的小写 Greek_omicron
SunFA_Acute	Greek_upsilon	带 tonos 的小写 Greek_upsilon
SunFA_Acute	Greek_ALPHA	带 tonos 的大写 Greek_alpha
SunFA_Acute	Greek_EPSILON	带 tonos 的大写 Greek_epsilon
SunFA_Acute	Greek_ETA	带 tonos 的大写 Greek_eta
SunFA_Acute	Greek_IOTA	带 tonos 的大写 Greek_iota
SunFA_Acute	Greek_OMICRON	带 tonos 的大写 Greek_omicron
SunFA_Acute	Greek_UPSILON	带 tonos 的大写 Greek_upsilon
SunFA_Acute	Greek_OMEGA	带 tonos 的大写 Greek_omega

表 5-9 希腊语输入模式的书写键序列 (续)

按下并释放	按下并释放	结果
SunFA_Diaeresis	i	带 dialytika 的小写 Greek_iota
SunFA_Diaeresis	y	带 dialytika 的小写 Greek_upsilon
SunFA_Diaeresis	I	带 dialytika 的大写 Greek_iota
SunFA_Diaeresis	Y	带 dialytika 的大写 Greek_upsilon
SunFA_Diaeresis	Greek_iota	带 dialytika 的小写 Greek_iota
SunFA_Diaeresis	Greek_upsilon	带 dialytika 的小写 Greek_upsilon
SunFA_Diaeresis	Greek_IOTA	带 dialytika 的大写 Greek_iota
SunFA_Diaeresis	Greek_UPSILON	带 dialytika 的大写 Greek_upsilon

表 5-10 使用三个键的希腊语输入模式的书写键序列

按下并释放	按下并释放	按下并释放	结果
分号	冒号	y	带 dialytika 和 tonos 的小写 Greek_upsilon
冒号	分号	y	带 dialytika 和 tonos 的小写 Greek_upsilon
分号	冒号	i	带 dialytika 和 tonos 的小写 Greek_iota
冒号	分号	i	带 dialytika 和 tonos 的小写 Greek_iota
dead_acute	dead_diaeresis	y	带 dialytika 和 tonos 的小写 Greek_upsilon
dead_diaeresis	dead_acute	y	带 dialytika 和 tonos 的小写 Greek_upsilon
dead_acute	dead_diaeresis	i	带 dialytika 和 tonos 的小写 Greek_iota
dead_diaeresis	dead_acute	i	带 dialytika 和 tonos 的小写 Greek_iota
dead_acute	dead_diaeresis	Greek_upsilon	带 dialytika 和 tonos 的小写 Greek_upsilon

表 5-10 使用三个键的希腊语输入模式的书写键序列 (续)

按下并释放	按下并释放	按下并释放	结果
dead_diaeresis	dead_acute	Greek_upsilon	带 dialytika 和 tonos 的小写 Greek_upsilon
dead_acute	dead_diaeresis	Greek_iota	带 dialytika 和 tonos 的小写 Greek_iota
dead_diaeresis	dead_acute	Greek_iota	带 dialytika 和 tonos 的小写 Greek_iota
SunFA_Acute	SunFA_Diaeresis	i	带 dialytika 和 tonos 的小写 Greek_iota
SunFA_Diaeresis	SunFA_Acute	i	带 dialytika 和 tonos 的小写 Greek_iota
SunFA_Acute	SunFA_Diaeresis	y	带 dialytika 和 tonos 的小写 Greek_upsilon
SunFA_Diaeresis	SunFA_Acute	y	带 dialytika 和 tonos 的小写 Greek_upsilon
SunFA_Acute	SunFA_Diaeresis	Greek_iota	带 dialytika 和 tonos 的小写 Greek_iota
SunFA_Diaeresis	SunFA_Acute	Greek_iota	带 dialytika 和 tonos 的小写 Greek_iota
SunFA_Acute	SunFA_Diaeresis	Greek_upsilon	带 dialytika 和 tonos 的小写 Greek_upsilon
SunFA_Diaeresis	SunFA_Acute	Greek_upsilon	带 dialytika 和 tonos 的小写 Greek_upsilon

表 5-11 使用四个键的希腊语输入模式的书写键序列

按下并释放	按下并释放	按下并释放	按下并释放	结果
分号	冒号	分号	冒号	希腊语 dialytika tonos
冒号	分号	冒号	分号	希腊语 dialytika tonos

希伯来语输入模式

要切换到希伯来语输入模式，请按下 **Compose h h**，或者从输入模式选择窗口中选择希伯来语。有关访问输入模式选择窗口的信息，请参见第 111 页“访问输入法”。

下图显示希伯来语键盘布局。

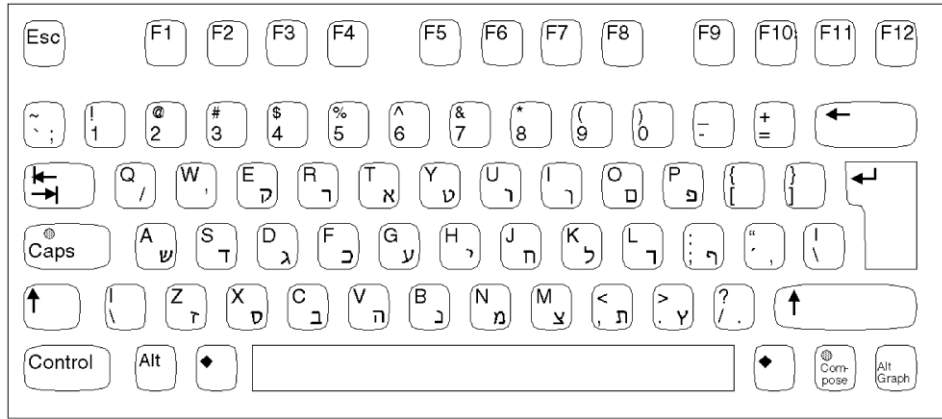


图 5-6 希伯来语键盘

日语输入模式

要切换到日语输入模式，请按下 `Compose ja`，或者从输入模式选择窗口中选择日语。有关访问输入模式选择窗口的信息，请参见第 111 页“访问输入法”。

要使用本地日语输入系统，需要安装一个或多个日语语言环境，然后重新引导系统。安装了日语语言环境后，您可以在所有 UTF-8 语言环境中使用 ATOK12。Wnn6 在 UTF-8 语言环境中不可用（`ja_JP.UTF-8` 除外）。

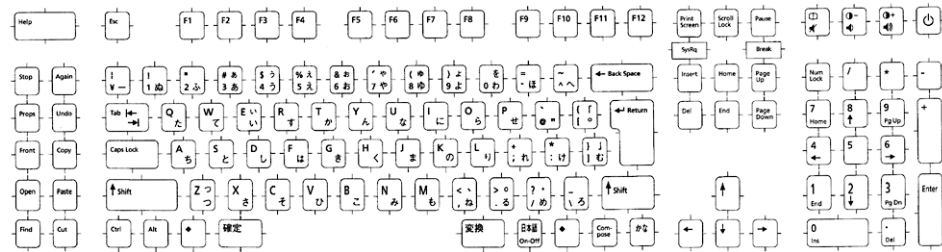


图 5-7 日语键盘

朝鲜语输入模式

要切换到朝鲜语输入模式，请按下 **Compose h h**，或者从输入模式选择窗口中选择朝鲜语。有关访问输入模式选择窗口的信息，请参见第 111 页“访问输入法”。

要使用本地朝鲜语输入系统，需要在系统上安装一个或多个朝鲜语语言环境。有关如何使用朝鲜语输入系统的详细信息，请参阅《*Korean Solaris User's Guide*》。

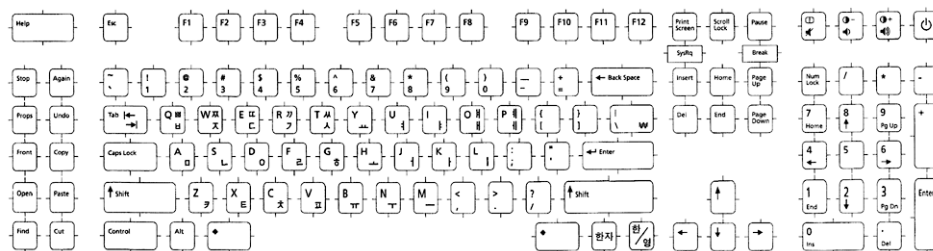


图 5-8 朝鲜语键盘

简体中文输入模式

要切换到简体中文输入模式，请按下 **Compose s c**，或者从输入模式选择窗口中选择简体中文。第 111 页“访问输入法”。

要使用本地简体中文输入系统，需要在系统上安装一个或多个简体中文语言环境。有关如何使用简体中文输入系统的详细信息，请参阅《*简体中文 Solaris 用户指南*》。

繁体中文输入模式

要切换到繁体中文输入模式，请按下 **Compose t c**，或者从输入模式选择窗口中选择繁体中文。有关访问输入模式选择窗口的信息，请参见第 111 页“访问输入法”。

要访问本地繁体中文输入系统，需要在系统上安装一个或多个繁体中文语言环境。有关如何使用繁体中文输入系统的详细信息，请参阅《*繁体中文 Solaris 使用者指南*》。

繁体中文（中国香港特别行政区）输入模式

要切换到繁体中文输入模式，请按下 **Compose h k**，或者从输入模式选择窗口中选择繁体中文（中国香港特别行政区）。有关访问输入模式选择窗口的信息，请参见第 111 页“访问输入法”。

要访问本地繁体中文（中国香港特别行政区）输入系统，需要在系统上安装一个或多个繁体中文（中国香港特别行政区）语言环境。

Unicode 十六进制输入模式

要切换到 Unicode 十六进制代码输入模式，请按下 **Compose u h**，或者从输入模式选择窗口中选择 Unicode 十六进制。要切换到八进制数字系统，请按下 **Compose u o** 或者选择 Unicode 八进制。有关访问输入模式选择窗口的信息，请参见第 111 页“访问输入法”。

要使用这些输入模式，您需要了解字符的十六进制或八进制代码点值。有关代码点值和字符之间的映射，请参阅《Unicode 标准，版本 4.0》。

如果您正处于 Unicode 十六进制代码输入模式，则要输入一个字符，需要键入四位十六进制数字。以下是一些十六进制值样例：

- 00A1 表示倒置叹号
- 03B2 表示希腊语小写字母 Beta
- AC00 表示朝鲜语 Hangul 音节
- 30A1 表示日语片假名字母 A
- 4E58 表示统一汉字

您可以使用 A、B、C、D、E 和 F 的大写和小写字母形式来表示十六进制数字。如果您希望使用八进制数字系统而不是十六进制数字系统，可以输入八进制数字 0 至 7。如果输入了一两位错误数字，可以使用 **Delete** 或 **Backspace** 键删除相应数字。

查表输入模式

要切换到查表输入模式，请按下 **Compose ll**，或者从输入模式选择窗口中选择查表。有关访问输入模式选择窗口的信息，请参见第 111 页“访问输入法”。

第二个查找窗口中显示仅按组显示的候选项，一次最多显示 80 个候选项。按下 **Control n** 转到下一组候选项，或者按下 **Control p** 转到上一组候选项。

系统环境

本部分描述语言环境环境变量、TTY 环境设置、32 位和 64 位 STREAMS 模块以及终端支持。

语言环境环境变量

确保在系统上安装 `en_US.UTF-8` 语言环境。要检查各种类中的当前语言环境设置，请使用 `locale` 实用程序。

```
system% locale
LANG=en_US.UTF-8
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_ALL=
```

要使用 `en_US.UTF-8` 语言环境桌面环境，请首先选择该语言环境。在 TTY 环境中，通过将 `LANG` 环境变量设置为 `en_US.UTF-8` 首先选择语言环境，如下面的 C-shell 实例所示：

```
system% setenv LANG en_US.UTF-8
```

确保 `LC_ALL`、`LC_COLLATE`、`LC_CTYPE`、`LC_MESSAGES`、`LC_NUMERIC`、`LC_MONETARY` 和 `LC_TIME` 种类未设置，或者设置为 `en_US.UTF-8`。如果设置了这些种类中的任何一个，则它们将覆盖较低优先级的 `LANG` 环境变量。有关环境变量分层结构的详细信息，请参见 `setlocale(3C)` 手册页。

您也可以从 CDE 桌面上启动 `en_US.UTF-8` 环境。在 CDE 登录屏幕的 `Options -> Language` 菜单上，选择 `en_US.UTF-8`。

TTY 环境设置

根据所使用的终端和终端仿真器的不同，您可能需要将某些特定代码集的 STREAMS 模块推送到流上。

有关 STREAMS 模块和一般流的详细信息，请参见《*STREAMS Programming Guide*》。

下表列出在终端环境中 `en_US.UTF-8` 语言环境支持的 64 位 STREAMS 模块。有关详细信息，请参见《*Solaris 64-bit Developer's Guide*》。

表 5-12 `en_US.UTF-8` 所支持的 STREAMS 模块

32 位 STREAMS 模块	说明
<code>/usr/kernel/strmod/sparcv9/u8lat1</code>	UTF-8 和 ISO8859-1（西欧语言）之间的代码转换 STREAMS 模块
<code>/usr/kernel/strmod/sparcv9/u8lat2</code>	UTF-8 和 ISO8859-2（东欧语言）之间的代码转换 STREAMS 模块
<code>/usr/kernel/strmod/sparcv9/u8koi8</code>	UTF-8 和 KOI8-R（西里尔语）之间的代码转换 STREAMS 模块

注意 – 从 Solaris 10 发行版开始，SPARC sun4u 平台不再支持 32 位内核。表 5-12 仅适用于 x86 平台的 32 位内核。有关详细信息，请参阅发行说明。

下表列出 `en_US.UTF-8` 支持的 64 位 STREAMS 模块。

表 5-13 `en_US.UTF-8` 所支持的 64 位 STREAMS 模块

64 位 STREAMS 模块	说明
<code>/usr/kernel/strmod/sparcv9/u8lat1</code>	UTF-8 和 ISO8859-1（西欧语言）之间的代码转换 STREAMS 模块
<code>/usr/kernel/strmod/sparcv9/u8lat2</code>	UTF-8 和 ISO8859-2（东欧语言）之间的代码转换 STREAMS 模块
<code>/usr/kernel/strmod/sparcv9/u8koi8</code>	UTF-8 和 KOI8-R（西里尔语）之间的代码转换 STREAMS 模块

▼ 如何装入 STREAMS 内核模块

1. 作为超级用户，请确定您运行的是 64 位 Solaris 还是 32 位 Solaris 系统。

```
system# isainfo -v
```

- 如果返回以下信息，则运行的是 64 位 Solaris 系统：

```
64-bit sparcv9 applications
32-bit sparc applications
```

- 如果返回以下信息，则运行的是 32 位 Solaris 系统：

```
32-bit sparc applications
```

- 如果返回以下信息，则运行的是 32 位 x86 系统：

```
32-bit i386 applications
```

2. 确定您的系统是否已装入 STREAMS 模块。

```
system# modinfo | grep modulename
```

如果已安装了 STREAMS 模块（如 `u8lat1`），则输出将类似于以下内容：

```
system# modinfo | grep u8lat1
89 ff798000 4b13 18 1 u8lat1 (UTF-8 <--> ISO 8859-1 module)
```

3. 如果该模块尚未装入，请使用 `modload(1M)` 命令装入它。

- 在 32 位系统上，键入：

```
system# modload /usr/kernel/strmod/u8lat1
```

- 在 64 位系统上，键入：

```
system# modload /usr/kernel/strmod/sparcv9/u8lat1
```

适当的 `u8lat1` STREAMS 模块已装入内核中。此时您可以将其推送到流中了。

▼ 如何卸载 STREAMS 内核模块

1. 作为超级用户，确认已装入内核模块。

例如，要确认已装入 `u8lat1`，应键入：

```
system# modinfo | grep u8lat1
89 ff798000 4b13 18 1 u8lat1 (UTF-8 <--> ISO 8859-1 module)
```

2. 使用 `modunload(1M)` 命令卸载内核。

例如，要卸载 `u8lat1` 模块，应键入：

```
system# modunload -i 89
```

▼ 如何设置 Latin-2 终端和 STREAMS 模块

1. 使用第二个命令行中所示的 `strchg(1M)`。

```
system% cat > tmp/mystreams
ttcompat
ldterm
u8lat1
ptem
```

```
^D
system% strchg -f /tmp/mystreams
确保在使用 strchg(1) 时您是超级用户或此设备的属主。
```

2. 要查看当前配置，请运行 **strconf** 命令。

```
system% strconf
ttcompat
ldterm
u8lat1
ptem
pts
system%
```

3. 要复位初始配置，请运行 **strchg** 命令。

```
system% cat > /tmp/orgstreams
ttcompat
ldterm
ptem
^D
system% strchg -f /tmp/orgstreams
```

能够输入和输出 UTF-8 字符的 dtterm 和 xterm 终端

与 Solaris 操作系统以前的发行版不同，支持 UTF-8 代码集输入和输出的 dtterm 和 xterm 终端仿真器和任何其他终端在它们的流中无需附加任何的 STREAMS 模块。如果您使用 stty(1) 实用程序设置终端环境，ldterm 模块现在与代码集无关，并且支持 Unicode/UTF-8。

要为 Unicode 语言环境设置正确的终端环境，请使用 stty(1) 实用程序。

```
system% /bin/stty defeucw
```

要查询当前的设置，请使用 stty 实用程序的 -a 选项，如下所示：

```
system% /bin/stty -a
```

注意 - 由于 /usr/ucb/stty 未经国际化，因此请改用 /bin/stty。

Latin-1、Latin-2 或 KOI8-R 的终端支持

对于仅支持 Latin-1 (ISO8859-1)、Latin-2 (ISO8859-2) 或 KOI8-R 的终端，您应具有下列 STREAMS 配置：

```
head <-> ttcompat <-> ldterm <-> u8lat1 <-> TTY
```

此配置仅适用于支持 Latin-1 的终端。对于 Latin-2 终端，请将 STREAMS 模块 u8lat1 替换为 u8lat2。对于 KOI8-R 终端，请将该模块替换为 u8koi8。

