



國際語言環境指南

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

文件號碼: 819-0403-10
2005 年 1 月

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 版權所有

此產品或文件受著作權的保護，其使用、複製、分送與取消編譯均受軟體使用權限制。未經Sun及其授權許可頒發機構的書面授權，不得以任何方式、任何形式複製本產品或本文件的任何部分。至於協力廠商的軟體，包括字型技術，亦受著作權保護，並經過 Sun 供應商授權使用。

本產品的某些部分從 Berkeley BSD 系統衍生而來，經 University of California 許可授權。UNIX 是在美國和其他國家/地區的註冊商標，由 X/Open Company, Ltd. 獨家授權。

Sun、Sun Microsystems、Sun 標誌、docs.sun.com、AnswerBook、AnswerBook2、SunOS 是 Sun Microsystems, Inc. 在美國和其他國家/地區的商標或註冊商標。以及 Solaris 都是 Sun Microsystems, inc. 在美國和其它國家/地區的商標、註冊商標或服務標記。所有的 SPARC 商標都是在獲得授權的情況下使用，而且是 SPARC International, Inc. 在美國和其他國家/地區的商標或註冊商標。有 SPARC 商標的產品均基於 Sun Microsystems, Inc. 所開發的基本架構。Netscape Navigator 是 Netscape Communications Corporation 在美國和其他國家/地區的商標或註冊商標。PostScript 是 Adobe Systems, Incorporated 的商標或註冊商標，可能以特定管轄權註冊。

OPEN LOOK 和 Sun™ Graphical User Interface 是 Sun Microsystems Inc. 為其使用者和授權許可持有人而開發的。Sun 認可 Xerox 研發電腦業之視覺化或圖形化使用者介面觀念的先驅貢獻。對於「Xerox 圖形使用者介面」，Sun 保有來自於 Xerox 的非獨家授權，這項授權的適用也涵蓋取得 Sun 的授權而使用 OPEN LOOK GUI、或者遵循 Sun 的書面授權合約的廠商。

本說明文件以「現狀」提供，所有明示或暗示的條件、陳述與保證，包括對於適銷性、特定用途的適用性或非侵權行為的任何暗示性保證在內，均恕不負責，除非此免負責聲明在法律上被認為無效。



050118@10536



目錄

前言	13
1 Solaris 國際化概述	17
關於 Solaris 國際化架構	17
新的國際化與本土化功能	18
國際化與本土化概述	21
國際化的基本步驟	21
Solaris 介面中的本土化功能	22
何謂語言環境？	23
C 語言環境 - 預設語言環境	24
完整與部份語言環境	24
受語言環境影響的行為	25
語言環境的種類	25
為本土化使用語言環境種類	26
時間格式	26
日期格式	27
數字格式	27
國際性貨幣格式	28
語言文字與字母的差異	30
文字分割元	30
排序	30
字元集	30
鍵盤差異	33
紙張大小的差異	33

2	一般國際化功能	35
	字碼集獨立性的支援	35
	CSI 方法	36
	已啓用 CSI 的指令	36
	已啓用 CSI 的程式庫	37
	語言環境資料庫	37
	程序碼格式	38
	多位元組支援環境	38
	動態連結應用程式	38
	已變更介面	39
	ctype 巨集	40
	libc 的國際化 API	40
	genmsg 公用程式	47
	使用者自行定義和使用者可擴充字碼的轉換	48
	國際化網域名稱 (IDN) 支援	49
3	Solaris 環境中的本土化	51
	本土化的軟體支援	51
	Solaris 語言環境套裝軟體摘要	51
	受支援的語言環境	52
	語言環境的多重按鍵組合序列	58
	Solaris 環境中的鍵盤支援	59
	SPARC 系統上各種鍵盤之間的變更	60
	在 Intel 系統上變更鍵盤	63
	鍵盤配置圖例	63
	新的 Solaris 鍵盤軟體支援	71
	▼ 如何存取愛沙尼亞文 Type 6 USB 鍵盤支援	72
	▼ 如何存取加拿大法語 Type 6 USB 鍵盤支援	72
	▼ 如何存取波蘭程式設計師 Type 5 鍵盤支援	72
4	支援的亞洲語言環境	73
	日文本土化	73
	日文語言環境	73
	日文字元集	73
	日文字型	74
	日文輸入系統	75
	▼ 如何使用 ATOK 輸入法	75

日文終端機的終端機設定	75
日文 iconv 模組	76
使用者定義的字元支援	76
部份和完整語言環境之間的差異	76
印度文本土化	77
▼ 如何使用印度文輸入法	77
印度文鍵盤	78
瞭解對映	81
以連續語音為基礎之輸入法的對映	81
連續語音輸入法的運作方式	103
泰文本土化	104
泰文輸入法	104
泰文鍵盤配置	104
泰文輸入法輔助視窗	105
5 UTF-8 語言環境支援概述	107
Unicode 概述	107
Unicode 語言環境：en_US.UTF-8 支援	108
關於桌面輸入法	110
字體選擇和輸入模式	110
使用輸入模式	111
輸入模式切換鍵序列	112
英文/歐洲輸入模式	113
阿拉伯文輸入模式	128
斯拉夫文輸入模式	128
希臘文輸入模式	129
希伯來文輸入模式	136
日文輸入模式	137
韓文輸入模式	137
簡體中文輸入模式	138
繁體中文輸入模式	138
繁體中文 (香港) 輸入模式	138
Unicode 十六進位輸入模式	139
查表輸入模式	139
系統環境	140
語言環境變數	140
TTY 環境設定	140
字碼轉換	144

- DtMail 支援 145
- 程式設計環境 147
 - 使用 X 應用程式的 FontSet 148
 - CDE/Motif 應用程式中的 FontList 定義 148
- 6 複合文字版面配置 149**
 - CTL 技術概述 149
 - CTL 架構概述 150
 - 基於 X Library 的應用程式的 CTL 支援 150
 - XOC 資源 150
 - 支援 CTL 技術的 Motif 變更 151
 - XmNlayoutDirection 資源 151
 - XmStringDirection 資源 152
 - XmRendition 資源 152
 - XmText 和 XmTextField 資源 154
 - XmTextFieldGetLayoutModifier 資源 157
 - XmTextGetLayoutModifier 資源 158
 - XmTextFieldSetLayoutModifier 資源 158
 - XmTextSetLayoutModifier 資源 158
 - XmStringDirectionCreate 資源 159
 - UIL 引數 159
 - 開發 CTL 應用程式 160
 - 控制版面配置方向 160
 - 在資源檔案中建立 Render 表格 163
 - 水平定位點 164
 - 滑鼠選取 165
 - 鍵盤選取 166
 - 文字資源與幾何學 166
 - 移植指令 166
- 7 藉由 mp 所進行的列印過濾器增強 169**
 - UTF-8 的列印 169
 - mp 列印過濾器增強概述 170
 - 以語言環境特定的字型配置檔 mp.conf 配合 mp 來使用 170
 - 藉由語言環境特定的 PostScript Prologue 檔案來使用 mp 171
 - 將 mp 視為 Xprt (X 列印伺服器) 用戶端來使用 171
 - 使用 mp.conf 配置檔進行本土化 171

- ▼ 如何新增印表機內建字型 175
- ▼ 如何建立共用物件檔案 176
- 新增及自訂 prolog 檔案 177
 - PostScript 檔案自訂 177
 - .xpr 檔案 180

A iconv 字碼轉換 185

索引 213

表清單

表 1-1	合法的 UTF-8 位元組序列	20
表 1-2	國際化時間格式	26
表 1-3	國際日期格式	27
表 1-4	國際化數值慣例	28
表 1-5	國際幣制慣例	28
表 1-6	支援歐元貨幣的使用者語言環境	29
表 1-7	德文語言環境以及相對應的 LC_MONETARY 運算元	30
表 1-8	一般國際化頁面大小	33
表 2-1	libc 中的訊息函式	41
表 2-2	libc 中的字碼轉換	41
表 2-3	libc 中的常規表示式	41
表 2-4	libc 中的寬字元類別	42
表 2-5	libc 中的修改和查詢語言環境	42
表 2-6	libc 中的查詢語言環境資料	42
表 2-7	libc 中的字元分類和轉譯	42
表 2-8	libc 中的字元比較	43
表 2-9	libc 中的貨幣格式	44
表 2-10	libc 中的日期與時間格式	44
表 2-11	libc 中的多位元組處理	44
表 2-12	libc 中的寬字元和字串處理	45
表 2-13	libc 中的格式化寬字元輸入與輸出	46
表 2-14	libc 中的寬字串	46
表 2-15	libc 中的寬字元輸入與輸出	47
表 2-16	iconv 字碼轉換	50
表 3-1	亞洲語言環境	52
表 3-2	澳大拉西亞語言環境	53
表 3-3	中美洲語言環境	54

表 3-4	中歐語言環境	54
表 3-5	東歐語言環境	54
表 3-6	中東語言環境	55
表 3-7	北非語言環境	55
表 3-8	北美語言環境	56
表 3-9	北歐語言環境	56
表 3-10	南美語言環境	56
表 3-11	南歐語言環境	57
表 3-12	西歐語言環境	58
表 3-13	使用 Compose 按鍵所建立的分音符字元	58
表 3-14	區域性鍵盤支援	59
表 3-15	Type 4、5 和 5c 鍵盤的配置	61
表 4-1	日文點陣圖字型	74
表 4-2	日文 TrueType 字型	75
表 5-1	輸入模式切換鍵序列	113
表 5-2	常用 Latin-1 Compose 鍵序列	114
表 5-3	常用的 Latin-2 Compose 鍵序列	118
表 5-4	常用的 Latin-3 Compose 鍵序列	119
表 5-5	常用的 Latin-4 Compose 鍵序列	120
表 5-6	常用的 Latin-5 Compose 鍵序列	121
表 5-7	常用 Latin-9 Compose 鍵序列	121
表 5-8	以重音固定鍵為基礎的 Compose 鍵序列	122
表 5-9	希臘文輸入模式的 Compose 鍵序列	130
表 5-10	希臘文輸入模式中的三鍵 Compose 鍵序列	134
表 5-11	希臘文輸入模式中的四鍵 Compose 鍵序列	136
表 5-12	en_US.UTF-8 所支援的 STREAMS 模組	141
表 5-13	en_US.UTF-8 所支援的 64 位元 STREAMS 模組	141
表 6-1	XmRendition 中的新資源	153
表 6-2	Xm CTL 中的新資源	154
表 6-3	UIL	159
表 7-1	可選關鍵字/值對	175
表 7-2	STARTCOMMON/ENDCOMMON 關鍵字值	183
表 A-1	與 Unicode 有關的可用 iconv 字碼轉換模組	185
表 A-2	與 Unicode、IBM/Microsoft EBCDIC 和 PC 字碼頁有關的 iconv 字碼轉換模組	207
表 A-3	可用的 iconv 字碼轉換 - 從 IBM 和 Microsoft EBCDIC/PC 字碼頁轉換為 UTF-8	209

圖清單

圖 1-1	Solaris 作業系統中語言環境的功能與結構	21
圖 2-1	IDN 轉換為 ACE	49
圖 2-2	ACE 轉換為 IDN	49
圖 3-1	阿拉伯文鍵盤	63
圖 3-2	比利時文鍵盤	64
圖 3-3	斯拉夫文 (俄文) 鍵盤	64
圖 3-4	丹麥文鍵盤	65
圖 3-5	芬蘭文鍵盤	65
圖 3-6	法文鍵盤	65
圖 3-7	德文鍵盤	66
圖 3-8	義大利文鍵盤	66
圖 3-9	日文鍵盤	66
圖 3-10	韓文鍵盤	67
圖 3-11	荷蘭 (荷蘭文) 鍵盤	67
圖 3-12	挪威文鍵盤	67
圖 3-13	葡萄牙文鍵盤	68
圖 3-14	西班牙文鍵盤	68
圖 3-15	瑞典文鍵盤	68
圖 3-16	瑞士 (法文) 鍵盤	69
圖 3-17	瑞士 (德文) 鍵盤	69
圖 3-18	繁體中文鍵盤	69
圖 3-19	土耳其文 F 鍵盤	70
圖 3-20	土耳其文 Q 鍵盤	70
圖 3-21	英國鍵盤	70
圖 3-22	美國鍵盤	71
圖 3-23	美國/UNIX 鍵盤	71
圖 4-1	孟加拉文字音字母的對映	81

圖 4-2	孟加拉文母音字母的對映	82
圖 4-3	孟加拉文其他字元的對映	83
圖 4-4	古吉拉特文子音字母的對映	84
圖 4-5	古吉拉特文母音字母的對映	85
圖 4-6	古吉拉特文其他字元的對映	86
圖 4-7	果魯穆其文子音字母的對映	87
圖 4-8	果魯穆其文母音字母的對映	88
圖 4-9	果魯穆其文其他字元的對映	89
圖 4-10	北印度文子音字母的對映	90
圖 4-11	北印度文母音字母的對映	90
圖 4-12	北印度文其他字元的對映	91
圖 4-13	卡納達文子音字母的對映	92
圖 4-14	卡納達文母音字母的對映	93
圖 4-15	卡納達文其他字元的對映	94
圖 4-16	喀拉拉文子音字母的對映	95
圖 4-17	喀拉拉文母音字母的對映	96
圖 4-18	喀拉拉文其他字元的對映	97
圖 4-19	坦米爾文子音字母的對映	98
圖 4-20	坦米爾文母音字母的對映	99
圖 4-21	安得拉邦文子音字母的對映	101
圖 4-22	安得拉邦文母音字母的對映	101
圖 4-23	安得拉邦文其他字元的對映	102
圖 5-1	輸入模式選擇視窗	111
圖 5-2	阿拉伯文鍵盤	128
圖 5-3	斯拉夫文 (俄文) 鍵盤	128
圖 5-4	希臘文 Euro 鍵盤	129
圖 5-5	希臘文 UNIX 鍵盤	130
圖 5-6	希伯來文鍵盤	136
圖 5-7	日文鍵盤	137
圖 5-8	韓文鍵盤	137
圖 5-9	DtMail 新訊息視窗	146
圖 6-1	CTL 架構	150
圖 6-2	版面配置方向	160
圖 6-3	定位點行爲	164

前言

此「國際語言環境指南」將為您說明 Solaris™ 作業系統 (Solaris OS) 中新增的國際化功能。本指南包含了相當多的重要資訊，以指導您如何使用目前 Solaris 發行版本來建立支援各種語言和文化慣例的全球軟體產品。

您可以找到語言開發和支援等方面的概念性資訊。您也可以找到促進全球應用程式開發以及全球語言服務管理的 Solaris 功能的說明。

此外，有關本發行版本中國際化功能的更多資訊，本指南提供參考其他說明文件的連結。

注意 – 本指南中所有與作業系統有關的資訊，都是屬於 Solaris 作業系統範疇。

本前言包含下列內容：

- 第 14 頁的「關於本書」
- 第 14 頁的「本指南的編排方式」
- 第 14 頁的「相關的書籍與網站」
- 第 15 頁的「線上存取 Sun 文件」
- 第 15 頁的「印刷排版慣例」
- 第 16 頁的「指令範例中的 Shell 提示符號」

注意 – 此 Solaris 發行版本支援使用 SPARC® 及 x86 系列處理器架構的系統：UltraSPARC®、SPARC64、AMD64、Pentium 和 Xeon EM64T。支援的系統列示於「*Solaris 10 Hardware Compatibility List*」中，網址為 <http://www.sun.com/bigadmin/hcl>。本文件引用平台類型之間的執行差異。

在本文中，「x86」是指 64 位元和 32 位元的系統，使用與 AMD64 或 Intel Xeon/Pentium 產品系列相容處理器製造完成的。如需有關支援之系統的資訊，請參閱「*Solaris 10 Hardware Compatibility List*」。