确保已将 STREAMS 模块装入到了内核中。

将设置保存在 ~/.cshrc 中

假定必需的 STREAMS 模块已装入内核，可以在 .cshrc 文件中保存以下行（C shell 实例）以便于使用：

```
setenv LANG en_US.UTF-8
if ($?USER != 0 && $?prompt != 0) then
    cat >! /tmp/mystreams$$ << _EOF
    ttcompat
    ldtterm
    u8lat1
    ptem
_EOF
    /bin/strchg -f /tmp/mystreams$$
    /bin/rm -f /tmp/mystreams$$
    /bin/stty cs8 -istrip defeucw
endif
```

在 .cshrc 文件中有了这几行后，您就不必在每次使用 STREAMS 模块时都要键入所有这些命令了。请注意，第二个 _EOF 应从该文件的第一列开始。

代码转换

Unicode 语言环境通过 iconv 和 sdtconvtool 实用程序，增加了对在多个国家/地区的主要代码集之间进行各种代码转换的支持。

在当前 Solaris 环境中，实用程序 geniconvtbl 可启用用户定义的代码转换。使用 geniconvtbl 实用程序创建的用户定义代码转换可以同时用于 iconv(1) 和 iconv(3)。有关该实用程序的详细信息，请参阅 geniconvtbl(1) 和 geniconvtbl(4) 手册页。

附录 A 中的表中列出了可应用于 iconv、iconv_open 和 sdtconvtool 的 fromcode 和 tocode 名称。有关 iconv 代码转换的更多信息，请参见 iconv(1) 和 sdtconvtool(1) 手册页。有关可用代码转换的更多信息，请参见 iconv(5) 手册页。另请参见附录 A。

注意 – UCS-2、UCS-4、UTF-16 和 UTF-32 均为 Unicode/ISO/IEC 10646 表示形式，用于当 Unicode 4.0 和 ISO/IEC 10646-1:2000 标准定义的 Byte Order Mark (BOM) 字符出现在字符流开头时对这些字符加以区别。其他形式（如 UCS-2BE、UCS-4BE、UTF-16BE 和 UTF-32BE）为固定宽度的 Unicode/ISO/IEC 10646 表示形式，它们不识别 BOM 字符，并且同样采用大尾数法字节排序。其他表示形式（如 UCS-2LE、UCS-4LE、UTF-16LE 和 UTF-32LE）将采用小尾数法字节排序。这些形式也不识别 BOM 字符。

有关 ISO8859-* 和 KOI8-* 的关联脚本和语言的信息，请参见 <http://czyborra.com/charsets/iso8869.html>。

DtMail 支持

由于脚本的覆盖范围增加，因此在 en_US.UTF-8 语言环境中运行的 Solaris DtMail 支持下列字符集（按 MIME 名称指示）：

- US-ASCII (7 位的 US ASCII)
- UTF-8 (UCS 传输格式 8 位)
- UTF-7 (UCS 传输格式 7 位)
- ISO-8859-1 (拉丁语-1)
- ISO-8859-2 (拉丁语-2)
- ISO-8859-3 (拉丁语-3)
- ISO-8859-4 (拉丁语-4)
- ISO-8859-5 (拉丁文/西里尔语)
- ISO-8859-6 (拉丁文/阿拉伯语)
- ISO-8859-7 (拉丁文/希腊语)
- ISO-8859-8 (拉丁文/希伯来语)
- ISO-8859-9 (Latin-5)
- ISO-8859-10 (Latin-6)
- ISO-8859-13 (Latin-7/波罗的海语)
- ISO-8859-14 (Latin-8/凯尔特语)
- ISO-8859-15 (Latin-9)
- ISO-8859-16 (Latin-10)
- KOI8-R (西里尔语)
- ISO-2022-JP 和 EUC-JP (日语)
- ISO-2022-KR 和 EUC-KR (朝鲜语)
- ISO-2022-CN (简体中文)
- ISO-8859-13 (Latin-7/波罗的海语)
- ISO-8859-14 (Latin-8/凯尔特语)
- KOI8-U (西里尔语/乌克兰语)
- Shift_JIS (Shift JIS 中的日语)
- GB2312 (EUC 中的简体中文)

- TIS-620 (泰语)
- UTF-16 (UCS 传输格式 16 位)
- UTF-16BE (UTF-16 大尾数法)
- UTF-16LE (UTF-16 小尾数法)
- Windows-1250
- Windows-1251
- Windows-1252
- Windows-1253
- Windows-1254
- Windows-1255
- Windows-1256
- Windows-1257
- Windows-1258
- Big5 (繁体中文)
- UTF-32 (UCS 传输格式 32 位)
- UTF-32BE (UTF-32 大尾数法)
- UTF-32LE (UTF-32 小尾数法)

这种支持使用户能够在 `DtMail` 实例中查看以世界上任何地区的不同字符集编码的几乎任意种类的电子邮件。`DtMail` 通过查看 MIME 字符集以及随同电子邮件提供的内容传送编码来对收到的电子邮件进行解码。支持 Windows-125x MIME 字符集。

要发送电子邮件，您必须指定一个收件人邮件用户代理（邮件客户）能够理解的 MIME 字符集，也可以使用 `en_US.UTF-8` 语言环境提供的缺省 MIME 字符集。可以在“新建邮件”窗口中切换外发邮件的字符集，方法是按下 `Ctrl+Y`，或者单击“格式”菜单按钮，然后单击“更改字符集”按钮。下一个可用的字符集名称将显示在“发送”按钮顶部的左下角区域内。

如果您的电子邮件信息标题或信息正文包含无法由指定的 MIME 字符集表示的字符，则系统会自动将字符集切换为可表示任何字符的 UTF-8。

如果您的信息仅包含来自 7 位 US-ASCII 字符集的字符，则电子邮件的缺省 MIME 字符集将为 US-ASCII。任何邮件用户代理都可以对这种电子邮件信息进行解码，而不会丢失字符或信息。

如果您的信息包含来自各种语言的字符，则缺省 MIME 字符集将为 UTF-8。UTF-8 中的任何 8 位字符都是用 Quoted-Printable 编码来编码的。有关 MIME、已注册的 MIME 字符集和 Quoted-Printable 编码的详细信息，请参阅 RFC 2045、2046、2047、2048、2049、2279、2152、2237、1922、1557、1555 和 1489。



图 5-9 DtMail 的“新建邮件”窗口

程序设计环境

国际化的应用程序应自动启用 `en_US.UTF-8` 语言环境。但在应用程序的资源文件中需要有正确的 `FontSet/XmFontList` 定义。

有关国际化应用程序的信息，请参见《创建适用于全世界的软件：Solaris 国际开发者指南》第 2 版。

用于 X 应用程序的 FontSet

有关用于 X 应用程序的 `FontSet` 的信息，请参见第 108 页“Unicode 语言环境：`en_US.UTF-8` 支持”。

在 Solaris 桌面环境中，每个字符集都有一组关联字体。

以下是在当前 Solaris 环境中受支持的 Latin-1 字体的列表：

```
-dt-interface system-medium-r-normal-xxs sans utf-10-100-72-72-p-59-iso8859-1
-dt-interface system-medium-r-normal-xs sans utf-12-120-72-72-p-71-iso8859-1
-dt-interface system-medium-r-normal-s sans utf-14-140-72-72-p-82-iso8859-1
-dt-interface system-medium-r-normal-m sans utf-17-170-72-72-p-97-iso8859-1
-dt-interface system-medium-r-normal-l sans utf-18-180-72-72-p-106-iso8859-1
-dt-interface system-medium-r-normal-xl sans utf-20-200-72-72-p-114-iso8859-1
-dt-interface system-medium-r-normal-xxl sans utf-24-240-72-72-p-137-iso8859-1
```

有关 CDE 公共字体别名（包括 `-dt-interface user-*` 和 `-dt-application-*` 别名）的信息，请参见《*Common Desktop Environment: Internationalization Programmer's Guide*》。

在 `en_US.UTF-8` 语言环境中，`utf` 也作为 X 逻辑字体描述名的样式字段中的附加属性包括在该语言环境的公共字体别名中。因此，要有一组正确的字体，在字体集创建过程中就必须包括附加样式，如下例所示：

```
fs = XCreateFontSet(display,
"-dt-interface system-medium-r-normal-s*utf*",
&missing_ptr, &missing_count, &def_string);
```

CDE/Motif 应用程序中的 FontList 定义

与 `FontSet` 定义一样，应用程序的 `XmFontList` 资源定义也应包括该语言环境所支持的附加样式属性。

```
*fontList:\
-dt-interface system-medium-r-normal-s*utf*:
```


第 6 章

复杂文本布局

复杂文本布局 (CTL) 扩展使 Motif 的 API 能支持需要在逻辑文本表示和物理文本表示之间进行复杂转换的书写系统。阿拉伯语、希伯来语和泰语需要上述转换。CTL Motif 提供字符成形（如连写符、变音符号和段排序）功能。还提供静态和动态文本小部件转换支持，以及对动态文本小部件的双向文本和制表功能的支持。由于文本绘制是通过绘制层处理的，因此可以很容易地扩展其他小部件库以支持 CTL。

本章包含以下主题：

- 第 149 页 “CTL 技术概述”
- 第 150 页 “CTL 体系结构概述”
- 第 150 页 “对基于 X 库的应用程序的 CTL 支持”
- 第 150 页 “XOC 资源”
- 第 151 页 “支持 CTL 技术的 Motif 更改”
- 第 159 页 “开发 CTL 应用程序”

CTL 技术概述

要利用新功能，用户必须拥有可移植布局服务 (PLS) 库和适当的语言引擎。CTL 使用 PLS 作为语言引擎界面，并使用语言引擎转换文本，然后才能绘制文本。支持 CTL 的应用程序必须包含附加的资源，请见 CTL 文档中的说明。

特别地，XomCTL 支持基础语言环境相关的 PLS 模块转换提供的下列复杂符成形和重新排列功能：

- 位置变化
- 连接（多对一）和字符编写（一对多）
- 音调符号
- 双向性
- 对称交换
- 数字形状

- 字符串验证

CTL 体系结构概述

CTL 体系结构的组织如图 6-1 所示。栈顶部的 Dt Apps 使用 Motif CTL 功能来绘制文本。而 Motif 则使用 PLS 与特定语言环境的语言引擎交互，并执行变换以支持位置变化、数字成形等等。

CTL 体系结构使用特定于语言环境的引擎支持新语言。换句话说，无须改变 Motif 或 Dt Apps 即可添加对泰语和越南语的支持。

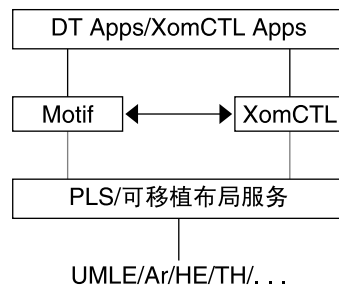


图 6-1 CTL 体系结构

对基于 X 库的应用程序的 CTL 支持

XomCTL (X 库输出模块中的复杂文本布局支持) 允许所有的纯粹 X Windows 应用程序 (如基于 X 的终端仿真器) 具有 CTL 支持。XomCTL 提供了包括 X11 哑字体支持在内的具有所有功能的 Open Source XI18N 实现。

XOC 资源

当前 Solaris 环境中提供下列 XOC 资源：

XNText

使用户能够设置需要在其中执行 CTL 操作的文本缓冲区

XNTextLayoutNumGlyphs	为文本缓冲区中的文本提供字型数目
XNTextLayoutModifier	与 XmNLayoutModifier 或 Motif 相同
XNTextLayoutProperty	与 PLS 特性相同（输入至输出和输出至输入）
XNTextLayoutMapInpToOut	与 PLS 特性相同（输入至输出和输出至输入）
XNTextLayoutMapOutToInp	与 PLS 特性相同（输入至输出和输出至输入）

这些资源的说明可以从 X/Open 或 PLS 可移植布局服务的规范中获得。

支持 CTL 技术的 Motif 更改

下列 Motif 更改支持 CTL 技术。

XmNlayoutDirection	控制对象布局
XmStringDirection	指定系统显示字符串中字符的方向
XmRendition	将新的伪资源添加到 XmRendition
XmText 和 XmTextField	影响与 XmRendition 关联的文本的布局行为
XmTextFieldGetLayoutModifier	返回绘制布局对象的布局修饰符字符串
XmTextGetLayoutModifier	返回与小部件关联的绘制的当前布局对象设置的值
XmTextFieldSetLayoutModifier	设置与其绘制关联的布局对象的布局修饰符值
XmTextSetLayoutModifier	修改与小部件关联的绘制的布局对象设置
XmStringDirectionCreate	创建复合字符串

XmNlayoutDirection 资源

XmNlayoutDirection 资源控制对象布局。此资源以如下所示的方式同 LayoutObject 的定向值进行交互。

有关 XmNlayoutDirection 的概述，尤其是对 XmStringDirection 和 XmNlayoutDirection 之间交互的说明，请参见 Motif 《程序员指南》（2.1 版）的 11.3 节。

确定布局方向

如果将 `XmNlayoutDirection` 指定为 `XmDEFAULT_DIRECTION`，则小部件的布局方向在创建时从主导伪 XOC 中进行设置。对于动态文本（`XmText` 和 `XmTextField`），则主导伪 XOC 是与用于小部件的 `XmRendition` 关联的伪 XOC。对于静态文本（`XmList`、`XmLabel` 和 `XmLabelG`），则布局方向从指定方向的第一个复合字符串组件中设置。此规范以下列两种方式之一发生：

- 组件类型为 `XmSTRING_COMPONENT_LAYOUT_PUSH` 或 `XmSTRING_COMPONENT_DIRECTION`。
- 组件类型为 `XmSTRING_COMPONENT_LOCALE_TEXT`、`XmSTRING_COMPONENT_WIDECHAR_TEXT` 或 `XmSTRING_COMPONENT_TEXT`，它们来自关联的 `XmRendition` 和 `LayoutObject`。

如果未将 `XmNlayoutDirection` 指定为 `XmDEFAULT_DIRECTION`，并且未在布局修饰符字符串中显式指定 `XmNlayoutModifier @ls orientation` 值，则 `XmNlayoutDirection` 值将传递到 XOC 及其 `LayoutObject`。

如果同时显式指定了 `XmNlayoutDirection` 和 `XmNlayoutModifier@ls orientation` 值，则行为将混合在一起。`XmNlayoutDirection` 控制小部件对象布局，而 `XmNlayoutModifier @ls orientation` 值控制布局变换。

请参见《CAE 规范：可移植布局服务：依赖于上下文的定向文本》（开放组：1997 年 2 月；ISBN 1-85912-142-X；文档号码 C616），了解对于处理依赖于上下文的双向文本转换的可移植功能的描述，这些功能是对现有 POSIX 语言环境模型的逻辑扩展。该文档适用于希望为复杂文本语言提供支持的系统和应用程序程序员。

XmStringDirection 资源

`XmStringDirection` 是用于指定系统显示字符串中字符方向的数据类型。

`XmNlayoutDirection` 资源为任何没有指定字符串方向的组件的复合字符串（`XmString`）设置缺省绘制方向。因此，要设置布局方向，需要为 `XmNlayoutDirection` 资源设置适当的值。不需要创建具有特定方向组件的复合字符串。当应用程序绘制 `XmString` 时，此应用程序应检查所创建的字符串是否具有显式方向（`XmStringDirection`）。如果没有方向组件，应用程序应检查当前小部件的 `XmNlayoutDirection` 资源的值并使用此值作为 `XmString` 的缺省绘制方向。

XmRendition 资源

CTL 将下表中列出的新的伪资源添加到 `XmRendition` 中。表之后是对伪资源的描述。

表 6-1 XmRendition 中的新资源

名称	类/类型	存取	缺省值
XmNfontType	XmCFontType/XmFontType	CSG	XmAS_IS
XmNlayoutAttrObject	XmClayoutAttrObject/String	CG	NULL
XmNlayoutModifier	XmClayoutModifier/String	CSG	NULL

XmNfontType

指定“绘制”字体对象的类型。对于 CTL，此资源的值必须是 XmFONT_IS_XOC 值。如果此值不匹配，则忽略 XmNlayoutAttrObject 和 XmNlayoutModifier 资源。

如果此资源的值为 XmFont_IS_XOC 且未指定 XmNfont 资源，则 XmNfontName 资源的值在创建时在 XmNlayoutAttrObject 资源指定的语言环境或当前语言环境中转换为 XOC 对象。此外，XmNlayoutModifier 资源的值被传递给与此 XOC 关联的任何布局对象。

XmNlayoutAttrObject

指定布局 AttrObject 参数。此资源用于创建与此 XOC（它与此 XmRendition 关联）关联的布局对象。有关此字符串的语法和语义，请参考布局服务 m_create_layout() 规范。有关“布局修饰符定向”输出值和 XmNlayoutDirection 小部件资源间交互作用的说明，请参见 XmNfontType 的描述。

XmNlayoutModifier

指定要传递给与此 XmRendition 的 XOC 一起使用的布局对象的布局值。有关该字符串的语法和语义，请参见《CAE 规范》。

使用 XmRendition{Retrieve,Update} 设置此资源会导致字符串传递给与此绘制的关联 XOC 关联的布局对象。该机制使您能动态配置布局服务。如果 Orientation、Context、TypeOfText、TextShaping 或 ShapeCharset 发生变更，则可能导致不可预测的行为。

附加布局行为

XmNlayoutModifier 影响与 XmRendition 关联的文本的布局行为。例如，如果数字的缺省布局处理是 NUMERALS_NOMINAL，则可以通过将 XmNlayoutModifier 设置为 @ls numerals=nominal:national 或 @ls numerals=:national 而更改为 NUMERALS_NATIONAL。

布局值可分为下列几个组：

- 编码说明—TypeOfText、TextShaping 和 ShapeCharset（以及语言环境编码集）

TypeOfText 从本质上而言是段排序，因此可以使用不透明块进行阐释。通过绘制对象动态修改这些值通常是无意义的，几乎肯定会导致不可预测的行为。

- 布局行为—Orientation、Context、ImplicitAlg、Swapping 和 Numerals。Orientation 和 Context 不应动态修改。您可以安全地修改 ImplicitAlg、Swapping 和 Numerals。
- 编辑行为—CheckMode

XmText 和 XmTextField 资源

Xm CTL 扩展了 XmText 和 XmTextField，方法是模仿 Motif 2.0 CStext 小部件，添加一组以可视方式运行的相似的移动和删除操作。标准 Motif 2.1 Text 和 TextField 不区分逻辑和物理顺序：下一个和向前意为“向右”，而上一个和向后意为“向左”。然而，CStext 进行了适当的区分并定义了一组具有严格物理名称的新操作（例如，left-character()、delete-right-word() 等等）。这些操作例程均被定义为对小部件的 XmNlayoutDirection 敏感，并可调用适当的 next- 或 previous- 操作。

Xm CTL 扩展比 CStext 扩展稍微复杂一些。Xm CTL 扩展对小部件的全局定向不敏感，但是对伪 XOC 所确定的光标周围物理字符的特定方向性（包括中性稳定性）敏感。

新资源名称使您能控制选择策略，提供绘制标签及控制对齐方式。

新 Xm CTL 操作集合大致是 {Move, Delete, Kill} 乘以 {Left, Right} 乘以 {Character, Word} 的向量积。下表中列出操作集。

表 6-2 Xm CTL 中的新资源

姓名	类/类型	存取	缺省值
XmNrenditionTag	XmCRenditionTag/XmRString	CSG	XmFONTLIST_DEFAULT_TAG
XmNalignment	XmCAlignment/XmRAlignment	CSG	XmALIGNMENT_BEGINNING
XmNeditPolicy	XmCEditPolicy/XmREditPolicy	CSG	XmEDIT_LOGICAL

XmNrenditionTag

指位于 XmNrenderTable 资源（用于小部件）中的 XmRendition 的绘制标签。

XmNalignment

指定此小部件中使用的文本对齐方式。仅支持 XmALIGNMENT_END 和 XmALIGNMENT_CENTER。

XmNeditPolicy

指定此小部件使用的编辑策略（XmEDIT_LOGICAL 或 XmEDIT_VISUAL）。如果指定 XmEDIT_VISUAL，则选择、光标移动和删除将以可视形式进行。设置该资源还可将标准键盘移动和删除事件的转换更改为新的“可视”操作列表或现有的逻辑操作。

字符定向操作例程

forward-cell() 和 backward-cell() 操作在指定的方向中查询字符的定向。如方向是从左向右，则这些操作调用相应的 next-/forward- 或 previous-/backward- 变式。

字符定向附加行为

这些操作通过使用布局服务变换 `OutToInp` 和 `Property` 缓冲区（对于嵌套级）来确定字符的方向。因而，小部件的行为取决于语言环境的特定变换。如果 `OutToInp`（尤其是 `Property` 缓冲区）中的信息不正确，则小部件可能会产生意外行为。此外，由于语言环境特定的模块处于该规范的范围之外，相同文本、应用程序、资源值和 `LayoutObject` 配置的双向编辑行为可能因平台的不同而不同。

可视模式操作导致显示基于单元的行为。逻辑模式操作导致基于字符的逻辑行为。例如，`delete-right-character()` 操作删除与显示单元相对应的输入缓冲区字符。即一个 `LayoutObject` 变换“特性”字节“新单元指示器”为 1 的输入缓冲区字符，以及“新单元指示器”为 0 的所有后继字符。

有关 `Property` 缓冲区的更多信息，请参见《CAE 规范》中的 `m_transform_layout()`。

同样，对于 `backward-character()`，插入点在输入缓冲区中向后移动一个字符，光标被在与关联的输出缓冲区字符相对应的可视位置重新绘制。因而，要进行若干次击键以在复合显示单元中移动。当插入点在“新单元指示器”为 0 的输入缓冲区字符（即音调符号或连字片段）中移动时，光标并不实际更改显示位置。

此行为意味着删除操作或者从逻辑/输入缓冲区端执行，或者从物理/输出端的显示单元级执行。没有严格的、物理的逐字符删除模式，因为在输入和输出缓冲区之间不存在一对一的对应关系。例如，给定的物理字符只能代表逻辑字符片段。

XmText 操作例程

下表描述 `XmText` 操作例程。

`left-character(extend)`

如果 `XmNeditPolicy` 为 `XmEDIT_LOGICAL` 且被不带参数调用，则插入光标在逻辑上向后移动一个字符。如果插入光标位于行首，则它将移动到上一行（如果存在）的最后一个逻辑字符。否则，插入光标位置不变。

如果 `XmNeditPolicy` 为 `XmEDIT_VISUAL`，则光标移动到光标位置的左侧。如果插入光标位于行首，则它将移动到上一行（如果存在）的结尾字符。

如果使用 `extend` 参数调用 `left-character()`，则插入光标将像没有参数时一样移动，并扩展当前所选内容。

`left-character()` 操作使用原因值 `XmCR_MOVING_INSERT_CURSOR` 生成对 `XmNmotionVerifyCallback` 过程的调用。如果使用 `extend` 参数进行调用，则此操作可以生成对 `XmNgainPrimaryCallback` 过程的调用。有关更多信息，请参见《*Motif* 程序员参考》中的回调说明。

`right-character(extend)`

如果 `XmNeditPolicy` 为 `XmEDIT_LOGICAL` 且不被带参数调用，则它将插入光标逻辑地向前移动一个字符。如果插入光标位于行的逻辑结尾处，则此操作将插入光标移动到下一行（如果存在）的逻辑开始处。

如果 `XmNeditPolicy` 为 `XmEDIT_VISUAL`，则光标移动到该光标位置的右侧。如果插入光标位于行尾，则它将插入光标移动到下一行（如果存在）的开始处。

如果使用 `extend` 参数进行调用，则 `XmNeditPolicy` 将像不带参数调用一样移动插入光标，并扩展当前所选内容。

`right-character()` 操作使用原因值 `XmCR_MOVING_INSERT_CURSOR` 生成对 `XmNmotionVerifyCallback` 过程的调用。如果使用 `extend` 参数进行调用，则此操作可以生成对 `XmNgainPrimaryCallback` 过程的调用。有关更多信息，请参见《*Motif 程序员参考*》中的回调说明。

`right-word(extend)`

如果 `XmNeditPolicy` 为 `XmEDIT_LOGICAL` 且被不带任何参数调用，则插入光标移动到逻辑后继字（如果存在）的逻辑开始处。否则，此光标移动到当前字的逻辑结尾处。如果插入光标位于行的逻辑结尾处或此行的最后一个逻辑字，则光标移动到下一行（如果存在）的第一个逻辑字；否则，光标移动到当前字的逻辑结尾处。

如果 `XmNeditPolicy` 为 `XmEDIT_VISUAL` 且被不带参数调用，则插入光标移动到行尾右侧或后面的第一个空格字符后的第一个非空格字符处。

如果使用 `extend` 参数进行调用，则就像不带参数调用一样移动插入光标，并扩展当前所选内容。

`left-word()` 操作使用原因值 `XmCR_MOVING_INSERT_CURSOR` 生成对 `XmNmotionVerifyCallback` 过程的调用。如果使用 `extend` 参数调用，则此操作可以生成对 `XmNgainPrimaryCallback` 过程的调用。有关更多信息，请参见《*Motif 程序员参考*》中的回调说明。

`delete-left-character()`

如果 `XmNeditPolicy` 为 `XmEDIT_LOGICAL`，则它等效于 `delete-previous-char()`。如果 `XmNeditPolicy` 为 `XmEDIT_VISUAL`，则在正常模式下，如果有一个非空选择，则它将删除此选择。否则，此操作删除插入光标左侧的字符。在添加模式下，如果有一个非空选择，光标未离开所选内容，并且 `XmNpendingDelete` 设置为真，则此操作删除此选择。否则，此操作删除插入光标左侧的字符，这可能会影响选择。

`delete-left-character()` 操作使用原因值 `XmCR_MODIFYING_TEXT_VALUE` 生成对 `XmNmodifyVerifyCallback` 过程的调用，并且使用原因值 `XmCR_VALUE_CHANGED` 生成对 `XmNvalueChangedCallback` 过程的调用。

`delete-right-character()`

如果 `XmNeditPolicy` 为 `XmEDIT_VISUAL`，则它等效于 `delete-next-character()`。如果 `XmNeditPolicy` 为 `XmEDIT_VISUAL`，则在正常模式下，如果有一个非空选择，则它删除此选择。否则，它删除插入光标右侧的字符。在添加模式中，如果有一个非空选择，并且光标未离开所选内容，则将 `XmNpendingDelete` 设置为真，并删除此选择；否则，删除插入光标右侧的字符。此操作可能会对选择造成影响。

`delete-right-character()` 操作使用原因值 `XmCR_MODIFYING_TEXT_VALUE` 生成对 `XmNmodifyVerify-Callback` 过程的调用，并且使用原因值 `XmCR_VALUE_CHANGED` 生成对 `XmNvalue-ChangedCallback` 过程的调用。

实现了一些基于单元的例程以支持字符组合、连字和音调符号。换句话说，两个或多个字符可以被占用一个表示单元的单个字形表示。

XmText 单元操作例程如下表所示。

backward-cell(extend)

将插入光标向后移动一个单元。如果 XmNeditPolicy 为 XmEDIT_LOGICAL，则插入光标移动到逻辑上位于当前单元之前的那个单元（如果存在）的开始处。否则，光标移动到当前单元的开始处。

如果 XmNeditPolicy 是 XmEDIT_VISUAL，则光标移动到光标左侧单元（如果存在）的开始处。prev-cell() 操作使用原因值 XmCR_MOVING_INSERT_CURSOR 生成对 XmNmotionVerifyCallback 过程的调用。如果使用 extend 参数进行调用，则此操作可以生成对 XmNgainPrimaryCallback 过程的调用。有关更多信息，请参阅《Motif 程序员参考》中的回调说明。

forward-cell(extend)

将插入光标移动到下一个逻辑单元（如果存在）的开始处。否则，此操作将光标移动到此单元的结尾处。如果 XmNeditPolicy 为 XmEDIT_LOGICAL，则光标向前移动一个单元。

如果 XmNeditPolicy 为 XmEDIT_VISUAL，则光标移动到此光标位置右侧单元的开始处（如果此单元存在）；否则，光标移动到当前单元的结尾处。forward-cell() 操作使用原因值 XmCR_MOVING_INSERT_CURSOR 生成对 XmNmotionVerifyCallback 过程的调用。如果使用 extend 参数进行调用，则此操作可以生成对 XmNgainPrimaryCallback 过程的调用。有关更多信息，请参阅《Motif 程序员参考》中的回调说明。

XmTextFieldGetLayoutModifier 资源

XmTextFieldGetLayoutModifier() 返回布局修饰符字符串，它反映了与其绘制关联的布局对象的状态。

XmTextFieldGetLayoutModifier() 的语法是：

```
#include <Xm/TextF.h>
字符串 XmTextFieldGetLayoutModifier(小部件 小部件)
```

XmTextFieldGetLayoutModifier() 存取与小部件关联的绘制的当前布局对象设置的值。当使用公用函数更改布局对象修饰符的值时，XmTextFieldGetLayoutModifier 函数返回布局对象的完整状态，而不仅仅返回更改后的值。

XmTextFieldGetLayoutModifier() 以字符串值的形式返回布局对象修饰符值。

XmTextGetLayoutModifier 资源

XmTextGetLayoutModifier() 返回布局修饰符字符串，它反映了与其绘制关联的布局对象的状态。

`XmTextGetLayoutModifier()` 的语法是：

```
#include <Xm/Text.h>
String XmTextGetLayoutModifier(小部件 小部件)
```

`XmTextGetLayoutModifier` 存取与小部件关联的绘制的当前布局对象设置的值。当使用公用函数更改布局对象修饰符值时，`XmTextGetLayoutModifier` 函数返回布局对象的完整状态，而不仅仅返回更改后的值。

`XmTextGetLayoutModifier` 以字符串值的形式返回布局对象修饰符值。

XmTextFieldSetLayoutModifier 资源

`XmTextFieldSetLayoutModifier()` 设置布局修饰符值，这会更改与其绘制关联的布局对象的行为。

`XmTextFieldSetLayoutModifier()` 的语法是：

```
#include <Xm/TextF.h> \
void XmTextFieldSetLayoutModifier(小部件 \
小部件字符串 layout_modifier)
```

`XmTextFieldSetLayoutModifier` 修改与小部件关联的绘制的布局对象设置。当布局对象修饰符值使用此公共函数设置时，只有输入参数中指定的属性被更改；其余属性保持不变。

XmTextSetLayoutModifier 资源

`XmTextSetLayoutModifier()` 设置布局修饰符值，这会更改与其绘制关联的布局对象的行为。

`XmTextSetLayoutModifier()` 的语法是：

```
#include <Xm/Text.h>
void XmTextSetLayoutModifier(小部件 小部件, 字符串 layout_modifier)
```

`XmTextSetLayoutModifier` 修改与小部件关联的绘制的布局对象设置。布局对象修饰符值使用此公共函数设置时，只有输入参数中指定的属性被更改；其余属性保持不变。

XmStringDirectionCreate 资源

`XmStringDirectionCreate` 用于创建复合字符串。

`XmTextSetLayoutModifier()` 的语法是：

```
#include <Xm/Xm.h>
XmString XmStringDirectionCreate(方向)
XmStringDirection 方向
```

XmStringDirectionCreate 创建一个具有单个组件（一个具有给定值的方向）的复合字符串。另一方面，XmNlayoutDirection 资源为任何不具有指定其方向的组件的复合字符串 (XmString) 设置缺省绘制方向。因此，要设置布局方向，需要为 XmNlayoutDirection 资源设置适当值。您不需要创建具有特定方向组件的复合字符串。

当应用程序绘制 XmString 时，此应用程序应检查所创建的字符串是否具有显式方向 (XmStringDirection)。如果没有向应用程序提供方向组件，则此应用程序应检查当前小部件的 XmNlayoutDirection 资源的值并将此值作为 XmString 的缺省绘制方向。

UIL 参数

下表显示了 UIL 参数名称和类型。

表 6-3 UIL

UIL 参数名称	参数类型
XmNlayoutAttrObject	String
XmNlayoutModifier	String
XmNrenditionTag	String
XmNalignment	Integer
XmNeditPolicy	Integer

开发 CTL 应用程序

下列几节说明了如何开发 CTL 应用程序。

控制布局方向

复合字符串的方向被存储起来，以便使数据结构对于描述从左到右的语言（如英语、西班牙语、法语和德语）的文本或者从右到左的语言（如希伯来语和阿拉伯语）的文本同样有用。在 Motif 应用程序中，可以使用 `VendorShell` 或 `MenuShell` 中的 `XmNlayoutDirection` 资源来设置布局方向。管理器和原始小部件（以及小配件）也具有 `XmNlayoutDirection` 资源。缺省值从具有相同资源的最近的祖先继承。

如果是 `XmText` 小部件，则必须指定垂直方向及水平方向。将 `layoutDirection` 设置为 `XmRIGHT_TO_LEFT` 将会使字符串方向成为从右到左，而光标则垂直向下移动。如果垂直方向重要且要求从顶到底对齐，确保指定 `XmRIGHT_TO_LEFT TOP TO BOTTOM`。本设置将组件布局首先指定为从右到左，然后从上到下，从而产生所需的行为。

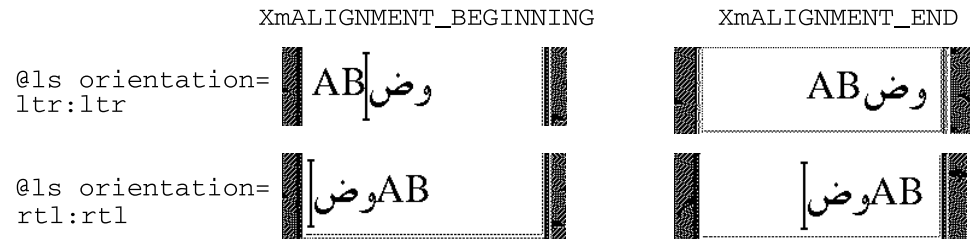
`XmText` 和 `TextField` 小部件的行为也受 `XmNalignment` 和 `XmNlayoutModifier` `XmRendition` 资源的影响。这些资源与 `XmNlayoutDirection` 共同控制文本小部件的布局行为。图 6-2 中阐释了此行为。

该图中使用的输入字符串是：

A B و ض

下图中 `XmNlayoutModifier` 字符串 `@ls orientation=` 设置值如左列中所示。

布局方向: XmLEFT_TO_RIGHT



布局方向: XmRIGHT_TO_LEFT

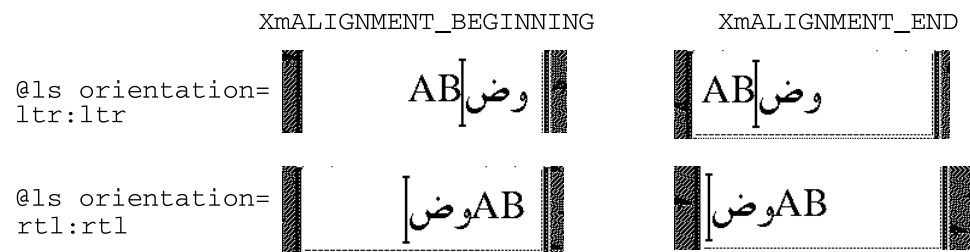


图 6-2 布局方向

如插图所示，XmNalignment 规定文本是否与布局方向一起右对齐或左对齐。XmNlayoutModifier 将文本分成多段，并根据定向值将它们从左到右或从右到左排列。换句话说，如果 XmNlayoutDirection 是 XmRIGHT_TO_LEFT，并且 XmNalignment 值是 XmALIGNMENT_BEGINNING，则此字符串右对齐。