關於本書

本指南是針對使用目前 Solaris 作業系統來設計和支援全球應用程式的軟體開發人員及系統管理員而編寫的。

本指南假定您具備了使用 C 程式設計語言的知識。

本指南的編排方式

本指南中章節的組織方式如下：

- 第 1 章說明目前 Solaris 發行版本中新增的國際化與本土化功能。
- 第 2 章提供字碼集獨立性 (CSI)、語言環境資料庫、libc API 以及其他國際化功能的介紹資訊。
- 第 3 章提供在目前 Solaris 作業系統中使用語言環境、字型和鍵盤等支援的資訊。
- 第 4 章說明目前 Solaris 發行版本中所提供的日文、北印度文和泰文的本土化支援。
- 第 5 章提供在目前 Solaris 作業系統版本中使用輸入法和字碼轉換功能等支援的資訊。
- 第 6 章說明複合文字版面配置 (CTL) 擴充，它可以使 Motif API 支援邏輯和實體文字間所需的繁複變換過程的寫入系統。需要繁複轉換程序的書寫系統包括阿拉伯文、希伯來文和泰文。
- 第 7 章說明 mp 列印篩選器的列印支援重點。
- 附錄 A 包含可用的 iconv 轉換表格。

相關的書籍與網站

下列書籍提供本指南所討論主題的進一步資訊：

- Solaris 國際化：
作者：Tuthill、Bill 以及 David Smallberg。書名：「*Creating Worldwide Software: Solaris International Developer's Guide*」第 2 版。出版商：Mountain View, California, Sun Microsystems Press, 1997。您可透過 books@sun.com 及 <http://www.sun.com/books/> 訂購本書。本書提供 Solaris 作業系統中國際化程序的一般概述。
- Solaris 一般桌上管理系統環境：

「*Solaris Common Desktop Environment: Programmer's Guide*」是隨附於 Solaris Documentation CD 的 CDE Developer's Collection 中的部分內容。

- 中文與韓文 Solaris 語言環境：
 - 「韓文 *Solaris* 使用者指南」
 - 「簡體中文 *Solaris* 使用者指南」
 - 「繁體中文 *Solaris* 使用者指南」
- OSF/Motif 應用程式開發：
 - 「*OSF/Motif Programmer's Guide, Release 1.2*」，Englewood Cliffs，New Jersey，Prentice-Hall，1993。本書是開放軟體基金會 (OSF) 針對如何使用 OSF/Motif 應用程式的程式設計介面來建立 Motif 應用程式的指南。

線上存取 Sun 文件

docs.sun.comSM 網站可讓您存取 Sun 線上技術文件。您可以瀏覽 docs.sun.com 的歸檔檔案或搜尋特定書名或主題。其 URL 是 <http://docs.sun.com>。

印刷排版慣例

下表說明本書在印刷上所作的變更。

表 P-1 印刷排版慣例

字體或符號	涵義	範例
AaBbCc123	指令、檔案和目錄的名稱；電腦螢幕的輸出	編輯您的 .login 檔案。 使用 <code>ls -a</code> 列出所有檔案。 machine_name% you have mail.
AaBbCc123	您輸入的內容，與電腦螢幕上的輸出相對照	machine_name% su Password:
<i>AaBbCc123</i>	指令行預留位置：用實際名稱或值取代	若要刪除檔案，請輸入 rm 檔案名稱。

表 P-1 印刷排版慣例 (續)

字體或符號	涵義	範例
<i>AaBbCc123</i>	書名、新字詞、專有名詞或要強調的字	請閱讀「 使用者手冊 」中的第 6 章。 這些稱為 類別 選項。 您必須是 <i>root</i> ，才能執行此動作。

指令範例中的 Shell 提示符號

以下表格列出使用於 C shell、Bourne shell 和 Korn shell 的預設系統提示符號以及超級使用者提示符號。

表 P-2 Shell 提示符號

Shell	提示符號
C shell 提示符號	machine_name%
C shell 超級使用者提示符號	machine_name#
Bourne shell 和 Korn shell 提示符號	\$
Bourne shell 和 Korn shell 超級使用者提示符號	#

第 1 章

Solaris 國際化概述

本章將說明 Solaris 國際化和本土化的新功能與關鍵概念。本章包含以下主題。

- 第 18 頁的「新的國際化與本土化功能」
- 第 21 頁的「國際化與本土化概述」
- 第 23 頁的「何謂語言環境？」
- 第 26 頁的「為本土化使用語言環境種類」
- 第 30 頁的「語言文字與字母的差異」
- 第 33 頁的「鍵盤差異」
- 第 33 頁的「紙張大小的差異」

關於 Solaris 國際化架構

目前 Solaris 發行版本有許多新功能，包括 UTF-8 語言環境的 Unicode 4.0 支援、增強型鍵盤支援以及多項 mp 列印過濾器改良功能。

透過 Solaris 國際化架構，您可以輕鬆地開發、部署以及管理全球的應用程式和語言服務。此單一多國語言產品提供了 39 種語言和 162 個語言環境的支援。此外，可以支援泰文和北印度文字體所需的複雜文字版面配置。也有支援阿拉伯文和希伯來文等語言的雙向文字能力。

支援許多不同 Solaris 語言環境的輸入法、字元集、字碼集轉換以及其他與語言有關的功能。您可以使用下列標準 API，在多重語言環境中部署應用程式。您也可以 Solaris 環境中自訂語言屬性、變更轉換器表格或增加新的輸入法編輯器。

Solaris X 全球化架構的原始碼，已在 2000 年秋天釋出到開放原始碼社群。此舉讓您得以遵循一個共同參考介面，來增強全球化應用程式的相容性與互通性。字碼集獨立的全局化方法，讓您在原生語言和 Unicode 語言環境中都可以作業。Solaris 架構提供擴展至各平台的能力。豐富的資料轉換集確保了多種編碼與不同協力廠商平台之間的互通性。

Solaris 平台也使得跨國公司將其伺服器管理擴展至世界各地。與其他相競爭的平台不同，Solaris 平台是透過以服務為基礎的方法來進行語言服務的管理。無論用戶端系統為何，伺服器管理者都可以透過全球網路以遠端方式啟用語言服務。此獨立於用戶端之外的方法，讓您無須變更用戶端應用程式即可進行系統升級。例如，使用者不必為了讀取從巴黎網咖寄來的阿拉伯文電子郵件，而變更本機的用戶端應用程式。

新的國際化與本土化功能

在目前 Solaris 發行版本中包含下列新功能：

- 自動編碼尋找程式

自動編碼尋找程式是用於處理全球字元的公用程式。透過一般用途介面，自動編碼尋找程式提供一種可輕鬆偵測特定檔案或字串編碼方法的途徑。編碼偵測可簡化對各種語言字元編碼的讀取程序。如需詳細資訊，請參閱「`auto_ef(1)` 線上說明手冊」或「`libauto_ef(3LIB)` 線上說明手冊」。

- 語言環境管理員

語言環境管理員可讓您透過指令行介面查詢與設定 Solaris 作業系統的語言環境。您可以使用 `localeadm(1M)` 工具，來顯示系統上已安裝的或是特定裝置或目錄上常駐之語言環境套裝軟體的資訊。您可以以區域為基礎，新增與移除目前系統中的語言環境。例如，您可以將「東歐」地區中的所有語言環境新增至目前的系統中。

在使用語言環境管理員之前，如果系統已安裝完成，則您必須新增/移除單獨的套裝軟體來變更機器上的語言環境。因為套裝軟體容易遺失或被忽略，使用單獨套裝軟體時經常會出現錯誤。

語言環境管理員是 Solaris 安裝程式中語言環境選擇邏輯的補充。安裝程式仍然是正確安裝 Solaris 語言環境的主要應用程式。

- `mp` 增強功能

`mp` 列印過濾器首見於 Solaris 9 作業系統，用來取代目前 Solaris 發行版本中的 `xutops` 列印過濾器。此發行版本中的 `mp` 列印過濾器，透過下列主要改良功能而增強了。

- 如果在 `mp.conf` 檔案中找不到已設定的字型，`mp` 程式將會繼續執行，直到出現使用未認可字型的文字影像。
- 當列印 TrueType 文字影像時，`mp` 輸出的大小會大量地減少。
- 為了更快速列印，會適當地調整 `mp` 所使用的字典機制。
- TrueType 引擎已增強，可以對所有各種寬度之空格字元進行處理。

注意 – Solaris 作業系統已不再支援 `xutops` 列印過濾器。 `xutops` 列印過濾器以往是用來在 UTF-8 語言環境中列印國際化文字。取代了 `xutops` 的 `mp` 列印過濾器，也就是 `xutops` 列印過濾器受支援功能的超集合。如需詳細資訊，請參閱「`mp(1)` 線上說明手冊」。

- 全新歐洲鍵盤支援

Sun I/O 鍵盤可支援波蘭程式設計師 Type 5 鍵盤和 Sun Ray™ USB Type 6 的俄文、愛沙尼亞文以及加拿大法文鍵盤。

注意 – 目前還沒有可供全新歐洲鍵盤類型使用的硬體。若要使用全新鍵盤的軟體，請參閱第 71 頁的「新的 Solaris 鍵盤軟體支援」中的程序。

- Unicode 4.0 支援

下列 UTF-8 語言環境已更新為可支援全新 4.0 版本的 Unicode 標準：

- `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`

此標準的新版本加入了 1,226 個其他新字元，並且包含多種基準的變更和廣泛的變更。

Unicode 3.2 定義了更嚴謹的 UTF-8 位元組序列以做為「UTF-8 勘誤表」：

表 1-1 合法的 UTF-8 位元組序列

字碼點	第 1 位元組	第 2 位元組	第 3 位元組	第 4 位元組
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	格式錯誤			
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

這些序列排除從 U+D800 到 U+DFFF 之間的代理字碼點。此序列也禁止任何其他非法的位元組值。為了遵循此新定義，Unicode 語言環境方法和 UTF-8 iconv 模組都已增強，可以對最新定義的 UTF-8 無效位元組序列進行偵測。

■ 泰文鍵盤配置

下列是支援泰文輸入法的鍵盤配置：

- Kedmanee (TIS820-2531) 鍵盤配置。Kedmanee 配置是針對打字機設計的，不適用於電腦鍵盤。由於打字機鍵盤上的按鍵有限，因此這個配置無法使用部分泰文特殊字元。TIS820-2531 在電腦鍵盤上採用 Kedmanee 配置。
- TIS820-2538 鍵盤配置。增強型 Kedmanee 配置是 TIS820-2531 配置的更新版本，其中包含原本無法在舊版 Kedmanee 配置中使用的部分泰文特殊字元。TIS820-2538 是目前唯一由泰國工業標準局發佈的泰文鍵盤配置標準。
- Pattajoti 鍵盤配置。Pattajoti 配置也是針對打字機設計的，但具有較佳的手指受力分佈區域。Pattajoti 是由 Royal Irrigation Department 的官員所發明的，並且仍廣泛使用於該部門中。
- 可配置的鍵盤配置，使用者可自行定義泰文輸入法的鍵盤配置。

■ 印度語言的輸入法支援

此發行版本包含一個與中文輸入法所用相類似的字碼表輸入法介面。以 IIIMF SDK 與 SunIM 語言介面為基礎的印度文輸入法，提供下列新功能：

- 以語音、字母轉譯為基礎的輸入法和鍵盤配置。受支援的鍵盤配置在 ISCII 標準中被定義為 INSCRIPT 鍵盤外觀。
- 標準輸入法切換。
- 印度文字體包括北印度文、坦米爾文、卡納達文、馬來亞拉姆文、特拉古文、古吉拉特文、旁遮普文和孟加拉文。您可以透過按下 F5 鍵來變更輸入字體的種類。

- 新鍵盤配置的 Plug-in 機制可在未來輕鬆達成擴充之效。語言引擎模組將會載入 Plug-in 設定檔。

國際化與本土化概述

國際化與本土化是不同的過程。**國際化**是使軟體可在各種語言或區域之間流通的程序，而**本土化**是針對特定語言或區域改寫軟體的程序。可使用依照特定文化需求修改執行時間程式行為的介面，來開發國際化的軟體。本土化牽涉到建立線上資訊以支援語言或區域，稱之為**語言環境**。

不像其他的軟體，必須經過完全重寫才能以不同地區的語言和慣例來作業，國際化的軟體並不需要重寫。國際化軟體不需經過變更就可從一種語言環境移入另一種語言環境。Solaris 為國際化的系統，可提供建立國際化軟體所需要的基礎架構和介面。

國際化的基本步驟

國際化應用程式的可執行影像可攜帶至各種語言和區域中。若要將軟體國際化：

- 使用本書所說明的介面，使用可藉由動態重新編譯而修改的環境來建立軟體。
- 將軟體分割為可執程式碼與使用者可以見到的所有訊息。將訊息字串保留在訊息目錄中。

訊息字串已針對語言或區域進行翻譯。一個**語言環境**包括訊息字串與指定排序的方法。

若要使用產品的已本土化版本，使用者必須設定部分環境變數。該產品才能夠顯示已轉譯為該語言環境之語言的訊息。日期、時間、貨幣以及其他資訊都會根據特定語言環境的轉換來進行格式化與顯示。可提供不同層級的訊息翻譯和線上說明內容，如以下圖表所述。

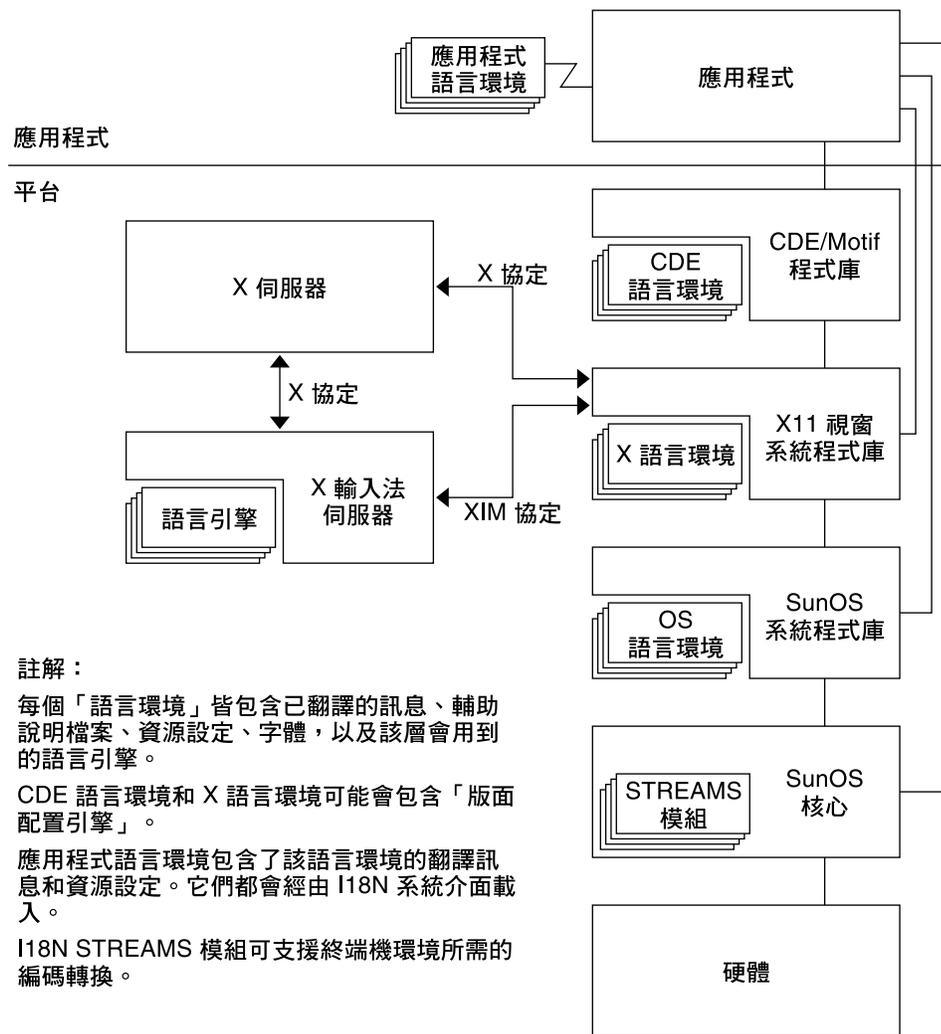


圖 1-1 Solaris 作業系統中語言環境的功能與結構

Solaris 介面中的本土化功能

OS (作業系統) 語言環境層提供了基本語言環境資料庫以及在應用程式執行時間插入至 OS 系統的功能。應用程式經由標準 API 存取這些作業系統語言環境模組。

該 X11 語言環境層提供 X 輸入方法與 X 輸出方法的介面，給 X11 應用程式用於本機文字輸入與顯示。字型可使應用程式顯示各種語言的字元。

CDE/Motif 內建於 X11 視窗系統的頂端。因此，CDE/Motif 可以透過 X11 API 來使用 X11 語言環境能力。爲了讓桌上管理系統能在目標語言環境中正常運作，Solaris 本土化具備了 CDE 應用程式的各種特定語言環境配置。提供不同層級的訊息翻譯與線上說明內容。