实例 6-1 创建绘制

下列代码使用标记为“ArabicShaped”的“绘制”创建 XmNlabelString 为 XmCHARSET_TEXT 类型的 XmLabel。使用“ar”（对应于阿拉伯语言环境的语言环境名称）的 XmNlayoutAttrObject 以及布局修饰符字符串来创建“绘制”。该布局修饰符字符串为输出缓冲指定了 NUMERALS_CONTEXTUAL 的 Numerals 值和“iso8859-6”的 ShapeCharset 值。

特定于语言环境的布局模块将其输入文本转换到使用 16 位 Unicode 代码集编码的物理字符的输出缓冲区。由于已经指定了显式布局语言环境，因此此文本能够独立于运行时语言环境设置正确绘制。在此实例中，用 ISO 8859-6 对输入进行编码。

```
int n;
Arg args[10];
Widget w;
XmString labelString;
XmRendition rendition;
XmStringTag renditionTag;
XmRenderTable renderTable;
```

实例 6-1 创建绘制 (续)

```
/* alef lam baa noon taa - iso8859-6 */
labelString = XmStringGenerate("\307\344\310\346\312", NULL
                                XmCHARSET_TEXT, "ArabicShaped");
w = XtVaCreateManagedWidget("a label", xmLabelWidgetClass, parent,
                              XmNlabelString, labelString,
                              XmNlabelType, XmSTRING,
                              NULL);
n = 0;
XtSetArg(args[n], XmNfontName, "-*-medium-r-normal-*-24-*-*-*-*");
n++;
XtSetArg(args[n], XmNfontType, XmFONT_IS_XOC); n++;
XtSetArg(args[n], XmNlayoutAttrObject, "ar"); n++;
XtSetArg(args[n], XmNlayoutModifier,
          "@ls numerals=:contextual, shapecharset=iso8859-6"); n++;
renditionTag = (XmStringTag) "ArabicShaped";
rendition = XmRenditionCreate(w, renditionTag, args,
                              n);
renderTable =
    XmRenderTableAddRenditions(NULL, &rendition, 1, XmREPLACE_MERGE);
XtVaSetValues(w, XmNrenderTable, renderTable, NULL);
```

实例 6-2 编辑绘制

下列代码使用一个 `Rendition` 创建 `TextField` 小部件和 `RenderTable`。
`XmNlayoutAttrObject` 和 `XmNlayoutModifier` 伪资源均未指定，因此缺省为 `NULL`。此值意味着与“`Rendition`”关联的布局对象属于缺省语言环境（如果存在）。

要使此实例正常工作，该语言环境必须设置为代码集是 ISO 8859-6，并且特定于该语言环境的布局模块可支持 `IMPLICIT_BASIC` 算法。`Rendition` 的 `LayoutObject` 的 `ImplicitAlg` 值通过 `Rendition` 的 `XmNlayoutModifier` 伪资源进行修改。

```
int n;
Arg args[10];
Widget w;
    XmRendition rendition;
XmStringTag renditionTag;
XmRenderTable renderTable;
w = XmCreateTextField(parent, "text field", args, 0);
n = 0;
    XtSetArg(args[n], XmNfontName, "-*-medium-r-normal-*-24-*-*-*-*");
    n++;
    XtSetArg(args[n], XmNfontType, XmFONT_IS_XOC); n++;
renditionTag = (XmStringTag) "ArabicShaped";
rendition = XmRenditionCreate(w, renditionTag, args, n);
renderTable =
    XmRenderTableAddRenditions(NULL, &rendition, 1, XmREPLACE_MERGE);
XtVaSetValues(w, XmNrenderTable, renderTable, NULL);
    ....
n = 0;
XtSetArg(args[n], XmNlayoutModifier, "@ls implicitalg=basic");
n++;
```

实例 6-2 编辑绘制 (续)

```
XmRenditionUpdate(rendition, args, n);
```

在资源文件中创建绘制表

对于正确国际化的应用程序，应在其资源文件中指定 **Rendition** 和绘制表。在文件中指定了绘制表后，程序二进制代码就与给定语言环境的特定需要无关，从而能够轻松地进行定制以满足本地需要。

绘制表是用以下语法在资源文件中指定的：*resource_spec*: [*tag* [, *tag*] *]

其中 *tag* 是适用于某个绘制的 **XmNtag** 资源的某个字符串。

该行将按指定要求创建包含一个或多个绘制的初始绘制表。这些绘制被附加到指定的标记上：

```
resource_spec [*|.] rendition [*|.] resource_name:value
```

下例阐释了与 **XmRendition** 相关的 CTL 资源，这些资源可用资源文件来设置。**fontType** 必须设置为 **FONT_IS_XOC** 才能使布局对象生效。使用 **@ls** 指定的 **layoutModifier** 通过绘制对象传递给布局对象。

有关在布局对象中可以使用 **layoutModifier** 设置的资源的完整列表，请参见《CAE 规范：可移植布局服务：依赖于上下文的定向文本》。开放组：1997 年 2 月；ISBN 1-85912-142-X；文档号码 C616。

实例 6-3 在应用程序中创建绘制表

在创建绘制表之前，应用程序必须至少先创建一个属于该表的绘制。顾名思义，**XmRenderTableAddRenditions()** 函数还用于用新的绘制来增大绘制表。要创建新的绘制表，请使用 **NULL** 参数调用 **XmRenderTableAddRenditions()** 函数来替换现有的绘制表。

下列代码使用在将 **XmNfontType** 设置为 **XmFONT_IS_XOC** 时创建的绘制来创建一个绘制表。

```
int n;
Arg args[10];
Widget w;
XmString labelString;
XmRendition rendition;
XmStringTag renditionTag;
XmRenderTable renderTable;
/* alef lam baa noon taa - iso8859-6 */
labelString = XmStringGenerate("\307\344\310\346\312\", NULL
                               XmCHARSET_TEXT, "ArabicShaped");
w = XtVaCreateManagedWidget("a label", xmLabelWidgetClass, parent,
                             XmNlabelString, labelString,
                             XmNlabelType, XmSTRING,
```

实例 6-3 在应用程序中创建绘制表 (续)

```
        NULL);  
n = 0;  
XtSetArg(args[n], XmNfontName, "-*-medium-r-normal-*24-*-*-*-*-*");  
    n++;  
XtSetArg(args[n], XmNfontType, XmFONT_IS_XOC); n++;  
XtSetArg(args[n], XmNlayoutAttrObject, "ar"); n++;  
XtSetArg(args[n], XmNlayoutModifier,  
    "@ls numerals=nominal:contextual, shapecharset=iso8859-6"); n++;  
renditionTag = (XmStringTag) "ArabicShaped";  
rendition = XmRenditionCreate(w, renditionTag, args, n);  
renderTable =  
    XmRenderTableAddRenditions(NULL, &rendition, 1, XmREPLACE);  
XtVaSetValues(w, XmNrenderTable, renderTable, NULL);
```

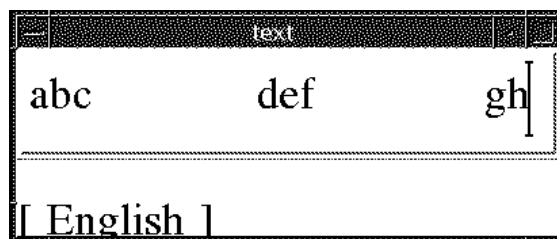
水平制表符

复合字符串可以包含控制文本位置的制表符字符。要在显示时解释这些字符，小部件实际上为复合字符串引用该绘制的制表停止位置列表。但是，动态小部件 `TextField` 和 `XmText` 并不使用此绘制的制表符资源。它使用 $8 * (\text{width of character } 0)$ 公式来计算制表符的宽度。

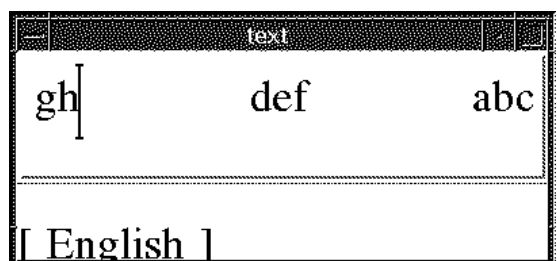
制表符度量值是与复合字符串显示的左边界的距离。如果布局方向是从右向左的，则该距离从右边界开始度量。无论文本方向如何（阿拉伯文从右向左或英文从左向右），制表符都按布局方向（`XmNlayoutDirection`）的规定，向右或向左插入空格。

制表符后面的文本总是在制表停止位置处对齐。制表停止位置从小部件的起始处算起，而小部件的起始处又受到 `XmNlayoutDirection` 的影响。下图阐释了制表符的行为及其与文本的方向性和小部件的 `XmNlayoutDirection` 的交互作用。

此说明的输入是 `abc\tdef\tgh`。



布局方向: XmLEFT_TO_RIGHT



布局方向: XmRIGHT_TO_LEFT

图 6-3 制表符行为

鼠标选定

用户主要用鼠标按钮 1 进行选择。按此按钮会取消对任何当前选择的选定，并将插入光标和锚点移到在文本中按下按钮的位置。按住鼠标按钮 1 并拖动它，可以选择锚点和指针位置之间的所有文本，并取消选定此范围外的任何文本。

选定的文本受资源 `XmNeditPolicy` 影响，而后者可以设置为 `XmEDIT_LOGICAL` 或 `XmEDIT_VISUAL`。如果 `XmNeditPolicy` 设置为 `XmEDIT_LOGICAL`，并且选定的文本是双向的，则选定的文本从视觉上将是不连续的，而是多个文本段的集合。逻辑缓冲区中的文本与显示内容之间没有一对一的对应关系。

因此，双向文本的逻辑字符的连续缓冲区不采用连续的字符流绘制。与此相反，如果 `XmNeditPolicy` 设置为 `XmEDIT_VISUAL`，则选定文本在视觉上可能是连续的，而在逻辑缓冲区中却是分段的。因此，在同一个光标点处按选定、删除和插入双向文本的顺序操作不会产生同一个字符串。

键盘选定

用鼠标进行的选定操作也可以使用键盘来完成。使用 Shift 和箭头键的组合选定文本。

选定的文本受资源 XmNeditPolicy 影响，而后者可设置为 XmEDIT_LOGICAL 或 XmEDIT_VISUAL。如果 XmNeditPolicy 设置为 XmEDIT_LOGICAL，并且选定的文本是双向的，则选定的文本在视觉上是不连续的。由于逻辑缓冲区中的文本与显示内容之间没有一对一的对应关系，因此双向文本中的逻辑字符的连续缓冲区不采用连续的字符流绘制。

与此相反，如果 XmNeditPolicy 设置为 XmEDIT_VISUAL，则选定的文本在视觉上可能是连续的，而在逻辑缓冲区中却是分段的。因此，在同一个光标点处按选定、删除和插入双向文本的顺序操作不会产生同一个字符串。

文本资源和几何图形

下列文本资源与几何学有关：

- 绘制表 XmNrenderTable，小部件用它来选择显示文本时使用的字体或字体集以及其他属性。
Text 和 TextField 小部件仅可使用与字体相关的绘制资源，如 XmNfontType。这些小部件还可以指定布局对象的属性，如 XmNlayoutAttrObject。这些小部件通常包含语言环境标识符和 XmNlayoutModifier，后者指定要传递到与 XOC（它与此 XmRendition 相关）相关的“布局对象”的布局值。
- 一种资源 (XmNwordWrap)，该资源指定在文本比小部件宽时是否在字边界处分行。
在字边界处分行不会将一个新行插入到文本中。对于手写体语言（如阿拉伯语），如果字长度超过小部件长度，则该字将换到下一行。但是，下一行的第一个字符在形式上独立于逻辑缓冲区中的前一个字符。

移植说明

为复杂文本布局 (CTL) 启用的新 Motif 库位于 /usr/dt/lib/libXm.so.4 中。如果您的应用程序链接到 libXm.so.3，则此应用程序不支持 CTL。ldd app_name 显示应用程序链接到的库。要移植现有的应用程序以启用 CTL，需要执行以下步骤：

1. 将 -DSUN_CTL 添加到 Makefile 中。

该标志很重要，它包括支持 CTL 所需的数据结构。此值应在编译期间进行设置。

2. 重新编译现有的应用程序。
此重新编译自动与启用 CTL 的 Motif 库 `libXm.so.4` 进行链接。
3. 将 `XmText.translations` 资源添加到应用程序资源文件中。如果没有这些资源，语言环境的布局引擎将不能启动。
4. 请参考附加在您的文档中的应用程序样例。

注意 - 在 `fontName` 资源中使用您的语言环境中提供的并适用于您的语言环境的字体名称。

例如，如果您希望在 `XmTextField` 或 `XmText` 小部件中进行基于单元的字符移动（泰语），请按以下方式设置相应小部件的转换：

```
XmText.translations: #override \n\  
<Key>osfRight:forward-cell() \n\  
<Key>osfLeft:backward-cell() \n\  
<Key>osfDelete:delete-next-cell() \n\  
<Key>osfBackSpace:delete-previous-cell() \n\  

```


第 7 章

具有 mp 的打印过滤器增强模式

本章介绍 mp 实用程序的打印增强功能。本章讨论下列主题：

- 第 169 页 “用于 UTF-8 的打印”
- 第 170 页 “mp 打印过滤器增强模式概述”
- 第 171 页 “使用 mp.conf 配置文件的本地化”
- 第 177 页 “依赖于语言环境的 prolog 文件”
- 第 177 页 “添加和定制 prolog 文件”
- 第 177 页 “PostScript 文件定制”
- 第 180 页 “.xpr 文件”

用于 UTF-8 的打印

可以打印多种输入文件格式（包括用 UTF-8 编写的纯文本格式）的增强 mp 打印过滤器可以在当前 Solaris 环境中获得。此打印过滤器使用可在 Solaris 系统中获得的 TrueType 和 Type 1 可缩放字体和 X11 位图字体。此过滤器还可以使用打印机常驻字体并可用作 X 打印服务器的客户机。

此实用程序的输出是标准的 PostScript™，可以发送给任何 PostScript 打印机。当配置为 X 打印服务器的客户机时，mp 实用程序还可以输出任何页面描述语言，mp 受打印服务器支持。

要使用此实用程序，键入下面的命令：

```
system% mp filename | lp
```

由于 mp 接受 stdin 流，因此还可以将此实用程序用作过滤器。

```
system% cat filename | mp | lp
```

可将该实用程序设置为一个行式打印机的打印过滤器。例如，下列命令通知打印机服务 LP，打印机 lp1 只接受 mp 格式文件。此命令还将打印机 lp1 安装在端口 /dev/ttya 上。有关更多信息，请参见 lpadm(1M) 手册页。

```
system# lpadmin -p lp1 -v /dev/ttya -I MP
system# accept lp1
system# enable lp1
```

使用 `lpfilter(1M)`，可以按以下方式为过滤器添加此实用程序：

```
system# lpfilter -f lp1 -F pathname
```

此命令通知 LP，可通过名为 `pathname` 的过滤器描述文件使用转换器（此处为 `mp`）。
`pathname` 包含以下信息：

```
Input types: simple
Output types: MP
Command: /usr/bin/mp
```

使用 `/usr/bin/mp` 过滤器将缺省类型文件的输入转换为 PostScript 输出。

要打印 UTF-8 文本文件，请使用下列命令：

```
system% lp -T MP UTF-8-file
```

有关更多信息，请参阅 `mp(1)` 手册页。

mp 打印过滤器增强模式概述

`mp` 打印过滤器在当前 Solaris 发行版中得到增强。最新版的 `mp` 可以用三种不同的模式进行内部处理，以便在一种语言环境中产生输出文件来打印国际文本。可用的模式包括：

- 处理特定于语言环境的字体配置文件 `mp.conf`
- 处理特定于语言环境的 PostScript `prolog.ps` 文件
- 用作 Xprt（X 打印服务器）客户机

下面部分将描述何时使用特定的打印方法以及 `mp` 针对这些打印方法所使用的配置和支持文件。

与特定于语言环境的字体配置文件 `mp.conf` 一起使用 `mp`

如果命令行中没有给出 `-D` 或 `-P` 选项，则此打印方法为缺省方法，除非 `/usr/openwin/lib/locale/$LANG/print` 或 `/usr/lib/lp/locale/$LANG/mp` 目录中存在 `prolog.ps` 文件。`prolog.ps` 文件强制 `mp` 使用文件中的 PostScript 嵌入字体进行打印。即使语言环境中存在 `prolog.ps`，使用 `-M` 选项也会忽略 `prolog.ps` 文件而使用 `mp.conf` 文件（如果存在）。

此方法使用 `/usr/lib/lp/locale/$LANG/mp/mp.conf` 字体配置文件。您可能不必更改此文件，除非您需要使用替换字体进行打印。此文件可使用 TrueType、Type 1 或 pcf 字体进行配置。

与特定于语言环境的 PostScript Prolog 文件一起使用 mp

`/usr/lib/lp/locale/C/` 目录包含对此打印模式通用的 `.ps` 打印页面布局文件。第 177 页“添加和定制 prolog 文件”中提供了如何定制这些文件的描述。

如果命令行中没有给出 `-D` 或 `-P` 选项，并且存在 `/usr/openwin/lib/locale/$LANG/print/prolog.ps`，则 `prolog.ps` 文件将被前置到输出中。根据 `.ps prolog` 页的打印样式，布局文件也被前置到输出中。

此打印方法仅使用 PostScript 字体文件。第 177 页“添加和定制 prolog 文件”中描述了 `prolog.ps` 文件的定制。

将 mp 用作 Xprt (X 打印服务器) 客户机

将 `mp` 用作 Xprt 客户机允许 `mp` 打印连接到网络上受 Xprt print 打印服务支持的任何打印机的输出。作为 Xprt 客户机，`mp` 支持 PostScript 和许多版本的 PCL。

Xprt 客户机按照下列规则尝试连接到 Xprt 服务器：

- 当 `-D printer_name@machine[:dispNum]` 或 `-P printer_name@machine[:dispNum]` 选项与 `mp` 命令一起使用时，`mp` 尝试用 `printer_name` 连接到 `machine[:dispNum]` 上的 Xprt 打印服务。

如果以上到 `machine[:dispNum]` 的尝试连接失败或指定给 `-D` 或 `-P` 的参数只有 `printer_name`，则 `mp` 命令为支持 `printer_name` 参数的 Xprt 服务器检查 `XPSEVERLIST`。例如：

```
system% setenv XPSEVERLIST "machine1[:dispNum1] machine2[:dispNum2] ..."
```

- 如果使用上述规则找不到服务器，则 `mp` 对设置为 `machine[:dispNum]` 的 `XPDISPLAY` 环境变量进行检查。例如：

```
system% setenv XPDISPLAY "machine[:dispNum]"
```

- 如果未设置 `XPDISPLAY` 变量或如果变量无效，则 `mp` 将尝试连接到缺省显示：2100。如果默认显示值无效，`mp` 退出，并显示错误消息。

`/usr/lib/lp/locale/C/mp` 目录包含 Xprt 客户机的 `.xpr` 打印页面样本布局文件。这些是为 300 dpi 打印机创建的样本文件。如果目标打印机具有不同的 dpi 值，则样本文件的 dpi 值自动转换为目标打印机的分辨率。

使用 mp.conf 配置文件的本地化

配置文件为添加或更改字体项或字体组项提供灵活性。

系统配置缺省文件是 `/usr/lib/lp/locale/$LANG/mp/mp.conf`，其中 `$LANG` 是有打印操作的语言环境中的语言环境变量。可以使用 `-u config.file path` 选项指定个人配置文件。

为了便于兼容，连字或变体字型已被编码为字符，这种形式称为 **变形显现形式**。`mp.conf` 文件主要用于将一种语言环境中的中间代码点映射为用来打印该代码点的字体编码中的表现形式。

中间代码点可以是宽字符，也可以是可移植布局服务 (PLS) 层的输出。复杂文本布局打印要求中间代码点为 PLS 输出。由 `mp` 生成的缺省中间代码为 PLS 输出。

当前受支持的字体格式是可移植已编译格式 (PCF)、TrueType 和 Type1 格式。既支持系统常驻 Type1 字体又支持打印机常驻 Type1 字体。请记住下列有关 `mp.conf` 配置文件的格式和内容的信息：

- 行必须以有效关键字（指令）开头。
- 关键字的参数必须与关键字出现在同一行。
- 以 `#` 字符开头的行被认为是注释，注释延续至该行末尾。
- 以 `0x` 开头的数字参数被认为是十六进制数。

`mp.conf` 文件中的不同部分包括：

- 字体别名
- 字体组定义
- 从中间代码范围映射到语言环境中的字体组
- 将每种字体与共享对象相关联，此共享对象把中间代码点映射到字体编码中的变形显现形式

字体别名

`mp.conf` 文件的字体别名部分用于定义打印时使用的每种字体的别名。该部分的每一行都具有以下形式：

`FontNameAlias 字体别名 字体类型 字体路径`

字体别名

为字体名称起别名的常用惯例是，指定字体的编码/脚本名称，名称后面跟一个字母，表示该字体是 Roman 字体、粗体、斜体或粗斜体（R、B、I 或 BI）。

例如，由于 `/usr/openwin/lib/X11/fonts/75dpi/courR18.pcf.Z` 是 iso88591 Roman 字体，因此给它起的别名为 `iso88591R`。

字体类型

对于 `.pcf` 字体，其可能的值为 PCF，对于 Adobe Type1 字体，为 Type1；而对于 TrueType 字体，则为 TrueType。在此 `mp.config` 文件中只能配置此三种字体。

字体路径

字体文件的绝对路径名称。对于 Type1 打印机常驻字体，只指定字体名称，如 Helvetica。

例如，

```
FontNameAlias prnHelveticaR Type1 Helvetica
```

字体组定义

您可以将同一类型的字体合并为一个字体组。字体组的格式如下：

关键字	FontGroup。
字体组名称	字体的组名称。
组类型	字体类型。只为同一类型的字体（PCF、Type1 和 TrueType）创建字体组。
<i>Roman</i>	字体组中的 Roman 字体名称。
粗体	字体组中的粗体字体名称。
斜体	字体组中的斜体字体名称。
粗斜体	字体组中的粗斜体字体名称。

对于创建组，只需要 Roman 字体项。粗体、斜体和粗斜体字体均为可选。例如，邮件或新闻文章的标题可用不同类型的字体来显示。如果仅定义了 Roman 字体，则将使用此字体代替其他字体。

映射部分

mp.conf 文件的映射部分从中间代码范围映射到语言环境中的字体组。本部分中每一行的格式如下所示。

关键字	MapCode2Font。
<i>range_start</i>	以 0x 开头的 4 字节十六进制值表示要映射到一个或多个字体组的代码范围的开始。
<i>range_end</i>	表示要映射的代码范围的结尾。如果值是 '-', 则只有一个中间代码点被映射到目标字体。
组	一个 Type1、PCF 或 TrueType 字体组，变形显现形式通过它进行打印。

关联部分

mp.conf 文件的关联部分将每种字体与共享对象相关联，此共享对象把中间代码点映射到字体编码中的变形显现形式。本部分中每行的格式如下所示：

关键字	CnvCode2Font。
字体别名	为字体定义的别名。
映射函数	获取中间代码并返回字体编码中的变形显现形式，变形显现形式再获取字型索引，从而绘制字型。

具有映射函数的文件路径 包含映射函数的 .so 文件名。您可以使用 dumpcs 中的实用程序来确定 EUC 语言环境的中间代码集。

注意 – 由 mp (1) 使用的当前 TrueType 引擎只能处理格式 4 和 PlatformID 3 cmap。您只能配置 Microsoft .ttf 文件。另外, 要想使 TrueType 字体正常工作, 字符映射编码必须是 Unicode 或 Symbol。由于 Solaris 环境中的大多数 .ttf 字体遵循这些限制, 因此您可以在 mp.conf 文件内映射 Solaris 软件中的所有 TrueType 字体。

您可以创建一个共享对象, 此对象映射字体以与 PCF type1 X 逻辑字体描述 (XLFD) 保持一致。然后, 您可以创建一个从中间代码范围映射到 XLFD 指定的编码的共享对象。例如:

```
-monotype-arial-bold-r-normal-bitmap-10-100-75-75-p-54-iso8859-8
```

相应的 PCF 字体为:

```
/usr/openwin/lib/locale/iso_8859_8/X11/fonts/75dpi/ariabd10.pcf.Z
```

此字体以 ISO 8859-8 进行编码, 因此共享对象必须在中间代码和对应的 ISO 8859-8 代码点之间映射。

如果带有 XLFD 的 TrueType 字体:

```
-monotype-arial-medium-r-normal--0-0-0-0-p-0-iso8859-8
```

具有对应的字体:

```
/usr/openwin/lib/locale/iso_8859_8/X11/fonts/TrueType/arial_h.ttf
```

您应该在中间代码和 Unicode 之间映射, 因为前一种 TrueType 字体的 cmap 编码在 Unicode 中。在此 TrueType 字体实例中, 假定 en_US.UTF-8 语言环境中与希伯来字符 (由 PLS 层产生) 相对应的样例中间代码为 0xe50000e9。由于该字体为 Unicode 编码, 因此以这样的方式来设计对应的 .so 模块内的函数, 即当您传递 0xe50000e9 时, 输出与 Unicode 中的变形显现形式相对应。此处的实例为 0x000005d9。

映射函数的函数原型应为:

```
unsigned int function(unsigned int inter_code_pt)
```

以下是可在 mp.conf: 中使用的可选关键字/值对:

```
PresentationForm      WC/PLSOutput
```

缺省值为 PLSOutput。如果用户指定 WC, 则生成的中间代码点为宽字符。对于 CTL 打印, 应使用此缺省值。

如果语言环境为非 CTL 语言环境并且键盘值为 PLSOutput, 则该值将被忽略, 而且 mp 将生成宽字符代码。

如果语言环境支持 CTL，则您可以使用下表中列出的可选关键字/值对。这些变量可以采用该表中间列中给出的任何可能值。

表 7-1 可选关键字/值对

可选关键字	可选值	Default
Orientation	ORIENTATION_LTR/	ORIENTATION_LTR
	ORIENTATION_RTL/	
	ORIENTATION_CONTEXTUAL	
Numerals	NUMERALS_NOMINAL/	NUMERALS_NOMINAL
	NUMERALS_NATIONAL/	
	NUMERALS_CONTEXTUAL	
TextShaping	TEXT_SHAPED/	TEXT_SHAPED
	TEXT_NOMINAL/	
	TEXT_SHFORM1/	
	TEXT_SHFORM2/	
	TEXT_SHFORM3/	
	TEXT_SHFORM4	

▼ 如何添加打印机驻留的字体

下列过程中的实例阐释了如何将新的 PCF、TrueType 或 Type1 打印机驻留的字体添加到配置文件中。

完成此过程替换当前配置的字体。在前两步骤中，用于显示范围 0x00000021 - 0x0000007f 中字符的 PCF 字体由 TrueType 字体替代。

1. 在添加新字体前，请查看配置文件中与当前配置字体相对应的各种组件。

```
FontNameAlias iso88591R PCF /usr/openwin/lib/X11/fonts/75dpi/courR18PCF.Z
FontNameAlias iso88591B PCF /usr/openwin/lib/X11/fonts/75dpi/courB18PCF.Z
.
.
FontGroup iso88591 PCF iso88591R iso88591B
.
.
MapCode2Font 0x00000020 0x0000007f iso88591
.
.
CnvCode2Font iso88591R _xuiso88591 /usr/lib/lp/locale/$LANG/mp/xuiso88591.so
CnvCode2Font iso88591B _xuiso88591 /usr/lib/lp/locale/$LANG/mp/xuiso88591.so
```

例如，可以将

`/usr/openwin/lib/locale/ja/X11/fonts/TT/HG-MinchoL.ttf` 字体映射到 `en_US.UTF-8` 语言环境。由于 `HG-MinchoL.ttf` 是 Unicode TrueType 字体文件，因而可以使用 `.so` 模块映射函数直接返回传入的 `ucs-2` 代码点。

```
unsigned short _ttfjis0201(unsigned short ucs2) {
    return(ucs2);
}
```

a. 将此映射保存为 `ttfjis0201.c` 文件。

b. 创建共享对象文件。

```
cc -G -Kpic -o ttfjis0201.so ttfjis0201.c
```

2. 要映射 PCF 文件（如

`/usr/openwin/lib/locale/ja/X11/fonts/75dpi/gotmrk20.pcf.Z`），请检查下列与

`/usr/openwin/lib/locale/ja/X11/fonts/75dpi/fonts.dir` 文件中的 XLFD 对应的编码。

```
-sun-gothic-medium-r-normal--22-200-75-75-c-100-jisx0201.1976-0
```

a. 对于 `jisx0201` 编码，请准备一个从 `ucs-2` 映射到 `jisx0201` 的共享对象。获得用于创建 `.so` 模块的映射表。对于 Unicode 语言环境，查找映射到 ftp.unicode.org/pub/MAPPINGS/ 目录中 Unicode 的字符集。

b. 使用以下映射编写一个 `xu2jis0201.c` 文件：

```
unsigned short _xu2jis0201(unsigned short ucs2) {
    if(ucs2 >= 0x20 && ucs2 <= 0x7d )
        return (ucs2);
    if(ucs2==0x203e)
        return (0x7e);
    if(ucs2 >= 0xff61 && ucs2 <= 0xff9f)
        return (ucs2 - 0xff60 + 0xa0);
    return(0);
}
```

c. 在创建映射文件时，请将所有 `usc-2` 包括到 `jisx0201` 的各种事例中。

```
cc -G -o xu2jis0201.so xu2jis0201.c
```

▼ 如何创建共享对象文件

下列过程中的实例说明如何创建共享对象文件。

1. 要添加字体，请编辑与 `mp.conf` 文件中各部分对应的下列实例的行。

下例说明如何添加 TrueType 字体。`.so` 路径指向 `xu2jis0201.so` 文件。

```
FontNameAlias   jis0201R TrueType /home/fn/HG-MinchoL.ttf
FontGroup       jis0201 TrueType jis0201R
MapCode2Font    0x0020      0x007f jis0201
```



```
CnvCode2Font jis0201R _ttfjis0201 <..so path>
```

注意 – 要添加 PCF 字体，请将关键字从 TrueType 更改为 PCF。

2. 用已更改的 `mp.conf` 文件调用 `mp` 命令以打印新字体中的范围 `0x0020-0x007f`。也可以用同样的 `.so` 文件映射其他日文字符范围，例如，可以映射范围 `0x0000FF61 0x0000FF9F`。

注意 – 要保持向后兼容，可以使用 `/usr/openwin/lib/locale/$LANG/print/prolog.ps` 文件在当前语言环境中创建输出。在使用 `prolog.ps` 文件时，不需要任何配置文件。

您可以在 `/usr/lib/lp/locale/en_US.UTF-8/mp` 目录中找到样本 `mp.conf` 文件。

添加和定制 prolog 文件

prolog 文件可以分为两种主要类别：

- PostScript prolog 文件 (`.ps`)
- X 打印服务器客户机 prolog 文件 (`.xpr`)。

PostScript 文件定制

PostScript 文件属于下列种类：

- 通用 prolog 文件
- 打印布局 prolog 文件

依赖于语言环境的 prolog 文件

`prolog.ps` 文件的用途是设置非通用字体。应用程序使用这些预定义 PostScript 字体名称进行打印。prolog 文件必须为桌面设置日历管理器和 `mp` 至少定义下面字体名称：

- LC_Times-Roman
- LC_Times-Bold

- LC_Helvetica
- LC_Helvetica-Bold
- LC_Courier
- LC_Helvetica-BoldOblique
- LC_Times-Italic

下列实例使用这些字体打印指定的特殊本地字符集：

```
100 100 moveto
/LC_Times-Roman findfont 24 scale font setfont
(Any text string in your locale) show
```

Solaris 本地化工具包为日语环境提供了一个样例 `prolog.ps` 文件。另外，此文件还位于 `/usr/openwin/lib/locale/ja/print/` 目录中。

下例说明如何在现有的 `prolog.ps` 文件中添加或更改复合字体。

```
%
(Foo-Fine) makecodeset12
(Base-Font) makeEUCfont
%
```

例如，您可以定义名为 `LC_Base-Font` 的复合字体。`LC_Base-Font` 可能是包含语言环境字符集和 `Base-Font` 的 `Foo-Fine` 字体的复合体。添加或更改字体并不需要深入的 PostScript 编程知识。

对实例版本进行研究是创建 `prolog.ps` 文件的最好方法。在实例 `prolog.ps` 中，需要编写两个例程：`makecodeset12` 和 `makeEUCfont`。例程 `makecodeset12` 设置本地字体编码信息。此例程可能会因语言环境的不同而不同。例程 `makeEUCfont` 将基准字体和语言环境字体合并为一个复合字体。`prolog` 文件的创建者应具有良好的 PostScript 知识，以便编写 `makecodeset12` 和 `makeEUCfont`。

`prolog.ps` 文件支持只保持向后兼容性。在为语言环境生成打印输出时，不创建新的 `prolog.ps` 文件。请换而使用 `mp.conf`。

`prolog.ps` 文件的路径为

```
/usr/openwin/lib/locale/$LANG/print/prolog.ps
```

通用 PostScript prolog 文件

通用 `prolog` 文件是 `mp.common.ps`。

其他每个页面布局 `prolog` 文件都需要包含此文件。

`mp.common.ps` 文件驻留在 `/usr/lib/lp/locale/C/mp/` 目录中。此文件包含一个 PostScript 例程，将一个字体从标准编码重新编码为 ISO 8859-1 编码。`.reencodeISO` 例程从打印布局 `prolog` 文件中调用以更改字体的编码。通常情况下，此 `prolog` 文件不需要进行任何定制。如果您创建自己的 `prolog` 文件，请将环境变量 `MP_PROLOGUE` 设置为指向包含修改后的 `prolog` 文件的目录。

打印布局 prolog 文件

打印布局 prolog 文件、mp.*.ps 文件包含控制用于打印的页面布局的例程。除了为具有用户名、打印日期和页号的打印页提供页眉和页脚外，这些 prolog 文件还可以提供其他信息。例如，prolog 文件可提供有效的打印区域尺寸以及打印所要使用的横向和纵向模式。

打印布局 prolog 文件为：

- mp.pro.ps
- mp.pro.alt.ps
- mp.pro.fp.ps
- mp.pro.ps
- mp.pro.ts.ps
- mp.pro.altl.ps
- mp.pro.ff.ps
- mp.pro.l.ps
- mp.pro.ll.ps
- mp.pro.tm.ps

需要在每个 prolog 文件中定义一组标准函数。当新的打印页启动、某个打印页结束或新的列结束时这些函数即被调用。这些函数的实现定义了打印输出的打印属性。

下列 PostScript 变量在运行时由 mp 二进制程序定义。所有的打印布局文件都可以使用这些变量来打印动态信息，如用户名、主题和打印时间。从变量获取的此信息通常出现在打印页的页眉或页脚中。

用户	正在运行 mp 的用户的名称，可从系统 passwd 文件中获取。
MailFor	用于保存要打印的文章类型名称的变量。此变量的可能值为： <ul style="list-style-type: none">■ Listing for—当输入为文本文件时■ Mail for—当输入为邮件文件时■ Article from—当输入为新闻组文章时
主题	从邮件和新闻标题中获取的主题。可以使用 -s 选项将某个主题强制为邮件和新闻文件以及正常的文本文件。
Timenow	出现在标题和页脚中的打印时间。此信息从 localtime () 函数中获取。

下列函数在打印布局 prolog 文件中实现。所有这些函数都可以使用子函数。

endpage	用法:page_number endpage 当到达打印页的底部时调用。此函数恢复页面的图形上下文并发出 showpage。在某些 prolog 文件中，页眉和页脚信息按逐页模式而非逐列模式显示。您可以实现此函数以调用显示页眉和页脚灰度级菱形的子函数。
newpage	用法:page_number newpage