何謂語言環境？

應用程式的關鍵概念就是程式的**語言環境**。語言環境是原生語言環境的明確模組與定義。語言環境的概念是明確定義的，並包括在「ANSI C 語言」標準的程式庫定義中。

語言環境是由一些種類所組成，包括依賴於國家的格式或其他規格。程式的語言環境可定義自身的字碼集、資料和時間的格式慣例、幣制慣例、十進制的格式慣例以及分頁(排序)順序。

語言環境可由基礎語言、使用的國家(地區)以及可選字碼集來組成。通常會假設字碼集。例如，德語爲 `de`，爲德國的縮寫，而瑞士德語爲 `de_CH`，`CH` 爲瑞士邦聯的縮寫。此轉換可容許國家間的特定差異，如貨幣單位記號。

超過一個以上的語言環境可與一種特定語言相關聯，且可容許區域性的差異。例如，一位在美國說英語的使用者可以選取 `en_US` 語言環境(美式英語)，而另一位在英國說英語的使用者可以選取 `en_GB`(英式英語)。

語言環境名稱通常是由 `LANG` 環境變數指定的。語言環境種類隸屬於 `LANG`，但是可以個別設定，如此即會置換 `LANG`。如果已設定 `LC_ALL` 運算元，它會置換 `LANG` 和所有個別的語言環境種類。

語言環境命名慣例爲：

```
language[_territory][.codeset] [@modifier]
```

其中雙字母的 *language* 字碼是來自 ISO 639，雙字母的 *territory* 字碼是來自 ISO 3166，*codeset* 是的使用在語言環境中的字碼集名稱，*modifier* 是區分沒有修飾元之語言環境的特性名稱。

所有 Solaris 產品的語言環境都以 US-ASCII 字碼值保存「可攜字元集」的字元。

如需「可攜字元集」的詳細資訊，請參閱「X/Open CAE Specification: System Interface Definitions, Issue 5」(ISBN 1-85912-186-1)。

單一語言環境可以擁有一個以上的語言環境名稱。例如，`POSIX` 是與 `C` 相同的語言環境。

C 語言環境 - 預設語言環境

C 語言環境也稱為 POSIX 語言環境，是所有 POSIX 相容系統的 POSIX 系統預設語言環境。Solaris 作業系統即是 POSIX 系統。「單一 UNIX 規格」第 3 版中定義了 C 語言環境。請前往下列網址註冊，以便閱讀和下載此規格：<http://www.unix.org/version3/online.html>。

您可以透過下列兩種方法其中之一，指定您的國際化程式執行於 C 語言環境：

- 取消設定所有語言環境的環境變數。

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

取消設定所有語言環境的環境變數之後，便會以 C 語言環境來執行應用程式。

- 明確地設定語言環境為 C 或 POSIX。

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

部分應用程式會檢查 LANG 環境變數，而不實際呼叫 `setlocale(3C)` 來參閱目前的語言環境。在此情況下，`setenv` 會指定 LC_ALL and LANG 環境變數，以明確地設定 C 語言環境。如需瞭解語言環境的各個環境變數之間的優先順序關係，請參閱「`setlocale(3C)` 線上說明手冊」。

若要在終端機環境中檢查目前的語言環境設定，請執行 `locale(1)` 指令。

```
system% locale
```

完整與部份語言環境

完整的 Solaris 語言環境擁有所有列出的功能以及使用相關語言的本土化系統訊息。部份語言環境沒有安裝本土化的訊息。Solaris 環境中的所有語言環境都可顯示本土化訊息，可為已安裝的相關語言提供本土化訊息。例如，下列語言環境可以是部分或完整語言環境：

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

當您從 Languages CD 安裝德文訊息翻譯時，上述所有的語言環境都會變成**完整語言環境**，因為它們都可存取已完全翻譯的桌上管理系統。Languages CD 包含下列語言以及語言環境的訊息翻譯：

- 德文
- 法文
- 西班牙文

- 瑞典文
- 義大利文
- 日文
- 韓文
- 簡體中文語言環境
- 繁體中文語言環境

Software CD 提供了所有的部份語言環境。Language CD 提供了訊息翻譯。

所有英文語言環境也是完整的語言環境，並可自 Software CD 取得。

受語言環境影響的行為

不同的文化通常使用不同的數字格式慣例，來填寫日期和時間、劃分字詞和片語或引用文章和演講資料。語言環境會決定如何處理不同區域的下列作業、檔案、格式和表示式：

- 文字資料的編碼與處理
- 語言識別和資源檔案的編碼
- 文字字串的描繪與版面配置
- 用戶端之間的文字交換
- 符合字碼集的輸入法選擇以及選定字體的處理需求
- 特定文化的字型與圖示檔案
- 動作與檔案類型
- 使用者介面定義 (UID) 檔案
- 日期與時間格式
- 數值格式
- 貨幣格式
- 排序順序
- 語言環境的特定正規表示式處理
- 資訊以及診斷訊息的格式，與互動回應

Solaris 環境會將和語言及文化相關的資訊與應用程式分隔開來，並將資訊儲存在應用程式之外。此方法將不再需要針對每個市場來翻譯、重寫或重新編譯應用程式。進入新市場的唯一需求就是將外部資訊本土化為當地的語言和慣例。

語言環境的種類

語言環境的種類如下：

LC_CTYPE	控制字元處理功能的行為。
LC_TIME	指定日期和時間的格式，包括月份名稱、一星期的天數以及一般完整和縮寫的表示法。
LC_MONETARY	指定幣制格式，包括語言環境的貨幣符號、以千計分隔符號、記號位置、分數數字等。

LC_NUMERIC	指定十進制分割元 (或基數字元)、以千計分隔符號以及群組化。
LC_COLLATE	指定字元排序順序以及語言環境的常規表示式定義。
LC_MESSAGES	指定本土化訊息、語言環境的肯定與否定回應 (yes 和 no 字串和表示式) 所使用的語言。
LO_LTYPE	指定提供關於語言描繪資訊的版面配置引擎。語言描繪 (或文字描繪) 取決於字體的外形和方向屬性。

為本土化使用語言環境種類

在進行產品的本土化時，必須徵詢目標語言或地區的當地使用者。對於開發人員來說，某些資訊樣式和格式看來平淡無奇且經常使用。然而，對使用者來說，這些格式看來可能是棘手、錯誤，甚或是討厭的。下列章節說明在 Solaris 作業系統中，您可以根據產品本土化需求而自訂的元素。

時間格式

下表顯示數種不同語言環境填寫 11:59 P.M. 的方式。

表 1-2 國際化時間格式

語言環境	格式
加拿大	23:59
芬蘭文	23.59
德文	23.59 Uhr
挪威文	23.59
泰文	23:59
英文	23:59

時間將同時以 12 小時時鐘以及 24 小時時鐘來表示。小時與分鐘的分隔符號可以是冒號 (:) 或句號 (.)。

國家之間或國家之中有時區劃分。雖然可依據之前時數或之後時數、協調的格林威治標準時間，UTC (或「格林威治標準時間」，GMT) 來表示一個時區，但這個數字並不一定是整數。例如，紐芬蘭位於與鄰近時區相異半小時的時區中。

夏令時 (DST) 的起迄日期依各國情況而有所不同。許多國家根本不實施 DST。此外，同一時區中的夏令時也不盡相同。以美國為例，夏令時乃是依照州政府規定實施。

日期格式

下表顯示全球使用的數種日期格式。同一個國家中可能存在多種日期格式。

表 1-3 國際日期格式

語言環境	慣例	範例
加拿大 (英文)	dd/mm/yy	24/08/01
丹麥文	yyyy-mm-dd	2001-08-24
芬蘭文	dd.mm.yyyy	24.08.2001
法文	dd/mm/yyyy	24/08/2001
德文	yyyy-mm-dd	2001-08-24
義大利文	dd/mm/yy	24/08/01
挪威文	dd-mm-yy	24-08-01
西班牙文	dd-mm-yy	24-08-01
瑞典文	yyyy-mm-dd	2001-08-24
英國	dd/mm/yy	24/08/01
美國	mm-dd-yy	08-24-01
泰文	dd/mm/yyyy	24/08/2001

數字格式

英國與美國是全球少數兩個使用句號來指示小數點位置的國家。許多其他的國家使用逗號來替代。十進制分隔符號也稱做是**基數**數字元。同樣的，當英國和美國使用逗號來分隔千位組時，許多其他的國家則使用句號來代替，一些國家則是使用狹窄空間來分隔千位組。

當傳輸包含特定語言環境格式的資料檔案至另一個使用不同語言環境的系統時，常常發生解譯錯誤的情形。例如，包含法文格式數字的檔案就不能用在特定的英文程式。

下表顯示一些常用的數值格式。

表 1-4 國際化數值慣例

語言環境	龐大數字
加拿大 (英文)	4,294,967.00
丹麥文	4.294 967.295,00
芬蘭文	4 294 967 295,00
法文	4 294 967 295,00
德文	4,294,967.00
義大利文	4.294.967,00
挪威文	4.294.967.295,00
西班牙文	4.294.967.295,00
瑞典文	4 294 967 295,00
英國	4,294,967,295.00
美國	4,294,967,295.00
泰文	4,294,967,295.00

注意 – 並沒有哪一種特別的語言環境慣例會指定如何分隔清單中的數值。

國際性貨幣格式

全球的貨幣單位和表示方式具有極大差異。貨幣的本土與國際符號可能有所不同。下列表格顯示某些國家的貨幣格式。

表 1-5 國際幣制慣例

語言環境	貨幣型	範例
加拿大 (英文)	美元 (\$)	\$1,234.56
加拿大 (法文)	美元 (\$)	1 234,56\$
丹麥文	克羅納 (kr)	Kr 1.234,56
芬蘭文	歐元 (€)	€ 1 234,56
法文	歐元 (€)	€ 1,234
日文	日圓 (¥)	¥ 1,234

表 1-5 國際幣制慣例 (續)

語言環境	貨幣型	範例
挪威文	克羅納 (kr)	kr 1.234,56
瑞典文	克朗 (Kr)	1 234,56 Kr
英國	英鎊 (£)	£1,234.56
美國	美元 (\$)	\$1,234.56
泰文	Baht	2539 Baht
歐元	歐元 (€)	€ 5,000

目前發行版本支援歐元貨幣。本土貨幣符號仍然可以使用以提供回溯相容性。

表 1-6 支援歐元貨幣的使用者語言環境

區域	語言環境名稱	ISO 字碼集
奧地利	de_AT.ISO8859-15	8859-15
比利時 (法文)	fr_BE.ISO8859-15	8859-15
比利時 (法蘭德斯語)	nl_BE.ISO8859-15	8859-15
丹麥	da_DK.ISO8859-15	8859-15
愛沙尼亞	et_EE.ISO8859-15	8859-15
芬蘭	fi_FI.ISO8859-15	8859-15
法國	fr_FR.ISO8859-15	8859-15
德國	de_DE.ISO8859-15	8859-15
英國	en_GB.ISO8859-15	8859-15
愛爾蘭	en_IE.ISO8859-15	8859-15
義大利	it_IT.ISO8859-15	8859-15
荷蘭	nl_NL.ISO8859-15	8859-15
葡萄牙	pt_PT.ISO8859-15	8859-15
嘉泰羅尼亞西班牙文	ca_ES.ISO8859-15	8859-15
西班牙	es_ES.ISO8859-15	8859-15
瑞典	sv_SE.ISO8859-15	8859-15
美國	en_US.ISO8859-15	8859-15

歐元的語言環境是以 ISO8859-15 字碼集為基礎。

請注意，轉換後的貨幣數目所需要的空格數可能與原始空格數不同，例如，\$1,000 可能變為 € 1.307.000。

歐元地區語言環境的目前環境設定狀態是 locale 公用程式的 LC_MONETARY 運算元來表示。例如，下表顯示了德國的狀態。

表 1-7 德文語言環境以及相對應的 LC_MONETARY 運算元

語言環境	LC_MONETARY
de_DE.ISO8859-1	DM
de_DE.ISO8859-15	歐元
de_DE.UTF-8	歐元
de_DE.ISO8859-15@euro	歐元
de_DE.UTF-8@euro	歐元

語言文字與字母的差異

本節說明語言之間重要的差異。

文字分割元

以英文為例，文字通常是以空格字元來分隔的。不過，在中文、日文和泰文等語言中，文字之間通常沒有分隔。

排序

所有語言中的特殊字元排序是不相同的。例如，在德文中，“?” 字元是以一般的“o” 排序，但在瑞典則是分別排序，其中該字元為字母中的最後一個字。在某些語言中，字元有其重要性以決定字元序列的優先順序。例如，泰文字典中會根據不同的重要性來對字元的順序進行排序。

字元集

字元集的字母字元數和特殊字元數可以是不同的。當英文字母表只包含 26 個字元時，某些語言包含多出更多的字元。以日文為例，可能包含超過 20,000 個字元，而中文甚至可能包含更多的字元。

西歐字母表

大部分西歐國家的字母類似於英語系國家所使用的標準 26 個字元字母。這些字母通常會包含某些額外基本字元、某些標記或重音字元以及連音字。

日文文字

日文文字是由三種不同的字體混合組成的：

- 漢字，這是由中文衍生而來的表意文字
- 平假名和片假名，兩種語音字體 (或音節文字)

雖然每個平假名中的字元都有一個對等的片假名字元，但平假名仍是最常見的字體，它是一種草寫字母而不是區塊狀的字母格式。漢字字元用來撰寫字根。片假名通常用來表示「外來」文字，也就是從日文以外的語言所引進的文字。

漢字擁有數萬個字元，但是經常使用的數目正逐年穩定的減少當中。雖然一般的日本人平均擁有 2000 個漢字的能力，但現在仍經常使用的漢字大約只剩 3500 個。不過，為了符合日本業界標準 (JIS) 的需求，電腦系統必須支援超過 7000 個字元。另外，還有大約 170 個平假名和片假名字元。平均而言，55% 的日文文字為平假名，35% 是漢字以及 10% 為片假名。在日語文字中也會出現阿拉伯數字與羅馬字母。

雖然可以完全避免使用漢字，但大部分的日本人發現，文件若缺少漢字將變得難以理解。

韓文文字

可以使用語音書寫系統來寫出的韓文文字稱為 Hangul。Hangul 擁有超過 11,000 個字元，其是由稱為 jamos 的子音與母音所組成。全部的 Hangul 字彙中大約有 3000 字元為韓文電腦系統中所經常使用。韓國人也使用源自於中國文字的表意文字，稱為 Hanja。韓語文字需要超過 6000 個 Hanja 字元。當使用 Hangul 會造成語意不清時，大部份都是用 Hanja 來避免混淆。Hangul 字元是藉由合併子音與母音所形成。當這些字元合併之後，它們可以組合成一個音節，然後成為 Hangul 字元。Hangul 字元通常會呈現方塊狀，使其所佔的空間和 Hanja 字元相同。在韓語文字中也會出現阿拉伯數字、羅馬字母以及特殊符號字元。

泰文文字

泰文字元是以四個顯示格在顯示螢幕上定義為一個欄位位置。每個欄位位置最多可以有三個字元。顯示格的構成是以泰文字元分類為基礎。某些泰文字元可能是由另一個字元分類所組成。若是兩種字元可以組合在一起，則兩種字元會位於同一個顯示格中。否則，其位於分隔的顯示格中。

中文文字

中文幾乎全由稱為漢字的表意字體中的字元所組成。

- 在中華人民共和國 (PRC)，大約有 7000 個常用漢字字元使用於 GB2312 (zh 語言環境)，超過 20,000 個字元使用於 GBK 字元集 (zh.GBK 語言環境) 以及大約 30,000 字元使用於 GB18030-2000 字元集 (zh_CN.GB18030 語言環境)，包括 Unicode 3.0 中定義的所有 CJK 擴充 A 字元。
- 在台灣，最常使用的字元集為 CNS11643-1992 (zh_TW 語言環境) 和 Big5 (zh_TW.BIG5 語言環境)。它們共用大約 13,000 個漢字字元。
- 在香港，Big5-HKSCS 字元集 (zh_HK.BIG5HK) 是在 Big5 字元集中加入 4702 個字元而成的。