当某个新页开始时要执行的例程或命令。设置横向打印模式、保存打印图形上下文和转换页坐标是用于这些例程的一些函数。

endcol 用法:page_number col_number endcol

用于显示页眉和页脚信息，移到新打印位置，等等。

要添加新的打印布局 prolog 文件，需要在打印布局 prolog 文件内显式定义下列变量。

NumCols 打印页中的列数。缺省值为 2。

PrintWidth 打印区域的宽度（以英寸为单位）。缺省值为 6。

PrintHeight 打印区域的高度（以英寸为单位）。缺省值为 9。

.xpr 文件

缺省情况下，这些文件位于 /usr/lib/lp/locale/C/mp/。 .xpr 文件与每个 PostScript prolog 布局文件相对应，但 mp.common.ps. 除外。您可以通过定义 MP_PROLOGUE 环境变量来定义替换的 prolog 目录。

这些文件用作关键字/值对。以 # 开头的行被当做注释。空格分隔不同的标记（除非显式进行说明）。每个 .xpr 文件的三个主要部分由下列关键字对绑定：

- STARTCOMMON/ENDCOMMON
- STARTPAGE/ENDPAGE
- STARTCOLUMN/ENDCOLUMN
- STARTFORCEDPAGE/ENDFORCEDPAGE
- STARTFORCEDCOLUMN/ENDFORCEDCOLUMN

某些关键字/值对可用于这三个区域。在下节中介绍每个区域。

STARTCOMMON/ENDCOMMON 关键字

出现在 STARTCOMMON 关键字之后并且在 ENDCOMMON 关键字之前的所有关键字/值对打印页的常规属性进行定义。关键字的不同有效值使用斜线 (/) 字符分隔。

ORIENTATION 0/1

0 表示以纵向模式打印，而 1 表示以横向模式打印。

PAGELength 无符号整数

一个值，用于表示每个逻辑页的行数。

LINELENGTH 无符号整数

表示每行的单列字符数的值。

NUMCOLS 无符号整数

每个物理页的逻辑页数。

HDNGFONTSIZE 无符号整数

标题字体点尺寸（以小数点表示）。

BODYFONTSIZE 无符号整数

正文字体点尺寸（以小数点表示）。

PROLOGDPI 无符号整数

创建当前 .xpr 文件使用的每英寸点数比例。

YTEXTBOUNDARY 无符号整数

此 y 坐标确定了页或逻辑页（列）中文本打印的边界。此边界还可以用于检查文本打印是否发生在预期的区域内。在复杂文本布局和 EUC 打印时需要此边界，这是因为从相应的字体中获取的字符高度信息有可能是错的。

STARTTEXT 无符号整数无符号整数

以小数点表示的 x/y 点，这是物理页的第一个逻辑页中实际文本打印的开始位置。

PAGESTRING 0/1

1 表示需要在标题中的页号前附加一个页字符串。

0 表示只显示页号。

EXTRAHDNGFONT font string 1、font string 2 ... font string n

字体字符串是 X 逻辑字体描述。将关键字 EXTRAHDNGFONT 与逗号分隔的字体名称列表分开的标记是引号 " 字符，而不是空格或制表符。当打印标题时，给定的首选项是这些字体，而不是内置字体。通常情况下，EXTRABODYFONT 用于分配在 /usr/openwin/server/etc/XpConfig/C/print/models/<model name>/fonts 目录中配置的打印机常驻字体。

fonts.dir 文件包含打印机常驻字体的 XLFD。

在 .xpr 文件中，通常指定字体，如下例所示：

```
"-monotype-Gill Sans-Regular-r-normal- *-%d-*-*p-0-iso8859-2"
```

%d（如果存在）被 mp 替换为 .xpr 文件中当前标题字体的点尺寸。x 分辨率和 y 分辨率由 * 指定。平均宽度字段设置为 0 表示选择了可缩放字体（如有可能）。您还可以提供更多的特定字体名称。

EXTRABODYFONT font string 1、font string 2 ... font string n

与 EXTRAHDNGFONT 相同，除了这些字体被用来打印页面正文以外。

XDISPLACEMENT 带符号/无符号整数

提供要应用于此页的 x 坐标位移，以便在 x 方向移动该页的内容。此位移可以是 +ve 或 -ve 值。

YDISPLACEMENT 带符号/无符号整数

与 x 位移相同（除了移动发生在 y 方向）。

当您处理某些具有非标准页边距宽度并且要求在页中移动打印内容时的打印机时，这两个关键字很有用。

STARTPAGE/ENDPAGE 关键字

此部分中的关键字值对由 STARTPAGE 和 ENDPAGE 关键字绑定。此部分包含要应用于物理页的绘图和标题信息。物理页可包含许多逻辑页，但是，包含在这些关键字之间的所有绘图例程仅一次应用于逻辑页。

有效的绘图实体为 LINE 和 ARC。XDrawLine() 和 XDrawArc() 函数在这些关键字的值上执行。

此部分内的尺寸被映射在 PROLOGDPI 单元中。角度以度数表示。

LINE x1 y1 x2 y2	x/y 无符号坐标定义连接一条直线的两个点。
ARC x y width height angle1 angle2	x 和 y 均为表示弧线起始点的无符号整数。width 和 height 为表示弧线的宽度和高度的无符号整数。
USERSTRINGPOS x y	无符号坐标表示标题中用户信息的打印位置。
TIMESTRINGPOS x y	无符号坐标表示标题中打印时间的打印位置。
PAGESTRINGPOS x y	无符号坐标表示每个打印页的页字符串的打印位置。
SUBJECTSTRINGPOS x y	无符号坐标表示页中主题的打印位置。

STARTFORCEDPAGE/ENDFORCEDPAGE 部分

当 -n 选项分配给 mp 时，STARTPAGE/ENDPAGE 部分内的所有装饰都不打印。但是，即使给出 -n 选项，也会打印 STARTFORCEDPAGE/ENDFORCEDPAGE 部分内包含的任何内容。

STARTCOLUMN/ENDCOLUMN 部分

所有关键字与第 182 页“STARTPAGE/ENDPAGE 关键字”中描述的相同，除了此部分中的条目可 NUMCOLS 次应用于物理页以外。如果 NUMCOLS 为 3，则物理页的可打印区域将被分为三个部分，并且每页中直线、弧线或标题装饰将出现三次。

STARTFORCEDCOLUMN/ENDFORCEDCOLUMN 部分

当 -n 选项分配给 mp 时，STARTCOLUMN/ENDCOLUMN 部分内给定的所有装饰都不打印。但是，即使给出 -n 选项，也会打印 STARTFORCEDCOLUMN/ENDFORCEDCOLUMN 部分内包含的任何内容。

创建新的 .xpr 文件

当创建新的 .xpr prolog 文件时，只需指定与缺省值不同的值。

下表列出针对不同关键字的 mp 程序缺省值（如果在 STARTCOMMON/ENDCOMMON 部分的 .xpr 文件中没有指定这些值）：

表 7-2 STARTCOMMON/ENDCOMMON 关键字值

关键字	值
ORIENTATION	0
PAGELLENGTH	60
LINELLENGTH	80
YTEXTBOUNDARY	3005
NUMCOLS	01
HDNGFONTSIZE	120
PROLOGDPI	300
STARTTEXT	135 280
PAGESTRING	0

由 STARTPAGE/ENDPAGE 和 STARTCOLUMN/ENDCOLUMN 绑定的另两个部分不需要缺省值。

要创建不带装饰的页，请使用如下方式：每个物理页有四个逻辑页，打印格式为纵向格式。指定下列部分和值：

```
STARTCOMMON
NUMCOLS 04
LINELLENGTH 20
ENDCOMMON
```

在创建不带装饰的页面时，不需要指定下列两个部分：

```
STARTPAGE/ENDPAGE
STARTCOLUMN/ENDCOLUMN
```

如果您不在打印页中加入装饰，则不需要这些参数。所有坐标的缺省值都为 300 dpi，除非您没有指定 PROLOGDPI 关键字。如果目标打印机的分辨率不同，程序将调整 .xpr 文件以适应此分辨率。

在创建 .xpr 文件时，您必须事先知道纸张的尺寸。在美国使用的纸张为 8.5x11 英寸，对于分辨率为 300 dpi 的打印机，2550X3300 为总的尺寸。大多数打印机无法从纸张的左上角打印。相反，在物理纸张的四周会分配一些边距空间。即使您试图从 0,0 打印，也不会打印到页的左上角。当创建新的 .xpr 文件时，请考虑此限制。

附录 A

iconv 代码转换

本附录列出当前 Solaris 操作环境中可用的与 Unicode 相关的代码转换模块。

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块

从代码 (符号)	到代码 (符号)
646 (ISO 646)	UCS-2
646 (ISO 646)	USC-2BE
646 (ISO 646)	UCS-2LE
646 (ISO 646)	USC-4
646 (ISO 646)	USC-4BE
646 (ISO 646)	USC-4LE
646 (ISO 646)	UTF-8
646 (ISO 646)	UTF-16
646 (ISO 646)	UTF-16BE
646 (ISO 646)	UTF-16LE
646 (ISO 646)	UTF-32
646 (ISO 646)	UTF-32BE
646 (ISO 646)	UTF-32LE
ISO8859-11	UTF-8
8859-1 (ISO8859-1)	UCS-2
8859-1 (ISO8859-1)	UCS-2BE
8859-1 (ISO8859-1)	UCS-2LE

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
8859-1 (ISO8859-1)	UCS-4
8859-1 (ISO8859-1)	UCS-4BE
8859-1 (ISO8859-1)	UCS-4LE
8859-1 (ISO8859-1)	UTF-8
8859-1 (ISO8859-1)	UTF-16
8859-1 (ISO8859-1)	UTF-16BE
8859-1 (ISO8859-1)	UTF-16LE
8859-1 (ISO8859-1)	UTF-32
8859-1 (ISO8859-1)	UTF-32BE
8859-1 (ISO8859-1)	UTF-32LE
8859-2 (ISO8859-2)	UCS-2
8859-2 (ISO8859-2)	UCS-2BE
8859-2 (ISO8859-2)	UCS-2LE
8859-2 (ISO8859-2)	UCS-4
8859-2 (ISO8859-2)	UCS-4BE
8859-2 (ISO8859-2)	UCS-4LE
8859-2 (ISO8859-2)	UTF-8
8859-2 (ISO8859-2)	UTF-16
8859-2 (ISO8859-2)	UTF-16BE
8859-2 (ISO8859-2)	UTF-16LE
8859-2 (ISO8859-2)	UTF-32
8859-2 (ISO8859-2)	UTF-32BE
8859-2 (ISO8859-2)	UTF-32LE
8859-3 (ISO8859-3)	UCS-2
8859-3 (ISO8859-3)	UCS-2BE
8859-3 (ISO8859-3)	UCS-2LE
8859-3 (ISO8859-3)	UCS-4
8859-3 (ISO8859-3)	UCS-4BE
8859-3 (ISO8859-3)	UCS-4LE

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
8859-3 (ISO8859-3)	UTF-8
8859-3 (ISO8859-3)	UTF-16
8859-3 (ISO8859-3)	UTF-16BE
8859-3 (ISO8859-3)	UTF-16LE
8859-3 (ISO8859-3)	UTF-32
8859-3 (ISO8859-3)	UTF-32BE
8859-3 (ISO8859-3)	UTF-32LE
8859-4 (ISO8859-4)	UCS-2
8859-4 (ISO8859-4)	UCS-2BE
8859-4 (ISO8859-4)	UCS-2LE
8859-4 (ISO8859-4)	UCS-4
8859-4 (ISO8859-4)	UCS-4BE
8859-4 (ISO8859-4)	UCS-4LE
8859-4 (ISO8859-4)	UTF-8
8859-4 (ISO8859-4)	UTF-16
8859-4 (ISO8859-4)	UTF-16BE
8859-4 (ISO8859-4)	UTF-16LE
8859-4 (ISO8859-4)	UTF-32
8859-4 (ISO8859-4)	UTF-32BE
8859-4 (ISO8859-4)	UTF-32LE
8859-5 (ISO8859-5)	UCS-2
8859-5 (ISO8859-5)	UCS-2BE
8859-5 (ISO8859-5)	UCS-2LE
8859-5 (ISO8859-5)	UCS-4
8859-5 (ISO8859-5)	UCS-4BE
8859-5 (ISO8859-5)	UCS-4LE
8859-5 (ISO8859-5)	UTF-8
8859-5 (ISO8859-5)	UTF-16
8859-5 (ISO8859-5)	UTF-16BE

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
8859-5 (ISO8859-5)	UTF-16LE
8859-5 (ISO8859-5)	UTF-32
8859-5 (ISO8859-5)	UTF-32BE
8859-5 (ISO8859-5)	UTF-32LE
8859-6 (ISO8859-6)	UCS-2
8859-6 (ISO8859-6)	UCS-2BE
8859-6 (ISO8859-6)	UCS-2LE
8859-6 (ISO8859-6)	UCS-4
8859-6 (ISO8859-6)	UCS-4BE
8859-6 (ISO8859-6)	UCS-4LE
8859-6 (ISO8859-6)	UTF-8
8859-6 (ISO8859-6)	UTF-16
8859-6 (ISO8859-6)	UTF-16BE
8859-6 (ISO8859-6)	UTF-16LE
8859-6 (ISO8859-6)	UTF-32
8859-6 (ISO8859-6)	UTF-32BE
8859-6 (ISO8859-6)	UTF-32LE
8859-7 (ISO8859-7)	UCS-2
8859-7 (ISO8859-7)	UCS-2BE
8859-7 (ISO8859-7)	UCS-2LE
8859-7 (ISO8859-7)	UCS-4
8859-7 (ISO8859-7)	UCS-4BE
8859-7 (ISO8859-7)	UCS-4LE
8859-7 (ISO8859-7)	UTF-8
8859-7 (ISO8859-7)	UTF-16
8859-7 (ISO8859-7)	UTF-16BE
8859-7 (ISO8859-7)	UTF-16LE
8859-7 (ISO8859-7)	UTF-32
8859-7 (ISO8859-7)	UTF-32BE

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
8859-7 (ISO8859-7)	UTF-32LE
8859-8 (ISO8859-8)	UCS-2
8859-8 (ISO8859-8)	UCS-2BE
8859-8 (ISO8859-8)	UCS-2LE
8859-8 (ISO8859-8)	UCS-4
8859-8 (ISO8859-8)	UCS-4BE
8859-8 (ISO8859-8)	UCS-4LE
8859-8 (ISO8859-8)	UTF-8
8859-8 (ISO8859-8)	UTF-16
8859-8 (ISO8859-8)	UTF-16BE
8859-8 (ISO8859-8)	UTF-16LE
8859-8 (ISO8859-8)	UTF-32
8859-8 (ISO8859-8)	UTF-32BE
8859-8 (ISO8859-8)	UTF-32LE
8859-9 (ISO8859-9)	UCS-2
8859-9 (ISO8859-9)	UCS-2BE
8859-9 (ISO8859-9)	UCS-2LE
8859-9 (ISO8859-9)	UCS-4
8859-9 (ISO8859-9)	UCS-4BE
8859-9 (ISO8859-9)	UCS-4LE
8859-9 (ISO8859-9)	UTF-8
8859-9 (ISO8859-9)	UTF-16
8859-9 (ISO8859-9)	UTF-16BE
8859-9 (ISO8859-9)	UTF-16LE
8859-9 (ISO8859-9)	UTF-32
8859-9 (ISO8859-9)	UTF-32BE
8859-9 (ISO8859-9)	UTF-32LE
8859-10 (ISO8859-10)	UCS-2
8859-10 (ISO8859-10)	UCS-2BE

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
8859-10 (ISO8859-10)	UCS-2LE
8859-10 (ISO8859-10)	UCS-4
8859-10 (ISO8859-10)	UCS-4BE
8859-10 (ISO8859-10)	UCS-4LE
8859-10 (ISO8859-10)	UTF-8
8859-10 (ISO8859-10)	UTF-16
8859-10 (ISO8859-10)	UTF-16BE
8859-10 (ISO8859-10)	UTF-16LE
8859-10 (ISO8859-10)	UTF-32
8859-10 (ISO8859-10)	UTF-32BE
8859-10 (ISO8859-10)	UTF-32LE
8859-13 (ISO8859-13)	UCS-2
8859-13 (ISO8859-13)	UCS-2BE
8859-13 (ISO8859-13)	UCS-2LE
8859-13 (ISO8859-13)	UCS-4
8859-13 (ISO8859-13)	UCS-4BE
8859-13 (ISO8859-13)	UCS-4LE
8859-13 (ISO8859-13)	UTF-8
8859-13 (ISO8859-13)	UTF-16
8859-13 (ISO8859-13)	UTF-16BE
8859-13 (ISO8859-13)	UTF-16LE
8859-13 (ISO8859-13)	UTF-32
8859-13 (ISO8859-13)	UTF-32BE
8859-13 (ISO8859-13)	UTF-32LE
8859-14 (ISO8859-14)	UCS-2
8859-14 (ISO8859-14)	UCS-2BE
8859-14 (ISO8859-14)	UCS-2LE
8859-14 (ISO8859-14)	UCS-4
8859-14 (ISO8859-14)	UCS-4BE

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
8859-14 (ISO8859-14)	UCS-4LE
8859-14 (ISO8859-14)	UTF-8
8859-14 (ISO8859-14)	UTF-16
8859-14 (ISO8859-14)	UTF-16BE
8859-14 (ISO8859-14)	UTF-16LE
8859-14 (ISO8859-14)	UTF-32
8859-14 (ISO8859-14)	UTF-32BE
8859-14 (ISO8859-14)	UTF-32LE
8859-15 (ISO8859-15)	UCS-2
8859-15 (ISO8859-15)	UCS-2BE
8859-15 (ISO8859-15)	UCS-2LE
8859-15 (ISO8859-15)	UCS-4
8859-15 (ISO8859-15)	UCS-4BE
8859-15 (ISO8859-15)	UCS-4LE
8859-15 (ISO8859-15)	UTF-8
8859-15 (ISO8859-15)	UTF-16
8859-15 (ISO8859-15)	UTF-16BE
8859-15 (ISO8859-15)	UTF-16LE
8859-15 (ISO8859-15)	UTF-32
8859-15 (ISO8859-15)	UTF-32BE
8859-15 (ISO8859-15)	UTF-32LE
8859-16 (ISO8859-16)	UCS-2
8859-16 (ISO8859-16)	UCS-2BE
8859-16 (ISO8859-16)	UCS-2LE
8859-16 (ISO8859-16)	UCS-4
8859-16 (ISO8859-16)	UCS-4BE
8859-16 (ISO8859-16)	UCS-4LE
8859-16 (ISO8859-16)	UTF-8
8859-16 (ISO8859-16)	UTF-16

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
8859-16 (ISO8859-16)	UTF-16BE
8859-16 (ISO8859-16)	UTF-16LE
8859-16 (ISO8859-16)	UTF-32
8859-16 (ISO8859-16)	UTF-32BE
8859-16 (ISO8859-16)	UTF-32LE
ACE	UTF-8
ACE-ALLOW-UNASSIGNED	UTF-8
eucJP	UTF-8
gb2312	UTF-8
iso2022	UTF-8
ko_KR-cp933	UTF-8
ko_KR-euc	UTF-8
ko_KR-iso2022-7	UTF-8
ko_KR-johap	UTF-8
ko_KR-johap92	UTF-8
zh_TW-euc	UTF-8
zh_TW-cp937	UTF-8
zh_TW-iso2022-7	UTF-8
GBK	UTF-8
FujitsuJEF-ascii-code	UTF-8
FujitsuJEF-ascii-face	UTF-8
FujitsuJEF-kana-code	UTF-8
FujitsuJEF-kana-face	UTF-8
HitachiKEIS83	UTF-8
HitachiKEIS90	UTF-8
ISO-2022-JP	UTF-8
KOI8-R	UCS-2
KOI8-R	UCS-2BE
KOI8-R	UCS-2LE

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
KOI8-R	UCS-4
KOI8-R	UCS-4BE
KOI8-R	UCS-4LE
KOI8-R	UTF-8
KOI8-R	UTF-16
KOI8-R	UTF-16BE
KOI8-R	UTF-16LE
KOI8-R	UTF-32
KOI8-R	UTF-32BE
KOI8-R	UTF-32LE
KOI8-U	UCS-2
KOI8-U	UCS-2BE
KOI8-U	UCS-2LE
KOI8-U	UCS-4
KOI8-U	UCS-4BE
KOI8-U	UCS-4LE
KOI8-U	UTF-8
KOI8-U	UTF-16
KOI8-U	UTF-16BE
KOI8-U	UTF-16LE
KOI8-U	UTF-32
KOI8-U	UTF-32BE
KOI8-U	UTF-32LE
NECJIPS	UTF-8
PCK	UTF-8
PTCP154	UCS-2
PTCP154	UCS-2BE
PTCP154	UCS-2LE
PTCP154	UCS-4

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
PTCP154	UCS-4BE
PTCP154	UCS-4LE
PTCP154	UTF-16
PTCP154	UTF-16BE
PTCP154	UTF-16LE
PTCP154	UTF-32
PTCP154	UTF-32BE
PTCP154	UTF-32LE
PTCP154	UTF-8
UCS-2	646 (ISO 646)
UCS-2	8859-1 (ISO8859-1)
UCS-2	8859-2 (ISO8859-2)
UCS-2	8859-3 (ISO8859-3)
UCS-2	8859-4 (ISO8859-4)
UCS-2	8859-5 (ISO8859-5)
UCS-2	8859-6 (ISO8859-6)
UCS-2	8859-7 (ISO8859-7)
UCS-2	8859-8 (ISO8859-8)
UCS-2	8859-9 (ISO8859-9)
UCS-2	8859-10 (ISO8859-10)
UCS-2	8859-13 (ISO8859-13)
UCS-2	8859-14 (ISO8859-14)
UCS-2	8859-15 (ISO8859-15)
UCS-2	8859-16 (ISO8859-16)
UCS-2	KOI8-R
UCS-2	KOI8-U
UCS-2	PTCP154
UCS-2BE	PTCP154
UCS-2LE	PTCP154

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UCS-4	PTCP154
UCS-4BE	PTCP154
UCS-4LE	PTCP154
UTF-16	PTCP154
UTF-16BE	PTCP154
UTF-16LE	PTCP154
UTF-32	PTCP154
UTF-32BE	PTCP154
UTF-32LE	PTCP154
UTF-8	PTCP154
UCS-2	UCS-4
UCS-2	UCS-4BE
UCS-2	UCS-4LE
UCS-2	UTF-7
UCS-2	UTF-8
UCS-2BE	646 (ISO 646)
UCS-2BE	8859-1 (ISO8859-1)
UCS-2BE	8859-2 (ISO8859-2)
UCS-2BE	8859-3 (ISO8859-3)
UCS-2BE	8859-4 (ISO8859-4)
UCS-2BE	8859-5 (ISO8859-5)
UCS-2BE	8859-6 (ISO8859-6)
UCS-2BE	8859-7 (ISO8859-7)
UCS-2BE	8859-8 (ISO8859-8)
UCS-2BE	8859-9 (ISO8859-9)
UCS-2BE	8859-10 (ISO8859-10)
UCS-2BE	8859-13 (ISO8859-13)
UCS-2BE	8859-14 (ISO8859-14)
UCS-2BE	8859-15 (ISO8859-15)

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UCS-2BE	8859-16 (ISO8859-16)
UCS-2BE	KOI8-R
UCS-2BE	KOI8-U
UCS-2BE	UCS-4
UCS-2BE	UCS-4BE
UCS-2BE	UCS-4LE
UCS-2BE	UTF-8
UCS-2LE	646 (ISO 646)
UCS-2LE	8859-1 (ISO8859-1)
UCS-2LE	8859-2 (ISO8859-2)
UCS-2LE	8859-3 (ISO8859-3)
UCS-2LE	8859-4 (ISO8859-4)
UCS-2LE	8859-5 (ISO8859-5)
UCS-2LE	8859-6 (ISO8859-6)
UCS-2LE	8859-7 (ISO8859-7)
UCS-2LE	8859-8 (ISO8859-8)
UCS-2LE	8859-9 (ISO8859-9)
UCS-2LE	8859-10 (ISO8859-10)
UCS-2LE	8859-13 (ISO8859-13)
UCS-2LE	8859-14 (ISO8859-14)
UCS-2LE	8859-15 (ISO8859-15)
UCS-2LE	8859-16 (ISO8859-16)
UCS-2LE	KOI8-R
UCS-2LE	KOI8-U
UCS-2LE	UCS-4
UCS-2LE	UCS-4BE
UCS-2LE	UCS-4LE
UCS-2LE	UTF-8
UCS-2LE	UTF-32

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UCS-2LE	UTF-32BE
UCS-2LE	UTF-32LE
UCS-4	646
UCS-4	8859-1 (ISO8859-1)
UCS-4	8859-2 (ISO8859-2)
UCS-4	8859-3 (ISO8859-3)
UCS-4	8859-4 (ISO8859-4)
UCS-4	8859-5 (ISO8859-5)
UCS-4	8859-6 (ISO8859-6)
UCS-4	8859-7 (ISO8859-7)
UCS-4	8859-8 (SO 8859-8)
UCS-4	8859-9 (ISO8859-9)
UCS-4	8859-10 (ISO8859-10)
UCS-4	8859-13 (ISO8859-13)
UCS-4	8859-14 (ISO8859-14)
UCS-4	8859-15 (ISO8859-15)
UCS-4	8859-16 (ISO8859-16)
UCS-4	KOI8-R
UCS-4	KOI8-U
UCS-4	UCS-2
UCS-4	UCS-2BE
UCS-4	UCS-2LE
UCS-4	UTF-7
UCS-4	UTF-8
UCS-4	UCS-16
UCS-4	UCS-16BE
UCS-4	UCS-16LE
UCS-4	UTF-32
UCS-4	UCS-32BE

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UCS-4	UCS-32LE
UCS-4BE	646
UCS-4BE	8859-1 (ISO8859-1)
UCS-4BE	8859-2 (ISO8859-2)
UCS-4BE	8859-3 (ISO8859-3)
UCS-4BE	8859-4 (ISO8859-4)
UCS-4BE	8859-5 (ISO8859-5)
UCS-4BE	8859-6 (ISO8859-6)
UCS-4BE	8859-7 (ISO8859-7)
UCS-4BE	8859-8 (SO 8859-8)
UCS-4BE	8859-9 (ISO8859-9)
UCS-4BE	8859-10 (ISO8859-10)
UCS-4BE	8859-13 (ISO8859-13)
UCS-4BE	8859-14 (ISO8859-14)
UCS-4BE	8859-15 (ISO8859-15)
UCS-4BE	8859-16 (ISO8859-16)
UCS-4BE	KOI8-R
UCS-4BE	KOI8-U
UCS-4BE	UCS-2
UCS-4BE	UCS-2BE
UCS-4BE	UCS-2LE
UCS-4BE	UCS-8
UCS-4BE	UCS-16
UCS-4BE	UCS-16BE
UCS-4BE	UCS-16LE
UCS-4BE	UCS-32
UCS-4BE	UCS-32BE
UCS-4BE	UCS-32LE
UCS-4LE	646 (ISO 646)

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UCS-4LE	8859-1 (ISO8859-1)
UCS-4LE	8859-2 (ISO8859-2)
UCS-4LE	8859-3 (ISO8859-3)
UCS-4LE	8859-4 (ISO8859-4)
UCS-4LE	8859-5 (ISO8859-5)
UCS-4LE	8859-6 (ISO8859-6)
UCS-4LE	8859-7 (ISO8859-7)
UCS-4LE	8859-8 (SO 8859-8)
UCS-4LE	8859-9 (ISO8859-9)
UCS-4LE	8859-10 (ISO8859-10)
UCS-4LE	8859-13 (ISO8859-13)
UCS-4LE	8859-14 (ISO8859-14)
UCS-4LE	8859-15 (ISO8859-15)
UCS-4LE	8859-16 (ISO8859-15)
UCS-4LE	KOI8-R
UCS-4LE	KOI8-U
UCS-4LE	UCS-2
UCS-4LE	UCS-2BE
UCS-4LE	UCS-2LE
UCS-4LE	UTF-16
UCS-4LE	UTF-16BE
UCS-4LE	UTF-16LE
UCS-4LE	UTF-8
UTF-7	UCS-2
UTF-7	UCS-4
UTF-7	UCS-8
UTF-8	646 (ISO 646)
UTF-8	8859-1 (ISO8859-1)
UTF-8	8859-2 (ISO8859-2)

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UTF-8	8859-3 (ISO8859-3)
UTF-8	8859-4 (ISO8859-4)
UTF-8	8859-5 (ISO8859-5)
UTF-8	8859-6 (ISO8859-6)
UTF-8	8859-7 (ISO8859-7)
UTF-8	8859-8 (ISO8859-8)
UTF-8	8859-9 (ISO8859-9)
UTF-8	8859-10 (ISO8859-10)
UTF-8	8859-11 (ISO8859-11)
UTF-8	8859-13 (ISO8859-13)
UTF-8	8859-14 (ISO8859-14)
UTF-8	8859-15 (ISO8859-15)
UTF-8	8859-16 (ISO8859-16)
UTF-8	ACE
UTF-8	ACE-ALLOW-UNASSIGNED
UTF-8	eucJP
UTF-8	gb2312
UTF-8	iso2022
UTF-8	ko_KR-euc
UTF-8	ko_KR-johap
UTF-8	ko_KR-johap92
UTF-8	ko_KR-iso2022-7
UTF-8	zh_TW-euc
UTF-8	zh_TW-iso2022-7
UTF-8	zh_TW-cp937
UTF-8	FujitsuJEF-ascii-code
UTF-8	FujitsuJEF-ascii-face
UTF-8	FujitsuJEF-kana-code
UTF-8	FujitsuJEF-kana-face

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UTF-8	GBK
UTF-8	HitachiKEIS83
UTF-8	HitachiKEIS90
UTF-8	ISO-2022-JP
UTF-8	KOI8-R
UTF-8	KOI8-U
UTF-8	UTF-7
UTF-8	NECJIPS
UTF-8	PCK
UTF-8	UCS-2
UTF-8	UCS-2BE
UTF-8	UCS-2LE
UTF-8	UCS-4
UTF-8	UCS-4BE
UTF-8	UCS-4LE
UTF-8	UTF-7
UTF-8	UTF-8
UTF-8	UTF-16
UTF-8	UTF-16BE
UTF-8	UCS-16LE
UTF-16	646 (ISO 646)
UTF-16	8859-1 (ISO8859-1)
UTF-16	8859-2 (ISO8859-2)
UTF-16	8859-3 (ISO8859-3)
UTF-16	8859-4 (ISO8859-4)
UTF-16	8859-5 (ISO8859-5)
UTF-16	8859-6 (ISO8859-6)
UTF-16	8859-7 (ISO8859-7)
UTF-16	8859-8 (ISO8859-8)

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UTF-16	8859-9 (ISO8859-9)
UTF-16	8859-10 (ISO8859-10)
UTF-16	8859-13 (ISO8859-13)
UTF-16	8859-14 (ISO8859-14)
UTF-16	8859-15 (ISO8859-15)
UTF-16	8859-15 (ISO8859-15)
UTF-16	8859-16 (ISO8859-16)
UTF-16	KOI8-R
UTF-16	KOI8-U
UTF-16	UCS-4
UTF-16	UCS-4BE
UTF-16	UCS-4LE
UTF-16	UTF-8
UTF-16BE	646 (ISO 646)
UTF-16BE	8859-1 (ISO8859-1)
UTF-16BE	8859-2 (ISO8859-2)
UTF-16BE	8859-3 (ISO8859-3)
UTF-16BE	8859-4 (ISO8859-4)
UTF-16BE	8859-5 (ISO8859-5)
UTF-16BE	8859-6 (ISO8859-6)
UTF-16BE	8859-7 (ISO8859-7)
UTF-16BE	8859-8 (ISO8859-8)
UTF-16BE	8859-9 (ISO8859-9)
UTF-16BE	8859-10 (ISO8859-10)
UTF-16BE	8859-13 (ISO8859-13)
UTF-16BE	8859-14 (ISO8859-14)
UTF-16BE	8859-15 (ISO8859-15)
UTF-16BE	8859-16 (ISO8859-16)
UTF-16BE	KOI8-R

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UTF-16BE	KOI8-U
UTF-16BE	UCS-4
UTF-16BE	UCS-4BE
UTF-16BE	UCS-4LE
UTF-16BE	UTF-8
UTF-16LE	646 (ISO 646)
UTF-16LE	8859-1 (ISO8859-1)
UTF-16LE	8859-2 (ISO8859-2)
UTF-16LE	8859-3 (ISO8859-3)
UTF-16LE	8859-4 (ISO8859-4)
UTF-16LE	8859-5 (ISO8859-5)
UTF-16LE	8859-6 (ISO8859-6)
UTF-16LE	8859-7 (ISO8859-7)
UTF-16LE	8859-8 (ISO8859-8)
UTF-16LE	8859-9 (ISO8859-9)
UTF-16LE	8859-10 (ISO8859-10)
UTF-16LE	8859-13 (ISO8859-13)
UTF-16LE	8859-14 (ISO8859-14)
UTF-16LE	8859-15 (ISO8859-15)
UTF-16LE	8859-16 (ISO8859-16)
UTF-16LE	KOI8-R
UTF-16LE	KOI8-U
UTF-16LE	UCS-4
UTF-16LE	UCS-4BE
UTF-16LE	UCS-4LE
UTF-16LE	UTF-8
UTF-32	UTF-8
UTF-32	UCS-2
UTF-32	UCS-2BE

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UTF-32	UCS-2LE
UTF-32	UCS-4
UTF-32	UCS-4BE
UTF-32	UCS-4LE
UTF-32	UTF-16
UTF-32	UTF-16LE
UTF-32	UTF-32BE
UTF-32	646 (ISO 646)
UTF-32	ISO8859-1
UTF-32	ISO8859-2
UTF-32	ISO8859-3
UTF-32	ISO8859-4
UTF-32	ISO8859-5
UTF-32	ISO8859-6
UTF-32	ISO8859-7
UTF-32	ISO8859-8
UTF-32	ISO8859-9
UTF-32	ISO8859-10
UTF-32	ISO8859-13
UTF-32	ISO8859-14
UTF-32	ISO8859-15
UTF-32	ISO8859-16
UTF-32	KOI8-R
UTF-32	KOI8-U
UTF-32BE	UTF-8
UTF-32BE	UCS-2
UTF-32BE	UCS-2BE
UTF-32BE	UCS-2LE
UTF-32BE	UCS-4

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UTF-32BE	UCS-4BE
UTF-32BE	UCS-4LE
UTF-32BE	UTF-16
UTF-32BE	UTF-16BE
UTF-32 BE	UTF-16LE
UTF-32BE	646 (ISO 646)
UTF-32BE	ISO8859-1
UTF-32BE	ISO8859-2
UTF-32BE	ISO8859-3
UTF-32BE	ISO8859-4
UTF-32BE	ISO8859-5
UTF-32BE	ISO8859-6
UTF-32BE	ISO8859-7
UTF-32BE	ISO8859-8
UTF-32BE	ISO8859-9
UTF-32BE	ISO8859-10
UTF-32BE	ISO8859-13
UTF-32BE	ISO8859-14
UTF-32BE	ISO8859-15
UTF-32BE	ISO8859-16
UTF-32BE	KOI8-R
UTF-32BE	KOI8-U
UTF-32LE	UTF-8
UTF-32LE	UCS-2
UTF-32LE	UCS-2BE
UTF-32LE	UCS-2LE
UTF-32LE	UCS-4
UTF-32LE	UCS-4BE
UTF-32LE	UCS-4LE