如果字元並不是根字元，則它通常是由兩個或更多部分所組成，兩個為最常見。在兩部分的字元中，其中一個部分是用來表意，而另一個則用來表音。偶爾兩部份同時用來表意。部首為最重要的元素，而字元通常也是以部首來編排，部首的數量約為數百個。單一音調可以藉由許多不同的字元來表示，這些字元在用法上無法互相交換。單一字元可以擁有不同的音調。

部分字元比其他字元更相稱於給定的文意。此相稱字元使用音調來作語音上的區別。相比之下，日文與韓文口語中並沒有音調。

有數種語音系統可用來表示中文。在中華人民共和國最普遍的是**拼音**，其使用羅馬字元，並廣泛的在西方針對地名使用，例如 Beijing (北京)。威妥系統是較古老的語音系統，從前使用在地名上，像是 Peking。在台灣則是使用**注音** (或 ㄅ ㄆ ㄇ)，這種具有獨特文字形式的語音字母來取代拼音。

希伯來文文字

希伯來文文字是用來書寫希伯來文與意第緒文語言的字體。希伯來文使用雙向字體。希伯來文字母是從右向左書寫和閱讀，而數字是從左向右閱讀的。任何內嵌於希伯來語文字中的英語文字也是從左向右閱讀。

希伯來文使用 27 字元的字母，並採用標準拉丁文 (或英文) 字母中的發音標記和數字。希伯來文文字也包括母音與發音符號。這些標記是以基本字元中的點 (dagesh)、字元下的母音標記或字元左上角的重音來表示。這些符號一般只使用在禮拜儀式文字中，在日常使用中很少見。希伯來文沒有大寫字母。

北印度文文字

北印度文文字是以梵文字母 (稱為 Devanagari) 書寫的，其意為「上帝的寫作」。北印度文是語音語言，並且撰寫為一連串的音節。每一個音節是由三種類型的字母片斷 (梵文字母字元) 所組成：子音字母、獨立的母音以及具依賴性的母音符號。音節本身是由一個子音和母音核所組成，並附帶一個選擇性的前置子音。不像英文是從基準線開始，Devanagari 字元是從寫在字元最上方的水平線 (稱為前筆畫) 垂下。這些字元可根據上下文來結合或變更外形。如同希伯來文，北印度文文字在大寫與小寫字母之間沒有區別。

鍵盤差異

美式鍵盤上的所有字元不會都出現在其他鍵盤上。同樣地，其他鍵盤通常包含許多在美式鍵盤上不會看到的字元。

您可以使用任何鍵盤從任何語言環境輸入字元，因為輸入是由 Solaris 作業系統所處理的。

注意 – 在 SPARC® 機器上，Compose 鍵可用來在任何支援的 ISO8859 字元集中產生帶有區別符號的拉丁字元。Compose 鍵可以用在以 Latin 為基礎的語言環境，但是不能用在韓文、中文或日文語言環境，而 UTF-8 語言環境例外。

紙張大小的差異

在每個國家中，普遍使用小號的紙張大小。通常只有一種紙張大小會比其他的大小較為普遍。大部份的國家都遵循 ISO 標準 216：「寫入紙張以及某些類別的列印事務—整齊的大小—A 與 B 系列。」

國際化應用程式不應假設可用的頁面大小。Solaris 系統並不提供追蹤輸出頁面大小的支援。追蹤輸出頁面大小是應用程式的責任。下表顯示一般國際頁面大小。

表 1-8 一般國際化頁面大小

紙張類型	大小	國家
ISO A4	21.0 公分乘 29.7 公分	除了美國以外的所有地方
ISO A5	14.8 公分乘 21.0 公分	除了美國以外的所有地方
JIS B4	25.9 公分乘 36.65 公分	日本
JIS B5	18.36 公分乘 25.9 公分	日本
U.S. Letter	8.5 英吋乘 11 英吋	美國和加拿大
U.S. Legal	8.5 英吋乘 14 英吋	美國和加拿大

第 2 章

一般國際化功能

本章將討論 Solaris 作業系統中所包含的一些國際化功能。本章包含以下主題。

- 第 35 頁的「字碼集獨立性的支援」
- 第 37 頁的「語言環境資料庫」
- 第 38 頁的「程序碼格式」
- 第 38 頁的「多位元組支援環境」
- 第 38 頁的「動態連結應用程式」
- 第 39 頁的「已變更介面」
- 第 40 頁的「ctype 巨集」
- 第 40 頁的「libc 的國際化 API」
- 第 47 頁的「genmsg 公用程式」
- 第 48 頁的「使用者自行定義和使用者可擴充字碼的轉換」
- 第 49 頁的「國際化網域名稱 (IDN) 支援」

字碼集獨立性的支援

EUC 是擴充式 UNIX 編碼 (Extended UNIX® Code) 的縮寫。Solaris 作業系統支援非 EUC 的編碼方式，例如日本的 PC-Kanji (慣用名稱是 Shift_JIS)、台灣的 Big5 以及中華人民共和國的 GBK。由於大部分的電腦銷售需要非 EUC 字碼集支援，目前 Solaris 環境提供了一個可同時啓用 EUC 和非 EUC 字碼集支援的穩固架構。此支援稱為**字碼集獨立性 (CSI)**。

CSI 的目的是從 Solaris 作業系統程式庫與指令中，移除對特定字碼集或編碼方式的依賴性。CSI 架構可以使 Solaris 作業系統支援任何 UNIX 檔案系統安全編碼。CSI 支援一些新的字碼集，例如 UTF-8、PC-Kanji 以及 Big5。

CSI 方法

透過字碼集獨立性，應用程式與平台軟體開發人員能夠將其程式碼保持獨立於任何編碼方式，例如 UTF-8。CSI 也提供在不需修改原始程式碼的情況下採用任何新編碼方式的能力。此架構方法不同於 Java™ 國際化，因為應用程式不必受限於使用 UTF-16。

許多現有國際化的應用程式 (例如 Motif) 自動從基礎系統繼承 CSI 支援。這些應用程式在不需修改的情況下即可在新的語言環境中使用。

CSI 自始即獨立於任何字碼集。然而，下列關於檔案程式碼編碼方式 (字碼集) 的假設，仍然可適用於目前 Solaris 系統：

- 檔案程式碼為 ASCII 的超集合。
- NULL 位元組值 (0x00) 的出現不是為了支援以 null 為結尾的多位元組字元串，而做為多位元組字元位元組的一部分。
- ASCII 斜線字元位元組值 (0x2f) 的出現不是為了支援 UNIX 路徑名稱，而做為多位元組字元位元組的一部分。

已啓用 CSI 的指令

本節內容將列出目前 Solaris 環境中已啓用 CSI 的指令。每個指令的線上說明手冊中所包含的屬性區段，會指出該指令是否為已啓用 CSI 的指令。

所有的指令都位在 /usr/bin 目錄中，除非另有說明。

/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

已啓用 CSI 的程式庫

所有在 libc (/usr/lib/libc.so) 中的函式，幾乎都是已啓用 CSI 的函式。然而，下列在 libc 中的函式並非已啓用 CSI 的函式，而是依賴於 EUC 的函式：

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

在目前 Solaris 環境中，libgen /usr/ccs/lib/libgen.a 以及 libcurses /usr/ccs/lib/libcurses.a 皆已國際化，但沒有啓用 CSI。

語言環境資料庫

語言環境資料庫的格式與結構是私有的，並將在未來發行版本中有所變動。當您在開發國際化應用程式時，可以使用 libc 中的國際化 API。這些 API 的說明位於第 40 頁的「libc 的國際化 API」，而非連結到語言環境資料庫。

注意 – 當您在 Solaris 環境中作業時，請使用已包含於目前 Solaris 發行版本中的語言環境資料庫。請不要使用上一個 Solaris 版本的語言環境。

程序碼格式

程序碼格式，也就是 Solaris 作業系統中的寬字元程式碼格式，是私有的並將在未來發行版本中有所變動。因此，在開發國際化應用程式時，請不要假設程序碼格式是相同的。相反地，請使用 libc 中的國際化 API，這個部分將於第 40 頁的「libc 的國際化 API」說明。

注意 – 所有 Unicode 語言環境的程序碼皆使用 UTF 32 表示法。如需 UTF 32 的詳細資訊，請參閱「Unicode 標準附錄 #19：UTF 32」以及「Unicode 標準附錄 #27：Unicode 3.1」(摘自「Unicode 學術學會」或 <http://www.unicode.org/>)。

多位元組支援環境

多位元組字元是無法使用單一位元組來儲存的字元，例如中文、日文或韓文字元。這些字元需要 2、3 或 4 個位元組的儲存空間。ISO/IEC 9899：1990 子條款 3.13 中記載了更精確的定義。

ANSI C 的一號修正案 (即 ISO/IEC 9899：1990) 中，增加了新的國際化功能，並統稱為多位元組支援環境 (MSE)。一號修正案針對多位元組字碼集與較佳的寬字元處理支援，定義了附加的國際化 API 並隨附說明。

程式設計模型使得這些多位元組字元可以邏輯單位讀取，並如寬字元般儲存在內部。程式可以將這些寬字元視為邏輯實體來處理。最後，可以經由適當轉譯，將這些寬字元以邏輯單位寫出。

這個程序類似讀取、處理單一位元組字元，並再次寫出的方法。MSE 使程式可以使用與單一位元組字元所用相同之程式設計模型，來處理多位元組字元。

動態連結應用程式

您可以使用動態連結或靜態連結，將應用程式連結至系統程式庫，例如 libc。任何需要系統程式庫中國際化功能的應用程式，都必須是動態連結。如果應用程式已經靜態連結，那麼想要使用 setlocale 功能將語言環境設定為任何其他不是 C 和 POSIX 的作業將會失敗。靜態連結的應用程式只能在 C 和 POSIX 語言環境中作業。

依預設值，連結程式會嘗試動態連結至應用程式。如果連結程式與編譯程式的指令行選項包含 -Bstatic 或 -dn 規格，那麼您的應用程式可能已靜態連結。您可以檢查現有的應用程式是否為使用 /usr/bin/ldd 指令完成動態連結。