表 A-1 与 Unicode 相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UTF32-LE	UTF-16
UTF32-LE	UTF-16BE
UTF-32LE	UTF-16LE
UTF-32LE	646 (ISO 646)
UTF-32LE	ISO8859-1
UTF-32LE	ISO8859-2
UTF-32LE	ISO8859-3
UTF-32LE	ISO8859-4
UTF-32LE	ISO8859-5
UTF-32LE	ISO8859-6
UTF-32LE	ISO8859-7
UTF-32LE	ISO8859-8
UTF-32LE	ISO8859-9
UTF-32LE	ISO8859-10
UTF-32LE	ISO8859-13
UTF-32LE	ISO8859-14
UTF-32LE	ISO8859-15
UTF-32LE	ISO8859-16
UTF-32LE	KOI8-R
UTF-32LE	KOI8-U

注意 - UTF-EBCDIC 是一个新的 IBM 代码页名称。当前 Solaris 环境还支持 UTF-8 与 UTF-EBCDIC 代码页之间的双向转换。

下表列出了当前 Solaris 环境中可用的 Unicode 和 IBM/Microsoft EBCDIC 以及 PC iconv 代码转换模块。

表 A-2 与 Unicode 和 IBM/Microsoft EBCDIC 及 PC 代码页相关的 iconv 可用代码转换模块

从代码 (符号)	到代码 (符号)
UTF-8	IBM-037
UTF-8	IBM-273
UTF-8	IBM-277
UTF-8	IBM-278
UTF-8	IBM-280
UTF-8	IBM-284
UTF-8	IBM-285
UTF-8	IBM-297
UTF-8	IBM-420
UTF-8	IBM-424
UTF-8	IBM-500
UTF-8	IBM-850
UTF-8	IBM-852
UTF-8	IBM-855
UTF-8	IBM-856
UTF-8	IBM-857
UTF-8	IBM-862
UTF-8	IBM-864
UTF-8	IBM-866
UTF-8	IBM-869
UTF-8	IBM-870
UTF-8	IBM-871
UTF-8	IBM-875
UTF-8	IBM-880
UTF-8	IBM-1025
UTF-8	IBM-1026
UTF-8	IBM-1112
UTF-8	IBM-1122

表 A-2 与 Unicode 和 IBM/Microsoft EBCDIC 及 PC 代码页相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UTF-8	IBM-921
UTF-8	IBM-922
UTF-8	IBM-1046
UTF-8	IBM-1140
UTF-8	IBM-1141
UTF-8	IBM-1142
UTF-8	IBM-1143
UTF-8	IBM-1144
UTF-8	IBM-1145
UTF-8	IBM-1146
UTF-8	IBM-1147
UTF-8	IBM-1148
UTF-8	IBM-1149
UTF-8	CP850
UTF-8	CP852
UTF-8	CP855
UTF-8	CP857
UTF-8	CP862
UTF-8	CP864
UTF-8	CP866
UTF-8	CP869
UTF-8	CP874
UTF-8	CP1250
UTF-8	CP1251
UTF-8	CP1252
UTF-8	CP1253
UTF-8	CP1254

表 A-2 与 Unicode 和 IBM/Microsoft EBCDIC 及 PC 代码页相关的 iconv 可用代码转换模块 (续)

从代码 (符号)	到代码 (符号)
UTF-8	CP1255
UTF-8	CP1256
UTF-8	CP1257
UTF-8	CP1258

下表列出了可用的 iconv 代码转换, 可从 IBM 和 Microsoft EBCDIC/PC 代码页转换到 UTF-8。

表 A-3 可用的 iconv 代码转换—IBM 和 Microsoft EBCDIC/PC 代码页到 UTF-8

UTF-EBCDIC	UTF-8
IBM-037	UTF-8
IBM-273	UTF-8
IBM-277	UTF-8
IBM-278	UTF-8
IBM-280	UTF-8
IBM-284	UTF-8
IBM-285	UTF-8
IBM-297	UTF-8
IBM-420	UTF-8
IBM-424	UTF-8
IBM-500	UTF-8
IBM-850	UTF-8
IBM-852	UTF-8
IBM-855	UTF-8
IBM-856	UTF-8
IBM-857	UTF-8
IBM-862	UTF-8
IBM-864	UTF-8
IBM-866	UTF-8

表 A-3 可用的 iconv 代码转换—IBM 和 Microsoft EBCDIC/PC 代码页到 UTF-8 (续)

UTF-EBCDIC	UTF-8
IBM-869	UTF-8
IBM-870	UTF-8
IBM-871	UTF-8
IBM-875	UTF-8
IBM-880	UTF-8
IBM-921	UTF-8
IBM-922	UTF-8
IBM-1025	UTF-8
IBM-1026	UTF-8
IBM-1046	UTF-8
IBM-1112	UTF-8
IBM-1122	UTF-8
IBM-1140	UTF-8
IBM-1141	UTF-8
IBM-1142	UTF-8
IBM-1143	UTF-8
IBM-1144	UTF-8
IBM-1145	UTF-8
IBM-1146	UTF-8
IBM-1147	UTF-8
IBM-1148	UTF-8
IBM-1149	UTF-8
CP850	UTF-8
CP852	UTF-8
CP855	UTF-8
CP857	UTF-8
CP862	UTF-8
CP864	UTF-8
CP866	UTF-8

表 A-3 可用的 iconv 代码转换—IBM 和 Microsoft EBCDIC/PC 代码页到 UTF-8 (续)

UTF-EBCDIC	UTF-8
CP869	UTF-8
CP874	UTF-8
CP1250	UTF-8
CP1251	UTF-8
CP1252	UTF-8
CP1253	UTF-8
CP1254	UTF-8
CP1255	UTF-8
CP1256	UTF-8
CP1257	UTF-8
CP1258	UTF-8

索引

数字和符号

16 位 Unicode 3.0 代码集, 161

A

API, 国际化, 40

C

C 语言环境, 24

CDE

输入法, 110

语言环境支持, 108

.cshrc, STREAMS 模块设置, 143

ctype 宏, 40

D

DtMail, MIME 字符集, 144

dtterm, 142

E

en_US.UTF-8, FontSet 定义, 146

en_US.UTF-8, 支持, 108

F

FontSet 定义, 146

G

genmsg 实用程序, 47-48

GMT 偏移, 26

I

iconv

代码转换, 50

日语字符代码转换, 76

iconv 转换模块, Unicode, 185-206

iconv 转换模块, EBCDIC/PC 代码页, 209-211

ISO Latin-1, 23

ISO8859, 字符支持, 109

K

Kedmanee 键盘, 104

增强, 104

L

LANG 环境变量, 139

Latin-1, 书写键序列, 113

Latin-2, 书写键序列, 116

Latin-3, 书写键序列, 118

Latin-4, 书写键序列, 119
Latin-5, 书写键序列, 120
Latin-9, 书写键序列, 120
LC_ALL, 23

libc

API, 37-38
查询语言环境, 42
代码转换函数, 41
多字节处理, 44
货币格式, 44
宽字符串, 46
宽字符和字符串处理, 45
宽字符类, 42
宽字符输入和输出, 46, 47
日期和时间格式, 44
消息传送函数, 41
修改和查询语言环境, 42
应用程序链接, 35-50
正则表达式, 41
字符分类和拼写, 42
字符排序, 43

M

mbtowc, 40-47

Motif

TextField, 160
UIL 参数, 159
XmNalignment, 154, 160, 161
XmNeditPolicy, 154
XmNlayoutDirection, 151
XmNlayoutDirection, 160
XmNlayoutModifier, 160, 161
XmNrenditionTag, 154
XmRendition, 152, 160
XmStringDirection, 152
XmStringDirectionCreate, 158
XmText, 160
XmTextFieldGetLayoutModifier, 157
XmTextFieldSetLayoutModifier, 158
XmTextGetLayoutModifier, 157
XmTextSetLayoutModifier, 158

mp

PostScript 变量, 179
TrueType, 18
Xprt 客户机, 171
打印过滤器, 169, 170

mp.conf 文件, 172

P

Pattajoti 键盘, 105
POSIX 语言环境, 24
PostScript
 prolog 文件, 177
 运行时变量, 179
prolog 文件, 177

S

setlocale 命令, 139
SPARC 键盘, 63
strconf command, 142
STREAMS
 TTY 环境, 140
 代码转换, 140
 装入模块, 140
stty, 实用程序, 142
Sun Ray, USB 6 型键盘, 19

T

TTY 环境, 设置, 140
TypeOfText, 153

U

UIL, 159
Unicode
 概述, 107
 十六进制输入模式, 138
 转换模块, 185-206
UTC, 26
UTF-8, 支持, 108

X

X Print Xerver (Xprt), mp, 171
X 逻辑字体描述 (XLFD), 174

XmText
backward-cell(extend), 157
delete-left-character(), 156
delete-right-character(extend)
 , 156
forward-cell(extend), 157
left-character(extend), 155
right-character(extend), 155
right-word(extend), 156
XPG4 应用程序, 40
.xpr 文件, 177
xterm, 142

阿

阿拉伯语
 输入模式, 127
 字符支持, 109
阿拉伯语键盘, 64

埃

埃纳德语键盘, 79

本

本地化, 73-105
 定义, 21-23
 配置文件, 172
本地环境变量, 139

比

比利时语键盘, 64

编

编码集, 23

波

波兰语, 字符支持, 109

波罗的海语, 字符支持, 109

布

布局行为, 153

查

查表, 输入模式, 139

朝

朝鲜语
 朝鲜语汉字, 31
 朝鲜语文字, 31
 输入模式, 137
 字符支持, 109
朝鲜语汉字, 31
朝鲜语键盘, 67
朝鲜语文字, 31

打

打印过滤器, 169

代

代码集
 转换, 143
 字符支持, 109
代码集独立性 (CSI)
 ASCII 斜线, 36
 Java 国际化, 36
 NULL 字节, 36
 Shift-JIS 代码集, 35-37
 多字节字符, 38
 进程代码格式, 38
 库, 37
 扩展 UNIX 代码 (EUC), 35-37
 命令, 36-37
 语言环境数据库, 37-38
代码转换模块, 185-206, 207-209

丹

丹麦语键盘, 65

德

德语, 字符支持, 109

德语键盘, 66

多

多字节, 转换, 40-47

多字节字符, 38

俄

俄语, 字符支持, 109

法

法语键盘, 65

繁

繁体中文

输入模式, 137

字符支持, 109

繁体中文键盘, 69

芬

芬兰语键盘, 65

复

复杂文本布局 (CTL), 14, 149-167

Motif, 151-159

Motif 库, 166-167

XOC 资源, 150

XOM, 150

编辑绘制, 162

布局方向, 160-163

复杂文本布局 (CTL) (续)

创建绘制, 161

创建绘制表, 163-164

键盘选定, 166

连写符, 149-167

鼠标选定, 165

水平制表符, 164-165

体系结构, 150

文本定向, 149-167

文本资源, 166

古

古吉拉特语键盘, 78

国

国际化

ISO Latin-1, 23

定义, 21-23

国际化 API, 40

国际化域名 (IDN), 49

果

果鲁穆奇语键盘, 79

汉

汉字, 31

荷

荷兰 (荷兰语) 键盘, 67

简

简体中文

输入模式, 137

字符支持, 109

键

键盘

- 4 型、5 型和 5c 型, 61
- 6 型, 60
- CTL 选定, 166
- Kedmanee, 104
- Kedmanee 增强, 104
- Pattajoti, 105
- 阿拉伯语, 64, 127
- 埃纳德语, 79
- 比利时语, 64
- 布局, 63, 64, 78, 104
- 朝鲜语, 67, 137
- 丹麦语, 65
- 德语, 66
- 地区, 59
- 法语, 65
- 繁体中文, 69
- 芬兰语, 65
- 更改 SPARC 上的键盘设置, 63
- 更改布局, 61
- 古吉拉特语, 78
- 果鲁穆奇语, 79
- 荷兰 (荷兰语), 67
- 将布局更改为捷克语布局, 63
- 马拉雅拉姆语, 80
- 美国, 71
- 美国/UNIX, 71
- 孟加拉语, 78
- 挪威语, 67
- 葡萄牙语, 68
- 日语, 66, 136
- 瑞典语, 68
- 瑞士 (德语), 69
- 瑞士 (法语), 69
- 泰卢固语, 80
- 泰米尔语, 80
- 土耳其语 F 型, 70
- 土耳其语 Q 型, 70
- 西班牙语, 68
- 西里尔语, 64
- 西里尔语 (俄语), 128
- 希伯来语, 136
- 希腊语 UNIX, 129
- 希腊语欧式, 129
- 意大利语, 66
- 英国, 70
- 梵文, 78

脚

- 脚本选择, 110

捷

- 捷克语, 字符支持, 109

静

- 静态链接, 38-39

可

- 可移植布局服务 (PLS), 172

库

- 库, (CSI), 37

宽

- 宽字符, 支持, 40-47

连

- 连续拼音输入法, 印度语脚本, 103

链

- 链接, 应用程序, 38-39

马

- 马拉雅拉姆语键盘, 80

美

- 美国/UNIX 键盘, 71
- 美国键盘, 71

孟

孟加拉语键盘, 78

命

命令, CSI, 36-37

挪

挪威语键盘, 67

片

片假名, 31

拼

拼音, 32

平

平假名, 31

葡

葡萄牙语键盘, 68

日

日期格式, 27

日语

iconv模块, 76

本地化, 73-76

片假名, 31

平假名, 31

日语汉字, 31

输入法, 75

输入模式, 136

完全本地化软件包, 76

语言环境, 73

字符集, 73-74

日语 (续)

字符支持, 109

字体, 74-75

日语汉字, 31

日语键盘, 66

瑞

瑞典语键盘, 68

瑞士 (德语) 键盘, 69

瑞士 (法语) 键盘, 69

时

时间格式, 26

时区, 27

实

实用程序

genmsg, 48

genmsg, 47-48

iconv, 49

stty, 142

语言环境, 139

输

输入法

ATOK 方法, 75

日语, 75

泰语, 105

印度语, 81

印度语方法, 77

输入模式

en_US.UTF-8 语言环境, 110

Unicode 十六进制, 138

阿拉伯语, 127

查表, 139

朝鲜语, 137

繁体中文, 137

简体中文, 137

日语, 136

西里尔语, 127

输入模式 (续)

- 希伯来语, 135
- 希腊语, 128, 129
- 英语, 112

书

- 书写键, 33
- 书写键序列
 - Latin-1, 113
 - Latin-2, 116
 - Latin-3, 118
 - Latin-4, 119
 - Latin-5, 120
 - Latin-9, 120
- 希腊语, 129
- 希腊语, 三个键, 134
- 希腊语, 四个键, 135
- 重音死键, 120

斯

- 斯堪的纳维亚, 字符支持, 109

泰

- 泰卢固语键盘, 80
- 泰米尔语键盘, 80
- 泰语, 31
 - 键盘布局, 20
 - 输入法, 104, 105
 - 字符序列检查, 104-105
 - 字符支持, 109

土

- 土耳其语, 字符支持, 109
- 土耳其语 F 型键盘, 70
- 土耳其语 Q 型键盘, 70

完

- 完全本地化软件包, 日语, 76

西

- 西班牙语, 68
 - 字符支持, 109
- 西里尔语, 输入模式, 127
- 西里尔语键盘, 64
- 西欧语言
 - 字符支持, 109
 - 字符支持, 109

希

- 希伯来语
 - 输入模式, 135
 - 意第绪语, 32
 - 字符支持, 109
- 希腊语
 - 输入模式, 128, 129
 - 字符支持, 109
- 希腊语 UNIX, 键盘, 129
- 希腊语欧式, 键盘, 129

系

- 系统库, 将应用程序链接至, 38-39

夏

- 夏时制时间 (DST), 27

消

- 消息目录, 47-48

匈

- 匈牙利语, 字符支持, 109

意

- 意大利语键盘, 66

印

印度语

- 输入法, 77
- 天城书, 32
- 字符支持, 109

英

- 英国键盘, 70
- 英文, 埃纳德语字符, 92
- 英语
 - 输入模式, 112
 - 字符支持, 109

应

应用程序

- FontSet/XmFontList 定义, 146
- XPG4, 40
- 链接至系统库, 38-39

映

映射

- Telugu 字符, 101
- 古吉拉特语字符, 84
- 果鲁穆奇语字符, 87
- 马拉雅拉姆语字符, 95
- 孟加拉语字符, 81
- 泰米尔语字符, 98
- 印度语字符, 90
- 英文至印度语脚本的等效拼音, 81

语

语言环境

- C, 24
- POSIX, 24
- Solaris, 73-105
- 澳大拉西亚, 53
- 北非, 56
- 北美, 56
- 北欧, 56
- 部分, 24, 51

语言环境 (续)

- 单词分隔符, 30
- 定义, 23
- 东欧, 54
- 环境变量, 139
- 货币, 21
- 货币格式, 28
- 键盘差异, 33
- 南美, 57
- 南欧, 57
- 排序顺序, 30
- 日期格式, 27
- 日语, 73
- 时间格式, 26
- 书写键, 58
- 数字格式, 27
- 完全, 24, 51
- 文化惯例, 25
- 西欧, 58
- 亚洲, 52
- 纸张大小, 33
- 中东, 55
- 中美洲, 54
- 种类, 25
- 字符集, 30

纸

- 纸张大小, 通常大小, 33

中

- 中国, 32
- 中文
 - 汉语拼音字母, 32
 - 汉字, 32
 - 拼音, 32
 - 输入法, 20
 - 香港特别行政区, 32
 - 中国, 32
 - 中国台湾, 32
 - 注音, 32

终

终端

- 设置选项, 142
- 支持, 142

转

转换

- Unicode iconv 模块, 185-206
- 用户自定义代码集, 48

字

字符

- 多字节, 38
- 形状, 149-167
- 支持, 109
- 转换, 40-47

字符编码

- Unicode, 107
- UTF-16, 107
- UTF-32, 107
- UTF-8, 107

字符集独立性 (CSI)

- 动态链接应用程序, 38-39
- 文件代码编码, 36

字母表, 30

字体

- 别名, 172
- 日语 TrueType, 75
- 日语位图, 74
- 映射, 172

梵

梵文键盘, 78