例如，下列指令的回應表示 `/sbin/sh` 指令並不是一個動態連結的程式：

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

下列指令的回應表示 `/usr/bin/ls` 指令已動態連結至兩個程式庫，`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
```

已變更介面

`libw` 和 `libintl` 已經移至 `libc`，並且不再放置於 `libw` 和 `libintl`。

共用物件會確保現有應用程式的運行時間相容性，並伴隨歸檔的使用提供建立應用程式編譯環境的相容性。然而，您不再需要針對 `libw` 或 `libintl` 來建立應用程式。

下列清單將顯示 `libw` 中的 `stub` 進入點：

<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>wcsprbr</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>toupper</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>wssp</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>

下列清單將顯示 `libintl` 中的 `stub` 根進入點：

`bindtextdomain`

dcgettext
dgettext
gettext
textdomain

ctype 巨集

字元分類和轉換巨集定義在 `/usr/include/ctype.h` 之中。目前 Solaris 環境提供一組 `ctype` 巨集，此巨集支援由 XPG4 定義的字元分類和轉換語義。若要使所有 XPG4 和 XPG4.2 應用程式自動地存取新巨集，必須符合下列其中一項條件：

- 已定義 `_XPG4_CHAR_CLASS`。
- 已定義 `_XOPEN_SOURCE` 和 `_XOPEN_VERSION=4`。
- 已定義 `_XOPEN_SOURCE` 和 `_XOPEN_SOURCE_EXTENDED=1`。

由於 `_XOPEN_SOURCE`、`_XOPEN_VERSION` 和 `_XOPEN_SOURCE_EXTENDED` 引進除了新的 `ctype` 巨集以外的 XPG4 相關功能，因此非 XPG4 或 XPG4.2 應用程式就應該使用 `__XPG4_CHAR_CLASS__`。

同時存在相對應的 `ctype` 函式。目前 Solaris 環境函式也支援 XPG4 語義。

libc 的國際化 API

目前 Solaris 環境提供兩組 API：

- 多位元組 (檔案程式碼)
- 寬字元 (程序碼)

寬字元程式碼為固定寬度的邏輯實體單位。因此使用多位元組字元時，並不需要維護適當字元界限的記錄。

當程式讀取來自於檔案的輸入時，您可以使用像是 `fscanf` 和 `fwscanf` 的輸入函式或是在輸入之後使用像是 `mbtowc` 和 `mbsrtowcs` 的轉換函式，自動將檔案中的多位元組資料轉換為寬字元程序碼。若要將輸出資料從寬字元格式轉換為多位元組字元格式，請使用像是 `fwprintf` 和 `fprintf` 的輸出函式或在輸出之前套用像是 `wctomb` 和 `wcsrtombs` 的轉換函式。

本章其餘表格將說明包含在目前 Solaris 系統中的國際化 API。

下列表格將說明 `libc` 中的訊息傳送函式 API。

表 2-1 libc 中的訊息函式

程式庫常式	描述
<code>bindtextdomain()</code>	將路徑連結至訊息網域
<code>catclose()</code>	關閉訊息目錄
<code>catgets()</code>	讀取程式訊息
<code>catopen()</code>	開啓訊息目錄
<code>dcgettext()</code>	從訊息目錄中取得指定網域與種類的訊息
<code>dgettext()</code>	從訊息目錄中取得指定網域的訊息
<code>gettext()</code>	從訊息資料庫取回文字串
<code>textdomain()</code>	設定與查詢目前的網域

下列表格將說明 libc 中的字碼轉換函式 API。

表 2-2 libc 中的字碼轉換

程式庫常式	描述
<code>iconv()</code>	轉換字碼
<code>iconv_close()</code>	取消配置轉換描述元
<code>iconv_open()</code>	配置轉換描述元

下列表格將說明 libc 中的常規表示式 API。

表 2-3 libc 中的常規表示式

程式庫常式	描述
<code>fnmatch()</code>	符合的檔案或路徑名稱
<code>regcomp()</code>	編譯常規表示式
<code>regerror()</code>	提供錯誤代碼與錯誤訊息的對映
<code>regexec()</code>	執行常規表示式匹配
<code>regfree()</code>	釋放 <code>regcomp()</code> 所配置的記憶體

下列表格將說明 libc 中的寬字元函式 API。

表 2-4 libc 中的寬字元類別

程式庫常式	描述
wctrans()	定義字元對映
wctype()	定義字元類別

下列表格將列出 libc 中的修改和查詢語言環境。

表 2-5 libc 中的修改和查詢語言環境

程式庫常式	描述
setlocale()	修改以及查詢程式的語言環境

下列表格將列出 libc 中的查詢語言環境資料。

表 2-6 libc 中的查詢語言環境資料

程式庫常式	描述
localeconv()	取得目前語言環境的貨幣與數值格式資訊
nl_langinfo()	取得目前語言環境的語言與文化資訊

下列表格將說明 libc 中的字元分類函式 API。

表 2-7 libc 中的字元分類和轉譯

程式庫常式	描述
isalnum()	字元為字母或數字？
isalpha()	字元是否為字母？
isascii()	字元是否為 ASCII 字元？
iscntrl()	字元是否為控制字元？
isdigit()	字元是否為數字？
isenglish()	寬字元是否在補充字碼集的英文字母表中？
isgraph()	字元是否為可視字元？
isideogram()	寬字元是否為表意文字？
islower()	字元是否為小寫？
isnumber()	寬字元是否為補充字碼集中的數字？
isphonogram()	寬字元是否為音標符號？

表 2-7 libc 中的字元分類和轉譯 (續)

程式庫常式	描述
isprint()	字元是否可以列印？
ispunct()	字元是否為標點符號？
isspace()	字元是否為空白？
isspecial()	特殊寬字元是否源自補充字碼集？
isupper()	字元是否為大寫？
iswalnum()	寬字元是字母字元或數字？
iswalpha()	寬字元是否為字母？
iswascii()	寬字元是否為 ASCII 字元？
iswcntrl()	寬字元是否為控制字元？
iswdigit()	寬字元是否為數字？
iswgraph()	寬字元是否為可視字元？
iswlower()	寬字元是否為小寫？
iswprint()	寬字元是否為可列印的字元？
iswpunct()	寬字元是否為標點符號？
iswspace()	寬字元是否為空格？
iswupper()	寬字元是否為大寫？
iswxdigit()	寬字元是否為十六進位數字？
isxdigit()	字元是否為十六進位數字？
tolower()	將大寫字元轉換為小寫。
toupper()	將小寫字元轉換為大寫。
towctrans()	寬字元對映。
towlower()	將大寫寬字元轉換為小寫。
towupper()	將小寫寬字元轉換為大寫。

下列表格將說明 libc 中的字元比較函式 API。

表 2-8 libc 中的字元比較

程式庫常式	描述
strcoll()	排序字串

表 2-8 libc 中的字元比較 (續)

程式庫常式	描述
strxfrm()	變換字串以進行比較
wscoll()	排序寬字元字串
wcsxfrm()	變換寬字元字串以進行比較

下列表格將說明 libc 中的貨幣處理函式 API。

表 2-9 libc 中的貨幣格式

程式庫常式	描述
localeconv()	取得目前語言環境的貨幣格式資訊
strfmon()	將貨幣值轉換為字串表示式

下列表格將說明 libc 中的日期與時間格式。

表 2-10 libc 中的日期與時間格式

程式庫常式	描述
getdate()	轉換使用者格式日期與時間。
strftime()	將日期與時間轉換為字串表示式。%u 轉換函式遵循系統介面和標頭第 2 版第 4 次修訂中的 X/Open CAE 規格。此函式以十進位數字 [1,7] 來表示工作日，1 代表星期一。
strptime()	日期與時間轉換。

下列表格將說明 libc 中的多位元組處理函式 API。

表 2-11 libc 中的多位元組處理

程式庫常式	描述
btowc()	單一位元組至寬字元的轉換
mblen()	取得字元中的位元組數目
mbrlen()	取得字元中的位元組數目 (可重新開始計數)
mbrtowc()	將字元轉換為寬字元程式碼 (可重新開始計數)
mbsinit()	決定轉換物件的狀態
mbsrtowcs()	將字串轉換為寬字元字串 (可重新開始計數)
mbstowcs()	將字串轉換為寬字元字串

表 2-11 libc 中的多位元組處理 (續)

程式庫常式	描述
mbtowc ()	將字元轉換為寬字元程式碼。

下列表格將說明 libc 中的寬字元與字串處理。

表 2-12 libc 中的寬字元和字串處理

程式庫常式	描述
wcrtomb ()	將寬字元程式碼轉換為字元 (可重新開始計數)
wcscat ()	串接寬字元字串
wcschr ()	尋找寬字元字串中的字元
wcscmp ()	比較寬字元字串
wcscopy ()	複製寬字元字串
wcscspn ()	傳回未出現在另一個寬字元字串的字串長度
wcslen ()	取得寬字元字串的長度
wcsncat ()	將寬字元字串串接至長度 n
wcsncmp ()	比較寬字元字串至長度 n
wcsncpy ()	將寬字元字串複製為長度 n
wcspbrk ()	傳回出現在另一個寬字元字串的字元後的字串
wcsrchr ()	從右邊開始尋找寬字元字串中的字元
wcrtombs ()	將寬字元字串轉換為字串 (可重新開始計數)
wcsspn ()	傳回出現在另一個寬字元字串的字串長度
wcstod ()	將寬字元字串轉換為雙精度
wcstok ()	在寬字元字串之間移動記號
wcstol ()	將寬字元字串轉換為長整數
wcstombs ()	將寬字元字串轉換為多位元組字串
wcstoul ()	將寬字元字串轉換為無符號長整數
wscwcs ()	尋找寬字元字串中的字串
wcswidth ()	決定寬字元字串的欄數位置
wctob ()	寬字元至單一位元組的轉換
wctomb ()	將寬字元轉換為多位元組字元

表 2-12 libc 中的寬字元和字串處理 (續)

程式庫常式	描述
wcwidth()	決定寬字元的欄數位置
wscol()	傳回寬字元字串的顯示寬度
wsdup()	複製寬字元字串

下列表格將說明 libc 中的格式化寬字元輸入與輸出。

表 2-13 libc 中的格式化寬字元輸入與輸出

程式庫常式	描述
fwprintf()	列印格式化寬字元輸出
fwscanf()	轉換格式化寬字元輸入
swprintf()	列印格式化寬字元輸出
swscanf()	轉換格式化寬字元輸入
vfwprintf()	stdarg 引數清單的寬字元格式化輸出
vswprintf()	stdarg 引數清單的寬字元格式化輸出
wprintf()	列印格式化寬字元輸出
wscanf()	轉換格式化寬字元輸入
wsprintf()	根據格式產生寬字元字串
wsscanf()	格式化輸入轉換

這個表格將說明 libc 中的寬字串函式 API。

表 2-14 libc 中的寬字串

程式庫常式	描述
wcsstr()	尋找寬字元子字串
wmemchr()	在記憶體中尋找寬字元
wmemcmp()	比較記憶體中的寬字元
wmemcpy()	複製記憶體中的寬字元
wmemmove()	使用重疊區域複製記憶體中的寬字元
wmemset()	設定記憶體中的寬字元
wscasecmp()	比較寬字元字串，忽略大小寫的差異

表 2-14 libc 中的寬字串 (續)

程式庫常式	描述
wncasecmp()	程序碼字串作業

下列表格將說明 libc 中的寬字元輸入與輸出。

表 2-15 libc 中的寬字元輸入與輸出

程式庫常式	描述
fgetwc()	從串流中取得多位元組字元，並轉換為寬字元
fgetws()	從串流中取得多位元組字串，並轉換為寬字元
fputwc()	將寬字元轉換為多位元組字元，並放入串流中
fputws()	將寬字元轉換為多位元組字串，並放入串流中
fwide()	設定串流方向
getwchar()	從 stdin 中取得多位元組字元，並轉換為寬字元
getws()	從 stdin 中取得多位元組字串，並轉換為寬字元
putwchar()	將寬字元轉換為多位元組字元，並放入 stdin 串流中
putws()	將寬字元轉換為多位元組字串，並放入 stdin 串流中
ungetwc()	將寬字元後推至輸入串流中

genmsg 公用程式

新的 genmsg 公用程式可以與 catgets() 函式家族配合使用，以建立國際化來源訊息目錄。這個公用程式會檢查 catgets 中函式呼叫的來源程式檔案，並從尋獲的資訊建立來源訊息目錄。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 */
```

在上述的範例中，`genmsg` 會在來源檔案上執行 `example.c`，並產生了名為 `example.c.msg` 的來源訊息目錄。`-c` 選項 (包含引數 `NOTE`) 會使得 `genmsg` 在目錄中包含註釋。若是來源程式中的註釋包含指定的字串，則下一個從 `catgets` 呼叫中所擷取的字串之後的訊息目錄便會出現註釋。

您可以使用 `genmsg`，自動為訊息集中的訊息計數。

如需詳細資訊，請參閱「`genmsg(1)` 線上說明手冊」。

若要產生格式化的訊息目錄檔，請使用 `gencat(1)` 公用程式。

關於可攜式訊息檔案 (`.po` 檔案) 的訊息擷取公用程式以及如何從 `.po` 檔案產生訊息物件檔案 (`.mo` 檔案) 的資訊。

使用者自行定義和使用者可擴充字碼的轉換

您可以使用 `geniconvtbl` 公用程式，建立使用者自行定義的字碼集轉換器。

這個公用程式使用像是 `iconv(1)` 和 `iconv(3C)` 的標準系統公用程式和介面，來啓用使用者自行定義和使用者可自訂的字碼集轉換。這個功能加強了應用程式處理不相容資料類型 (特別是從專用及傳統應用程式所產生的資料) 的功能。也支援對現存 Solaris 字碼集轉換的修改。

此公用程式的範例輸入來源檔位於 `/usr/lib/iconv/geniconvtbl/srcs/` 目錄中。

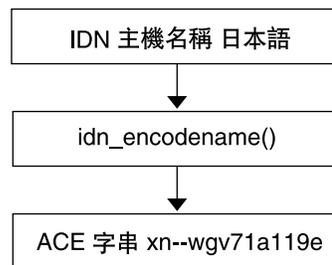
一旦使用者自行定義字碼轉換已準備好並正確放置之後，使用者可以使用 32 位元和 64 位元 Solaris 作業系統上 `iconv(1)` 公用程式和 `iconv(3C)` 函式的字碼轉換。

國際化網域名稱 (IDN) 支援

國際化網域名稱 (IDN) 使得主機和網域名稱可以使用非英文原生語言名稱。若要使用非英文的主機和網域名稱，請在將這些名稱傳送到解析程式常式 (RFC 3490 中指定的) 之前，先轉換為 ASCII 相容編碼 (ACE) 的編碼名稱。對於不支援 IDN 的系統管理應用程式，系統管理員也需要在檔案系統和應用程式中使用 ACE 名稱。

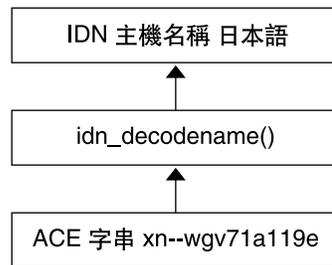
請參閱 RFC 3490 應用程式中的國際化網域名稱 (IDNA)。

`libidnkit(3EXT)` 中用於國際化網域名稱的 API，在 UTF-8 或應用程式語言環境的字碼集與 ACE 之間，提供方便的轉換。如果使用了 `idn_decodename2(3EXT)`，您也可以指定任意字碼集名稱做為輸入引數的字碼集。



使用 ACE 字串做為解析器常式 (如 `getaddrinfo(3SOCKET)`) 的輸入

■ 2-1 IDN 轉換為 ACE



ACE 字串從自解析器常式 (如 `getnameinfo(3SOCKET)`) 傳回

■ 2-2 ACE 轉換為 IDN

下列表格顯示您可以使用的雙向 `iconv` 字碼轉換。

表 2-16 iconv 字碼轉換

來源字碼	目的字碼
ACE	UTF-8
ACE-ALLOW-UNASSIGNED	UTF-8
UTF-8	ACE
UTF-8	ACE-ALLOW-UNASSIGNED

ACE 和 ACE-ALLOW-UNASSIGNED iconv 字碼轉換名稱具有下列含義：

- ACE。

在 iconv 字碼轉換中，ACE 是一個用來參閱到 ASCII 相容編碼 (定義於 RFC 3490) 的 fromcode 或 tocode 名稱。此轉換使用 STD3 ASCII 規則。不允許使用未指定的字元。ACE 通常用於儲存或是給定機器的主機或網域名稱。
- ACE-ALLOW-UNASSIGNED。

ACE-ALLOW-UNASSIGNED 執行與 ACE 相同的作業，並且 ACE-ALLOW-UNASSIGNED 允許使用未指定的字元。ACE-ALLOW-UNASSIGNED 通常用於查詢目的。

下列是從 ACE 轉換為 UTF-8 的範例，輸入來源是 hostnames.txt 檔案。輸出為標準輸出。

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

專屬的 IDN 轉換公用程式 idnconv(1) 提供 IDN 轉換及許多選項。這些選項可以控制轉換細節。

如需關於 IDN、轉換常式和 iconv 字碼轉換的資訊，請參閱「libidnkit(3LIB) 線上說明手冊」、「idn_decodename(3EXT) 線上說明手冊」、「idn_decodename2(3EXT) 線上說明手冊」、「idn_encodename(3EXT) 線上說明手冊」以及「iconv_en_US.UTF-8(5) 線上說明手冊」。

第 3 章

Solaris 環境中的本土化

本章就目前 Solaris 環境中的本土化功能加以討論。本章包含以下主題。

- 第 51 頁的「本土化的軟體支援」
- 第 52 頁的「受支援的語言環境」
- 第 58 頁的「語言環境的多重按鍵組合序列」
- 第 59 頁的「Solaris 環境中的鍵盤支援」
- 第 71 頁的「新的 Solaris 鍵盤軟體支援」

本土化的軟體支援

本節包含 Solaris 語言環境套裝軟體、CD-ROM 光碟片、本土化功能以及啓用字體的相關資訊。

Solaris 語言環境套裝軟體摘要

目前 Solaris 的所有語言環境套裝軟體已分類為完整語言環境或部分語言環境。

部分語言環境是語言環境的啓用程式。若系統上已安裝了部分語言環境，當 Solaris 作業系統上的 OS/GUI 訊息為英文時，使用者就可以在目標語言環境上輸入、顯示、列印文字及執行應用程式。Solaris Software CD 可提供所有的部分語言環境套裝軟體。日文和亞洲部分語言環境是依據語言來封裝的。部分語言環境是依據地理區域來封裝的。

完整語言環境套裝軟體包含軟體訊息的譯文、線上說明檔案、可選字型以及特定語言功能。完整語言環境套裝軟體提供了許多語言的全部語言功能集。以下列語言為基礎的語言環境為完整語言環境：

- 德文
- 法文

- 西班牙文
- 瑞典文
- 義大利文
- 日文
- 韓文
- 簡體中文
- 繁體中文

完整語言環境軟體係根據語言來進行套裝，並可在 Language CD 上取得。

注意 – 部分語言環境套裝軟體 (語言環境啓用程式) 必須按照順序進行安裝，使完整語言環境能產生作用。

在 Solaris 的安裝程序期間，系統會提示您選擇需要您支援的地理區域。安裝完成後的語言環境，會根據此階段的選擇來取得支援。部分語言環境是從 Solaris 作業系統的 Solaris Software CD-ROM 上進行安裝，而完整語言環境則是從 Language CD 上進行安裝。如果您不需要完整語言環境支援，您可以在安裝程序期間略過 Language CD 上的安裝。請注意，系統所預設安裝的語言環境為英文。

受支援的語言環境

下表列出了在 Solaris 環境中受支援的所有語言環境。語言環境名稱遵循國際命名標準。

表 3-1 亞洲語言環境

語言環境	使用者介面	區域	字碼集	語言支援
hi_IN.UTF-8	英文	印度	UTF-8	北印度文 (UTF-8) Unicode 4.0
ja	日文	日本	eucJP ¹	日文 (EUC) JIS X 0201-1976 JIS X 0208-1990 JIS X 0212-1990
ja_JP.eucJP	日文	日本	eucJP	日文 (EUC) JIS X 0201-1976 JIS X 0208-1990

¹ eucJP 是指日文 EUC 字碼集。ja_JP.eucJP 語言環境規格符合 UI_OSF 日文環境執行合約 1.1 版，而 ja 語言環境符合過去 Solaris 發行版本中的傳統規格。

表 3-1 亞洲語言環境 (續)

語言環境	使用者介面	區域	字碼集	語言支援
ja_JP.PCK	日文	日本	PCK ²	JIS X 0212-1990 日文 (PC 漢字)
ja_JP.UTF-8	日文	日本	UTF-8	JIS X 0201-1976 JIS X 0208-1990 日文 (UTF-8) Unicode 4.0
ko_KR.EUC	韓文	韓國	1001	韓文 (EUC) KS X 1001
ko_KR.UTF-8	韓文	韓國	UTF-8	韓文 (UTF-8) Unicode 4.0
th_TH.UTF-8	英文	泰國	UTF-8	泰文 (UTF-8) Unicode 4.0
th_TH.TIS620	英文	泰國	TIS620.2533	泰文 TIS620.2533
zh_CN.EUC	簡體中文	中國	gb2312 ³	簡體中文 (EUC) GB2312-1980
zh_CN.GBK	簡體中文	中國	GBK ⁴	簡體中文 (GBK)
zh_CN.GB18030	簡體中文	中國	GB18030-2000	簡體中文 (GB18030-2000) GB18030-2000
zh_CN.UTF-8	簡體中文	中國	UTF-8	簡體中文 (UTF-8) Unicode 4.0
zh_HK.BIG5HK	繁體中文	香港	Big5+HKSCS	繁體中文 (BIG5+HKSCS)
zh_HK.UTF-8	繁體中文	香港	UTF-8	繁體中文 (UTF-8) Unicode 4.0
zh_TW.EUC	繁體中文	台灣地區	cns11643	繁體中文 (EUC) CNS 11643-1992
zh_TW.BIG5	繁體中文	台灣地區	BIG5	繁體中文 (BIG5)
zh_TW.UTF-8	繁體中文	台灣地區	UTF-8	繁體中文 (UTF-8) Unicode 4.0

² PCK 也就是 Shift_JIS (SJIS)。

³ gb2312 表示簡體中文 EUC 的字碼集，其包含了 GB 1988-80 和 GB 2312-80。

⁴ GBK 是 GB 的延伸部分。這些延伸部分包含所有 GB 2312-80 字元和所有 ISO/IEC 10646-1 的統一漢字字集以及日文平假名和片假名字元。GBK 也同時包含許多 ISO/IEC 10646-1 的中文、日文與韓文的字元。

表 3-2 澳大拉西亞語言環境

語言環境	使用者介面	區域	字碼集	語言支援
en_AU.ISO8859-1	英文	澳大利亞	ISO8859-1	英文 (澳大利亞)
en_NZ.ISO8859-1	英文	紐西蘭	ISO8859-1	英文 (紐西蘭)

表 3-3 中美洲語言環境

語言環境	使用者介面	區域	字碼集	語言支援
es_CR.ISO8859-1	西班牙文	哥斯大黎加	ISO8859-1	西班牙文 (哥斯大黎加)
es_GT.ISO8859-1	西班牙文	瓜地馬拉	ISO8859-1	西班牙文 (瓜地馬拉)
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	巴拿馬	ISO8859-1	西班牙文 (巴拿馬)
es_SV.ISO8859-1	西班牙文	薩爾瓦多	ISO8859-1	西班牙文 (薩爾瓦多)

表 3-4 中歐語言環境

語言環境	使用者介面	區域	字碼集	語言支援
cs_CZ.ISO8859-2	英文	捷克共和國	ISO8859-2	捷克文 (捷克共和國)
de_AT.ISO8859-1	德文	奧地利	ISO8859-1	德文 (奧地利)
de_AT.ISO8859-15	德文	奧地利	ISO8859-15	德文 (奧地利, ISO8859-15 - Euro)
de_CH.ISO8859-1	德文	瑞士	ISO8859-1	德文 (瑞士)
de_DE.UTF-8	德文	德國	UTF-8	德文 (德國, Unicode 4.0)
de_DE.ISO8859-1	德文	德國	ISO8859-1	德文 (德國)
de_DE.ISO8859-15	德文	德國	ISO8859-15	德文 (德國, ISO8859-15 - Euro)
fr_CH.ISO8859-1	法文	瑞士	ISO8859-1	法文 (瑞士)
hu_HU.ISO8859-2	英文	匈牙利	ISO8859-2	匈牙利文 (匈牙利)
pl_PL.ISO8859-2	英文	波蘭	ISO8859-2	波蘭文 (波蘭)
pl_PL.UTF-8	英文	波蘭	UTF-8	波蘭文 (波蘭, Unicode 4.0)
sk_SK.ISO8859-2	英文	斯洛伐克	ISO8859-2	斯洛伐克文 (斯洛伐克)

表 3-5 東歐語言環境

語言環境	使用者介面	區域	字碼集	語言支援
bg_BG.ISO8859-5	英文	保加利亞	ISO8859-5	保加利亞文 (保加利亞)
et_EE.ISO8859-15	英文	愛沙尼亞	ISO8859-15	愛沙尼亞文 (愛沙尼亞)
hr_HR.ISO8859-2	英文	克羅埃西亞	ISO8859-2	克羅埃西亞文 (克羅埃西亞)

表 3-5 東歐語言環境 (續)

語言環境	使用者介面	區域	字碼集	語言支援
lt_LT.ISO8859-13	英文	立陶宛	ISO8859-13	立陶宛文 (立陶宛)
lv_LV.ISO8859-13	英文	拉脫維亞	ISO8859-13	拉脫維亞文 (拉脫維亞)
mk_MK.ISO8859-5	英文	馬其頓	ISO8859-5	馬其頓文 (馬其頓)
ro_RO.ISO8859-2	英文	羅馬尼亞	ISO8859-2	羅馬尼亞文 (羅馬尼亞)
ru_RU.KOI8-R	英文	俄羅斯	KOI8-R	俄文 (俄羅斯, KOI8-R)
ru_RU.ANSI1251	英文	俄羅斯	ansi-1251	俄文 (俄羅斯, ANSI 1251)
ru_RU.ISO8859-5	英文	俄羅斯	ISO8859-5	俄文 (俄羅斯)
ru_RU.UTF-8	英文	俄羅斯	UTF-8	俄文 (俄羅斯, Unicode 4.0)
sh_BA.ISO8859-2@bosnia	英文	波士尼亞	ISO8859-2	波士尼亞文 (波士尼亞)
sl_SI.ISO8859-2	英文	斯洛維尼亞	ISO8859-2	斯洛維尼亞文 (斯洛維尼亞)
sq_AL.ISO8859-2	英文	阿爾巴尼亞	ISO8859-2	阿爾巴尼亞文 (阿爾巴尼亞)
sr_YU.ISO8859-5	英文	塞爾維亞	ISO8859-5	塞爾維亞文 (塞爾維亞)
tr_TR.ISO8859-9	英文	土耳其	ISO8859-9	土耳其文 (土耳其)
tr_TR.UTF-8	英文	土耳其	UTF-8	土耳其文 (土耳其, Unicode 4.0)

表 3-6 中東語言環境

語言環境	使用者介面	區域	字碼集	語言支援
He	英文	以色列	ISO8859-8	希伯來文 (以色列)

表 3-7 北非語言環境

語言環境	使用者介面	區域	字碼集	語言支援
ar_EG.UTF-8	英文	埃及	UTF-8	阿拉伯文 (埃及)
Ar	英文	埃及	ISO8859-6	阿拉伯文 (埃及)

表 3-8 北美語言環境

語言環境	使用者介面	區域	字碼集	語言支援
en_CA.ISO8859-1	英文	加拿大	ISO8859-1	英文 (加拿大)
en_US.ISO8859-1	英文	美國	ISO8859-1	英文 (美國)
en_US.ISO8859-15	英文	美國	ISO8859-15	英文 (美國, ISO8859-15 - Euro)
en_US.UTF-8	英文	美國	UTF-8	英文 (美國, Unicode 4.0)
fr_CA.ISO8859-1	法文	加拿大	ISO8859-1	法文 (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 北歐語言環境

語言環境	使用者介面	區域	字碼集	語言支援
da_DK.ISO8859-1	英文	丹麥	ISO8859-1	丹麥文 (丹麥)
da_DK.ISO8859-15	英文	丹麥	ISO8859-15	丹麥文 (丹麥, ISO8859-15-Euro)
fi_FI.ISO8859-1	英文	芬蘭	ISO8859-1	芬蘭文, Unicode 4.0
fi_FI.ISO8859-15	英文	芬蘭	ISO8859-15	芬蘭文 (芬蘭, ISO8859-15-Euro)
fi_FI.UTF-8	英文	芬蘭	UTF-8	芬蘭文 (芬蘭)
is_IS.ISO8859-1	英文	冰島	ISO8859-1	冰島文 (冰島)
no_NO.ISO8859-1@bokmal	英文	挪威	ISO8859-1	挪威文 (Norway-Bokmal)
no_NO.ISO8859-1@nyorsk	英文	挪威	ISO8859-1	挪威文 (Norway-Nynorsk)
sv_SE.ISO8859-1	瑞典文	瑞典	ISO8859-1	瑞典文 (瑞典)
sv_SE.ISO8859-15	瑞典文	瑞典	ISO8859-15	瑞典文 (瑞典, ISO8859-15-Euro)
sv_SE.UTF-8	瑞典文	瑞典	UTF-8	瑞典文 (瑞典, Unicode 4.0)

表 3-10 南美語言環境

語言環境	使用者介面	區域	字碼集	語言支援
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	玻利維亞	ISO8859-1	西班牙文 (玻利維亞)
es_CL.ISO8859-1	西班牙文	智利	ISO8859-1	西班牙文 (智利)

表 3-10 南美語言環境 (續)

語言環境	使用者介面	區域	字碼集	語言支援
es_CO.ISO8859-1	西班牙文	哥倫比亞	ISO8859-1	西班牙文 (哥倫比亞)
es_EC.ISO8859-1	西班牙文	厄瓜多爾	ISO8859-1	西班牙文 (厄瓜多爾)
es_PE.ISO8859-1	西班牙文	秘魯	ISO8859-1	西班牙文 (秘魯)
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	烏拉圭	ISO8859-1	西班牙文 (烏拉圭)
es_VE.ISO8859-1	西班牙文	委內瑞拉	ISO8859-1	西班牙文 (委內瑞拉)
pt_BR.ISO8859-1	英文	巴西	ISO8859-1	葡萄牙文 (巴西)
pt_BR.UTF-8	英文	巴西	UTF-8	葡萄牙文 (巴西, Unicode 4.0)

表 3-11 南歐語言環境

語言環境	使用者介面	區域	字碼集	語言支援
ca_ES.ISO8859-1	英文	西班牙	ISO8859-1	嘉泰羅尼亞文 (西班牙)
ca_ES.ISO8859-15	英文	西班牙	ISO8859-15	嘉泰羅尼亞文 (西班牙, ISO8859-15 - Euro)
el_GR.ISO8859-7	英文	希臘	ISO8859-7	希臘文 (希臘)
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 - Euro)
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	義大利文	義大利	ISO8859-1	義大利文 (義大利)
it_IT.ISO8859-15	義大利文	義大利	ISO8859-15	義大利文 (義大利, ISO8859-15 - Euro)
it_IT.UTF-8	義大利文	義大利	UTF-8	義大利文 (義大利, Unicode 4.0)
pt_PT.ISO8859-1	英文	葡萄牙	ISO8859-1	葡萄牙文 (葡萄牙)
pt_PT.ISO8859-15	英文	葡萄牙	ISO8859-15	葡萄牙文 (葡萄牙, ISO8859-15 - Euro)

表 3-12 西歐語言環境

語言環境	使用者介面	區域	字碼集	語言支援
en_GB.ISO8859-1	英文	英國	ISO8859-1	英文 (英國)
en_IE.ISO8859-1	英文	愛爾蘭	ISO8859-1	英文 (愛爾蘭)
fr_BE.ISO8859-1	法文	比利時 - 華隆	ISO8859-1	法文 (比利時 - 華隆文，Unicode 4.0)
fr_BE.UTF-8	法文	比利時 - 華隆	UTF-8	法文 (比利時 - 華隆文，Unicode 4.0)
fr_FR.ISO8859-1	法文	法國	ISO8859-1	法文 (法國)
fr_FR.UTF-8	法文	法國	UTF-8	法文 (法國，Unicode 4.0)
nl_BE.ISO8859-1	英文	比利時 - 佛蘭芒	ISO8859-1	荷蘭文 (比利時 - 佛蘭芒)
nl_NL.ISO8859-1	英文	荷蘭	ISO8859-1	荷蘭文 (荷蘭)

語言環境的多重按鍵組合序列

在許多的 Solaris 語言環境中，特別是歐洲語系和 Unicode 語言環境，能夠使用所謂的「dead 鍵序列」(也稱為 Compose 鍵序列)來輸入各種字元。

Compose 鍵序列是用來輸入具有區別發音符號的字元以及輸入未顯示在鍵盤按鍵上的大寫字元。

下表顯示了 Compose 鍵序列的幾個範例。如需更多關於 Compose 鍵序列的完整資訊，請參閱第 113 頁的「英文/歐洲輸入模式」。

表 3-13 使用 Compose 按鍵所建立的分音符字元

標記	Compose 鍵組合	範例
分音符		Compose A " → 加上分音符的 A
Caron	v	Compose Z v → 加上 Caron 的 Z
短音符	u	Compose G u → 加上短音符的 G
Ogonek	a	Compose A a → 加上 Ogonek 的 A
Cedilla	,	Compose K , → K 字下面加小逗點
註冊標記	R O	Compose R O → 註冊標記
倒置的驚嘆號	!!	Compose !! → 倒置的驚嘆號

注意 – 只有在字元是目前語言環境中字碼集一部分的情況下，Compose 鍵序列才能產生該字元。例如，在 ISO8859-1 字碼集中沒有帶有 caron 的 Z，因此您不能在 en_US.ISO8859-1 語言環境中輸入帶有 caron 的 Z。

Solaris 環境中的鍵盤支援

SPARC 和 Intel Architecture (IA) 平台支援供特定區域使用而具有不同配置的鍵盤。Solaris 作業系統支援下表所列的區域性鍵盤：

表 3-14 區域性鍵盤支援

區域	國家	Sun 鍵盤 (Type 4/5/5c)	Sun 鍵盤 (Type 6)	PC 鍵盤
亞洲	日本	X	X	X
	韓國	X	X	X
	台灣地區	X	X	X
歐洲	比利時	X	X	X
	捷克共和國	X		X
	丹麥	X	X	X
	芬蘭		X	
	法國	X	X	X
	德國	X	X	X
	英國	X	X	X
	希臘	X		X
	匈牙利	X		X
	義大利	X	X	X
	拉脫維亞	X		X
	立陶宛	X		X
	荷蘭	X	X	X
	挪威	X	X	X
波蘭	X		X	

表 3-14 區域性鍵盤支援 (續)

區域	國家	Sun 鍵盤 (Type 4/5/5c)	Sun 鍵盤 (Type 6)	PC 鍵盤
	葡萄牙	X	X	X
	俄羅斯	X	X	X
	西班牙	X	X	X
	瑞典	X	X	X
	瑞士 (法國)	X	X	X
	瑞士 (德國)	X	X	X
	土耳其	X	X	X
美洲	加拿大 (法文)	X	X	X
	拉丁美洲 (西班牙文)	X		
	美國	X	X	X
中東	阿拉伯文	X	X	

鍵盤配置符合國際標準的區域，例如中國大陸，所使用的是提供給美國的鍵盤配置支援，藉以輸入語言環境的字元。基礎鍵盤對映是完全相同的。由於使用多重語言或出現多重鍵盤配置，使得一些如日本、土耳其和瑞士等國家都使用多重鍵盤。

Sun Type 4、5 和 5c 鍵盤透過 Mini DIN 8-pin 的連接來使用 Sun 輸入/輸出介面。Sun Type 6 鍵盤具有兩種版本的介面：

- 透過 Mini DIN 8-pin 連接的 Sun 輸入/輸出
- USB

每個 Sun 鍵盤的背面都印有 Sun 鍵盤的種類。

PC 鍵盤使用各種介面，如 PS/2 或 USB 等。

SPARC 系統上各種鍵盤之間的變更

您可以使用大部分 Sun Type 4、5 和 5c 鍵盤背面的 DIP 切換設定，來變更 Solaris 系統上的鍵盤配置。您可以在 `/usr/openwin/share/etc/keytables/keytable.map` 檔案中找到包括鍵盤類型、名稱以及用於 DIP 切換設定的相對應配置 ID 清單。

注意 – 由於 Type 6 鍵盤背面並沒有 DIP 切換，因此無法變更 Type 6 鍵盤的配置。部分 Type 5 和 5c 的鍵盤，例如美國、美國/UNIX 和日文鍵盤是以 jumper 來取代 DIP 切換。除了使用像是 `xmodmap(1)` 公用程式之外，您不可以使用 SPARC 平台或 IA 平台所提供的公用程式或工具來切換鍵盤。

以下是 Type 4、5 和 5c 鍵盤的配置 ID 值列表 (1 = 向上切換，0 = 向下切換)。

表 3-15 Type 4、5 和 5c 鍵盤的配置

DIP 切換	鍵盤 (Keytable 檔案)	二進制設定
0	美國 (US4.kt)	000000
1	美國 (US4.kt)	000001
2	比利時 (FranceBelg4.kt)	000010
3	加拿大 (Canada4.kt)	000011
4	丹麥 (Denmark4.kt)	000100
5	德國 (Germany4.kt)	000101
6	義大利 (Italy4.kt)	000110
7	荷蘭 (Netherland4.kt)	000111
8	挪威 (Norway4.kt)	001000
9	葡萄牙 (Portugal4.kt)	001001
10 (0x0a)	拉丁美洲/西班牙文 (SpainLatAm4.kt)	001010
11 (0x0b)	瑞典 (SwedenFin4.kt)	001011
12 (0x0c)	瑞士/法文 (Switzer_Fr4.kt)	001100
13 (0x0d)	瑞士/德文 (Switzer_Ge4.kt)	001101
14 (0x0e)	英國 (UK4.kt)	001110
16 (0x10)	韓國 (Korea4.kt)	010000
17 (0x11)	台灣 (Taiwan4.kt)	010001
23	俄文	100001
33 (0x21)	美國 (US5.kt)	100111
34 (0x22)	美國/UNIX (US_UNIX5.kt)	100010
35 (0x23)	法國 (France5.kt)	100011
36 (0x24)	丹麥 (Denmark5.kt)	100100
37 (0x25)	德國 (Germany5.kt)	100101
38 (0x26)	義大利 (Italy5.kt)	100110
39 (0x27)	荷蘭 (Netherland5.kt)	100111
40 (0x28)	挪威 (Norway5.kt)	101000
41 (0x29)	葡萄牙 (Portugal5.kt)	101001

表 3-15 Type 4、5 和 5c 鍵盤的配置 (續)

DIP 切換	鍵盤 (Keytable 檔案)	二進制設定
42 (0x2a)	西班牙 (Spain5.kt)	101010
43 (0x2b)	瑞典 (Sweden5.kt)	101011
44 (0x2c)	瑞士/法文 (Switzer_Fr5.kt)	101101
45 (0x2d)	瑞士/德文 (Switzer_Ge5.kt)	101110
46 (0x2e)	英國 (UK5.kt)	101111
47 (0x2f)	韓國 (Korea5.kt)	101111
48 (0x30)	台灣 (Taiwan5.kt)	110000
49 (0x31)	日本 (Japan5.kt)	110001
50 (0x32) 另請參閱 63 (0x3f)	加拿大/法文 (Canada_Fr5.kt)	110010
51 (0x33)	匈牙利 (Hungary5.kt)	110011
52 (0x34)	波蘭 (Poland5.kt)	110100
53 (0x35)	捷克 (Czech5.kt)	110101
54 (0x36)	俄羅斯 (Russia5.kt)	110110
55 (0x37)	拉脫維亞 (Latvia5.kt)	110111
56 (0x38) 另請參閱 62 (0x3e)	土耳其-Q5 (TurkeyQ5.kt)	111000
57 (0x39)	希臘 (Greece5.kt)	111001
58 (0x3a)	阿拉伯 (Arabic5.kt)	111011
59 (0x3b)	立陶宛 (Lithuania5.kt)	111010
60 (0x3c)	比利時 (Belgian5.kt)	111100
62 (0x3e)	加拿大/法文 (Canada_Fr5_TBITS5.kt)	111111
	加拿大法語	
	波蘭程式設計師	
	愛沙尼亞文	

帶有 4 的 Keytable 檔案名稱適用於 Type 4 鍵盤。帶有 5 的 Keytable 檔案名稱適用於 Type 5 鍵盤。

▼ 如何將鍵盤配置變更為捷克文配置

1. 在表格或 `/usr/openwin/share/etc/keytables/keytable.map` 檔案中決定正確的 DIP 切換 ID (或配置 ID)。 `keytable.map` 檔案中的配置 ID 值為十進位值。
以捷克文來說，配置 ID 為十進位值的 53 (十六進位則為 0x35)。
2. 將配置 ID 轉換為二進位或使用上表欄位中適當的二進位設定值。若要進行基本轉換，可使用計算機公用程式，例如 `dtcalc(1)`。
舉例來說，捷克文鍵盤的正確二進位值為 110101。
3. 關閉並切斷系統的電源。
4. 使用步驟 2 中的二進位值來變更鍵盤背面的 DIP 切換設定。
第一個 DIP 切換位於您的左側。1 表示向上切換，0 表示向下切換。
以下為捷克文鍵盤二進位值 110101 的設定：上上下下上上
5. 打開電源並重新啟動系統。

注意 – 不同於 Type 4 鍵盤，Type 5 和 5c 鍵盤只有五組 DIP 切換。您可以忽略 Type 5 和 5c 鍵盤上的第一個二進位數字。以捷克文 Type 5c 鍵盤為例，正確的 DIP 切換設定值為「上下上下上」，其只使用最後五個數字 10101。

在 Intel 系統上變更鍵盤

在 Intel Architecture 系統上，鍵盤是在 `kdmconfig(1M)` 安裝期間選擇的。若要在安裝後變更此設定，請退出 GUI 桌面環境並進入指令行模式。若您是超級使用者，請輸入 `kdmconfig` 以執行該程式。請遵照指示以取得想要的鍵盤配置。

鍵盤配置圖例

下列圖例顯示的是阿拉伯文鍵盤。

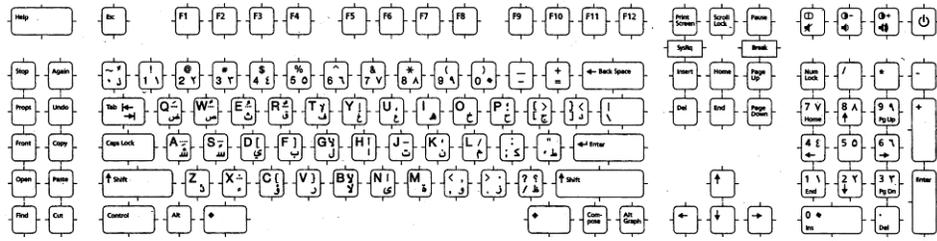


圖 3-1 阿拉伯文鍵盤

下列圖例顯示的是比利時文鍵盤。

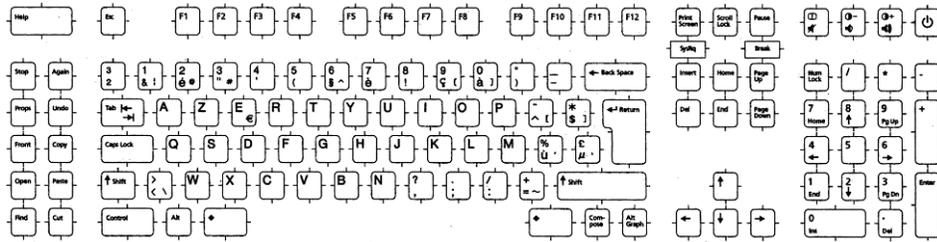


圖 3-2 比利時文鍵盤

下列圖例顯示的是斯拉夫文鍵盤。

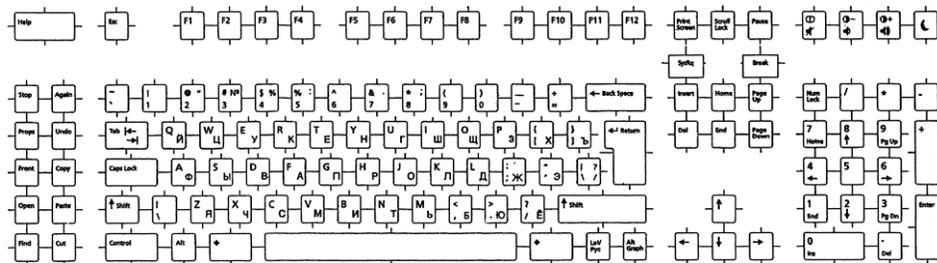


圖 3-3 斯拉夫文 (俄文) 鍵盤

下列圖例顯示的是丹麥文鍵盤。

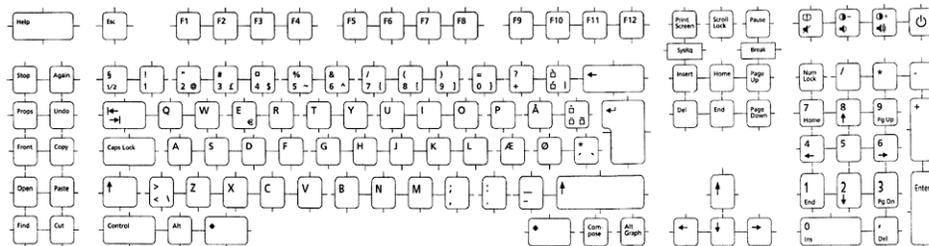


圖 3-4 丹麥文鍵盤

下列圖例顯示的是芬蘭文鍵盤。

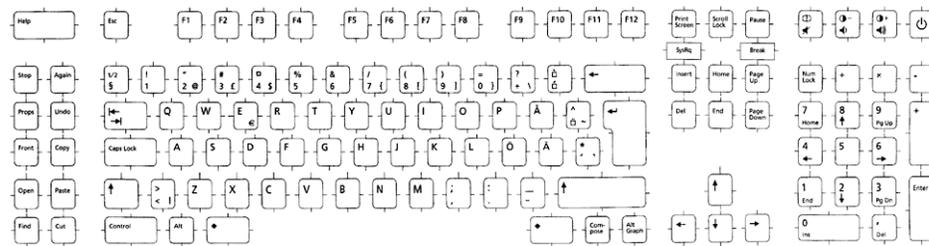


圖 3-5 芬蘭文鍵盤

下列圖例顯示的是法文鍵盤。

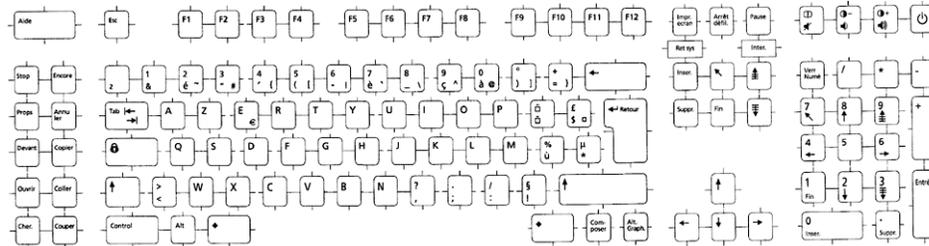


圖 3-6 法文鍵盤

下列圖例顯示的是德文鍵盤。

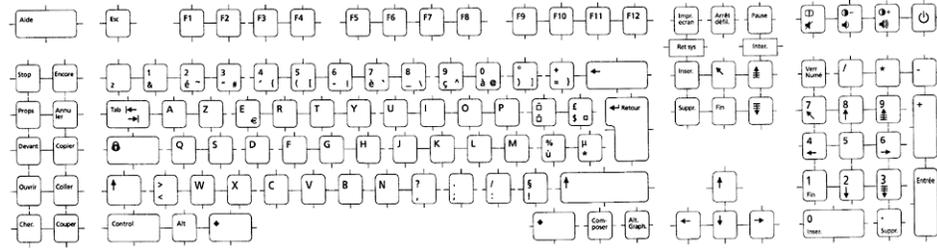


圖 3-7 德文鍵盤

下列圖例顯示的是義大利文鍵盤。

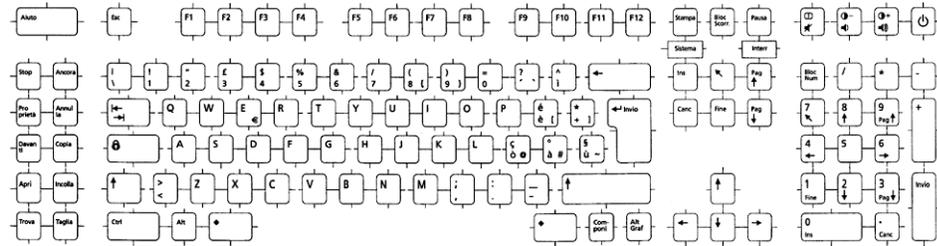


圖 3-8 義大利文鍵盤

下列圖例顯示的是日文鍵盤。

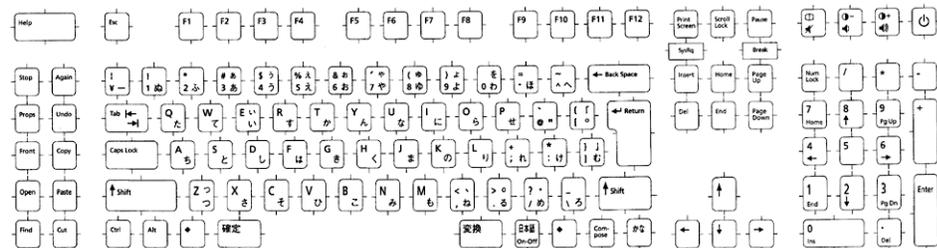


圖 3-9 日文鍵盤

下列圖例顯示的是韓文鍵盤。

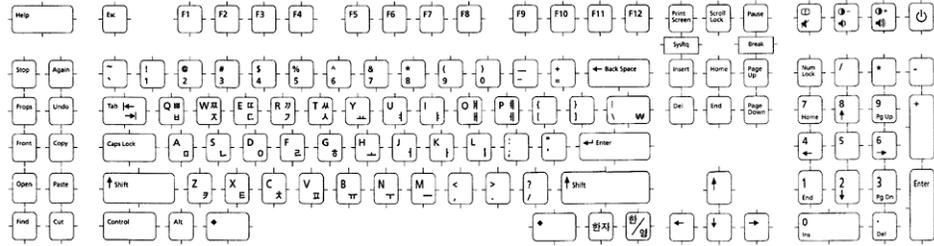


圖 3-10 韓文鍵盤

下列圖例顯示的是荷蘭 (荷蘭文) 鍵盤。

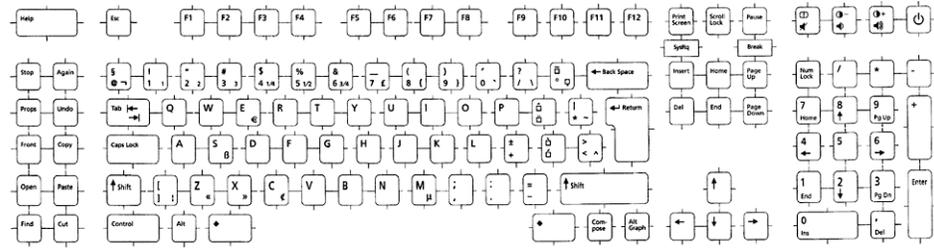


圖 3-11 荷蘭 (荷蘭文) 鍵盤

下列圖例顯示的是挪威文鍵盤。

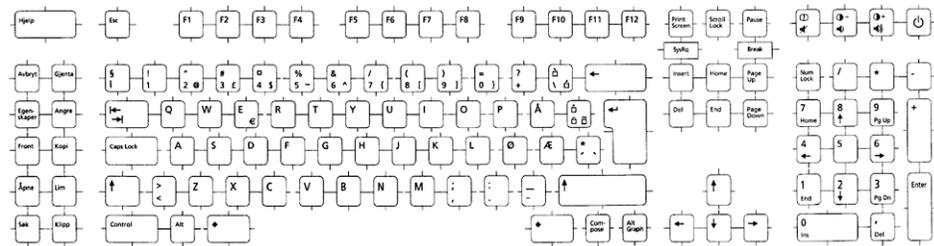


圖 3-12 挪威文鍵盤

下列圖例顯示的是葡萄牙文鍵盤。

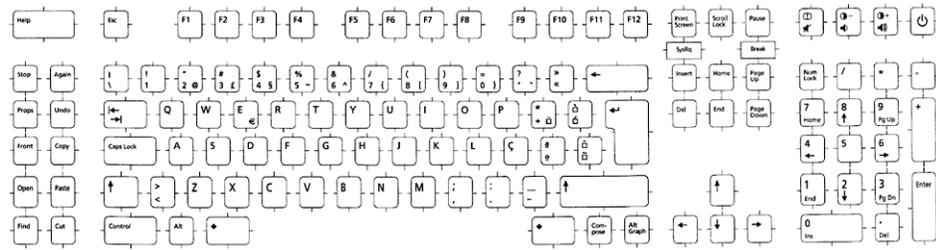


圖 3-13 葡萄牙文鍵盤

下列圖例顯示的是西班牙文鍵盤。

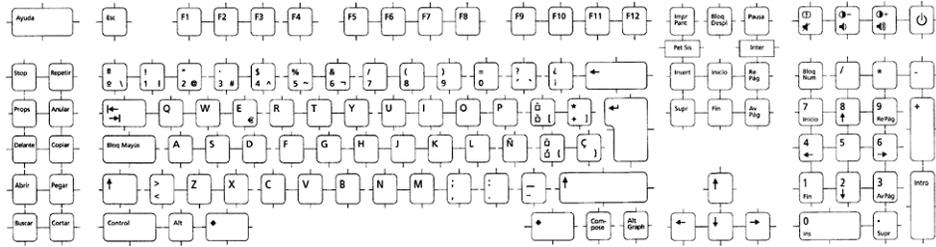


圖 3-14 西班牙文鍵盤

下列圖例顯示的是瑞典文鍵盤。

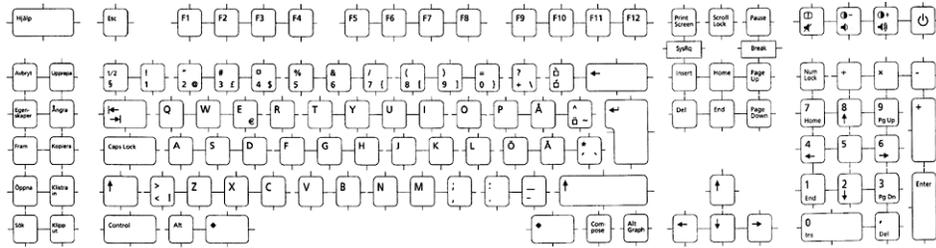


圖 3-15 瑞典文鍵盤

下列圖例顯示的是瑞士 (法文) 鍵盤。

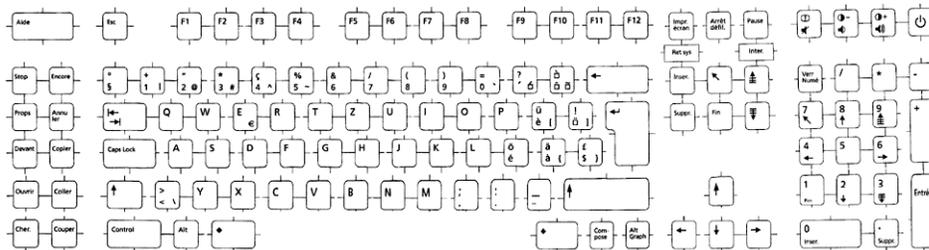


圖 3-16 瑞士 (法文) 鍵盤

下列圖例顯示的是瑞士 (德文) 鍵盤。

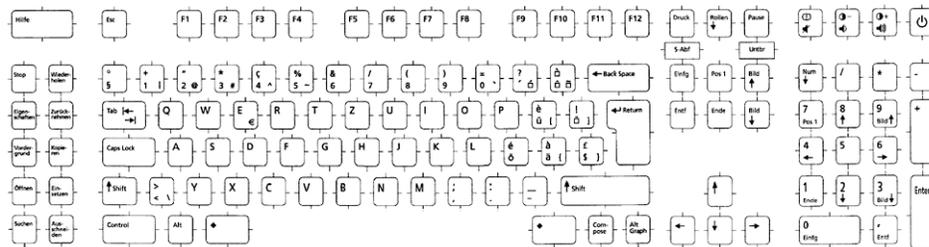


圖 3-17 瑞士 (德文) 鍵盤

下列圖例顯示的是繁體中文鍵盤。

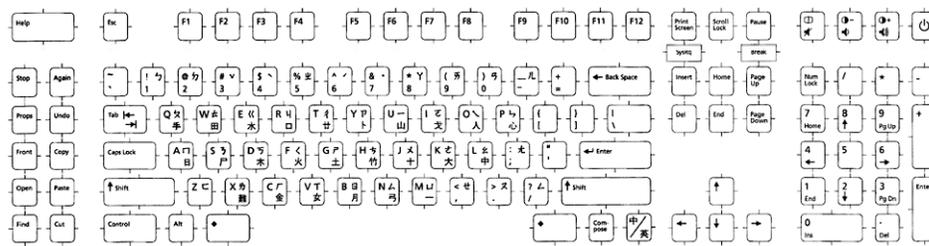


圖 3-18 繁體中文鍵盤

下列圖例顯示的是美國鍵盤。

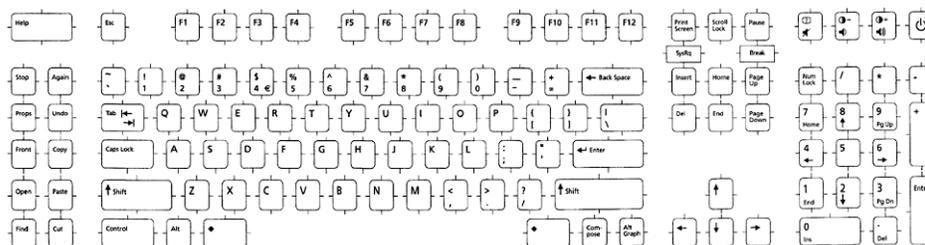


圖 3-22 美國鍵盤

下列圖例顯示的是美國/UNIX 鍵盤。

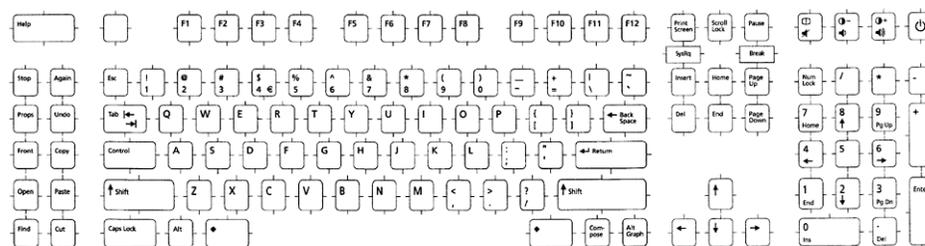


圖 3-23 美國/UNIX 鍵盤

新的 Solaris 鍵盤軟體支援

下列是此發行版本新增的額外鍵盤軟體支援。

- 俄文 Type 6 USB 鍵盤
- 愛沙尼亞文 Type 6 USB 鍵盤
- 加拿大法語 Type 6 USB 鍵盤
- 波蘭程式設計師 Type 5 鍵盤

此軟體可讓俄文、加拿大文、愛沙尼亞文及波蘭文的使用者，修改標準型美國鍵盤的配置，以符合獨特語言的需求。目前還沒有可供額外鍵盤類型使用的硬體。若要使用這個新的鍵盤軟體，請執行本節程序中的步驟。

▼ 如何存取愛沙尼亞文 Type 6 USB 鍵盤支援

1. 將 `/usr/openwin/share/etc/keytables/keytable.map` 檔案中的 `US6.kt` 項目變更為 `Estonia6.kt`。

修改後的項目應如下所示：

```
6          0          Estonia6.kt
```

2. 將下列其中一項加入到 `/usr/openwin/share/lib/locale/iso_8859_15/Compose` 檔案。

修改後的項目應如下所示：

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

3. 重新啟動系統以使變更生效。

▼ 如何存取加拿大法語 Type 6 USB 鍵盤支援

1. 將 `/usr/openwin/share/etc/keytables/keytable.map` 檔案中的 `US6.kt` 項目變更為 `Canada6.kt`。

修改後的項目應如下所示：

```
6          0          Canada6.kt
```

2. 重新啟動系統以使變更生效。

▼ 如何存取波蘭程式設計師 Type 5 鍵盤支援

1. 將 `/usr/openwin/share/etc/keytables/keytable.map` 檔案中的 `Poland5.kt` 項目變更為 `Poland5_pr.kt`。

修改後的項目應如下所示：

```
6          0          Poland5_pr.kt
```

2. 重新啟動系統以使變更生效。

第 4 章

支援的亞洲語言環境

本章提供有關於日文、印度文和泰文的本土化資訊。本章的內容包括：

- 第 73 頁的「日文本土化」
- 第 77 頁的「印度文本土化」
- 第 104 頁的「泰文本土化」

日文本土化

本節說明日文特定語言環境資訊。

日文語言環境

目前 Solaris 環境中有四種可支援不同字元編碼的日文語言環境。其中兩種是以日文 EUC 為基礎的 `ja` 和 `ja_JP.eucJP` 語言環境。`ja_JP.eucJP` 語言環境符合「UI-OSF 日文環境執行協定 1.1 版」，`ja` 語言環境符合起始於較早 Solaris 發行版本的傳統規格。另外兩種是以 PC 漢字字碼 (稱為 `Shift_JIS`) 為基礎的 `ja_JP.PCK` 語言環境以及以 UTF-8 為基礎的 `ja_JP.UTF-8` 語言環境。

請參閱「`eucJP(5)` 線上說明手冊」，以取得日文 EUC 與字元集的對映資訊。請參閱「`PCK(5)` 線上說明手冊」，以取得 PC 漢字字碼與字元集的對映資訊。

日文字元集

支援的日文字元集包括：

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

- JIS X 0212-1990
- JIS X 0213-2000 (僅限 Unicode 4.0 中定義的字元)

JIS X 0212-1990 在 ja_JP.PCK 語言環境中不支援。JIS X 0213-2000 只能在 ja_JP.UTF-8 語言環境中獲得支援。並非所有在 JIS X 0213-2000 中定義的字元都可以使用。只可以使用 Unicode 4.0 字元集中定義的字元。

也支援供應商定義的字元 (VDC) 以及使用者定義的字元 (UDC)。VDC 佔用 JIS X 0208-1990 或 JIS X 0212-1990 中未使用 (保留的) 的字碼點。除了那些配置給 VDC 的字碼點之外，UDC 會佔用與 VDC 相同的字碼點。

日文字型

支援三種日文字型格式：點陣圖字型、TrueType 字型以及 Type1 字型。日文 Type1 字型僅包含用於列印的 JIS X 0212。UDC 也使用 Type 1 字型。

下表說明日文點陣圖字型

表 4-1 日文點陣圖字型

完整家族名稱	子家族	格式	供應商	編碼
sun gothic	R、B	PCF(12,14,16,20,24)		JIS X 0208-1983, JIS X 0201-1976
sun mincho	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

下列表格中說明日文 TrueType 字型。

表 4-2 日文 TrueType 字型

完整家族名稱	子家族	格式	供應商	編碼
ricoh hg gothic b	固定的	TrueType	RICOH	JIS X 0208-1983, JIS X 0201-1976
ricoh hg mincho l	固定的	TrueType	RICOH	JIS X 0208-.1983, JIS X 0201-1976
ricoh hg gothicb sun	固定的，成比例的	TrueType	RICOH	JIS X 0201-176，JIS X 0208-1983，JIS X 0213-2000
ricoh hg minchol sun	固定的，成比例的	TrueType	RICOH	JIS X 0201-1976, JIS X 0208-1983, JIS X 0213-2000
ricoh heiseimin	固定的	TrueType	RICOH	JIS X 0212-1990

日文輸入系統

在目前 Solaris 環境中，ATOK12 是預設的日文輸入系統。若您已安裝日文語言環境，則可以在所有的日文語言環境和所有的 UTF-8 語言環境中使用 ATOK12。所有日文語言環境也可以使用 Wnn6 日文輸入系統。您可以在「桌面」功能表中切換輸入系統。Solaris 1.x BCP 支援可以使用 kkcvt 日文輸入系統。

以下程序說明如何用 ATOK12 輸入法來輸入日文文字。

▼ 如何使用 ATOK 輸入法

1. 按下 **Control-空格鍵** 以開啓輸入轉換。
2. 鍵入您想要轉換文字的假名字元。
例如，鍵入漢字 henkan 的假名。
3. 按下 **空格鍵** 以顯示假名拼寫可轉換的漢字候選字。
4. 鍵入您想要選取的轉換候選字編號。
5. 按下 **Return** 以送出整句假名拼寫的漢字。
或是按下向下箭頭按鍵僅送出所選取的字元。
6. 按下 **Control-空格鍵** 以關閉輸入轉換。

日文終端機的終端機設定

您必須使用終端機設定來使行編輯正確運作，才能在以字元為基礎的終端機 (TTY) 上使用日文語言環境。

- 若您的終端機是 CDE 終端機模擬器 (dtterm)，請在任何一個日文語言環境 (ja、ja_JP.PCK 或 ja_JP.UTF-8) 中，以引數 -defeucw 來使用 stty(1)。例如，在 ja 語言環境中，您必須鍵入：

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

- 若您的終端機並不是 CDE 終端機模擬器，但終端機的字碼集與目前的語言環境相同，則可以引數 -defeucw 來使用 stty(1)。
- 若是終端機的字碼集與目前語言環境的字碼集不相符，請使用 setterm(1) 以啟動字碼轉換。例如，若您在 ja 語言環境中，但是終端機需要 PCK (Shift_JIS 字碼)，您必須鍵入：

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

請參閱「setterm(3CURSES) 線上說明手冊」，以取得詳細資訊。

日文 iconv 模組

有多種日文字碼集轉換支援 iconv(1) 和 iconv(3)。請參閱「iconv_ja(5) 線上說明手冊」，以取得詳細資訊。

使用者定義的字元支援

使用者定義字元公用程式 sdtudctool 可處理外框 (Type1) 和點陣圖 (PCF) 字型。使用舊版中公用程式來建立的 UDC 字型，可以使用像是 fontedit、type3creator 和 fontmanager 等公用程式來遷移。

部份和完整語言環境之間的差異

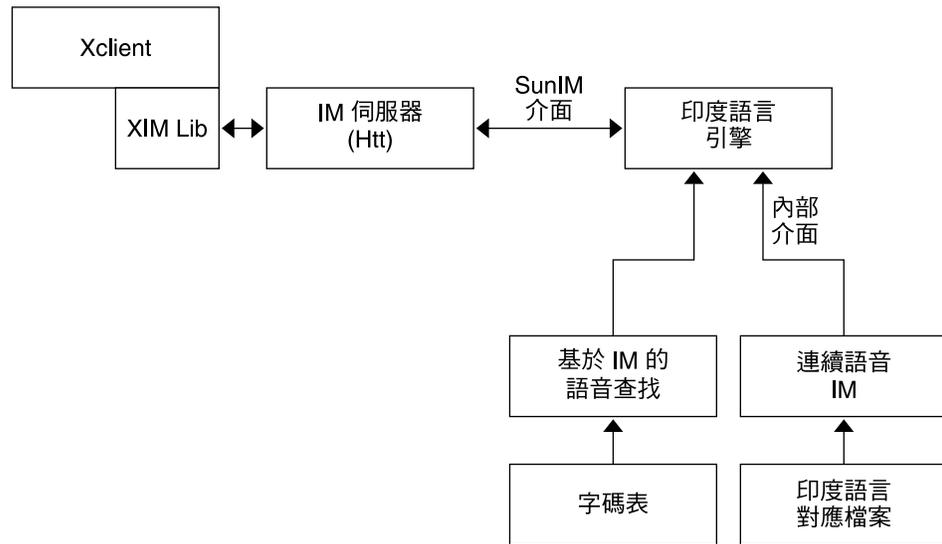
下列元件只可以在 Languages CD 的日文完整語言環境中取得：

- 已翻譯的訊息、說明與線上說明手冊
- Wnn6 日文輸入系統
- 日文 Solaris 1.x BCP 支援
- Mincho (min*) 字體點陣圖字型
- 供列印使用的 JIS X 0212 Type1 字型
- 特定的日文無智型印表機與 postprint 支援
- 傳統日文公用程式，例如 Kanji(1)

印度文本土化

在 UTF-8 語言環境中所支援的所有印度文語言，都可以使用以語音查找為基礎的輸入法 (Shabdalipi) 和連續語音輸入法。使用此輸入法和虛擬鍵盤，可以讓您在所有 CDE 應用程式中輸入印度文字。

下列資料流程展示印度文輸入程序的各項工作。



▼ 如何使用印度文輸入法

1. 按一下輸入狀態區，以顯示輸入法選擇功能表。
2. 在功能表中選取一種輸入法。
或是按下 F6 鍵，從可用的輸入法中選取一種輸入法。
您也可以鍵入 Compose-hi 鍵順序，以選取前一次所使用的輸入法。
3. 按下 F5 鍵以選取您想要使用的印度文字體。
 - a. 針對以鍵盤為基礎 (印度 INSCRIPT 鍵盤) 的輸入法，請使用顯示於第 78 頁的「印度文鍵盤」的鍵盤影像。
 - b. 針對以語音查找為基礎的輸入法，請鍵入與目標字體中字元相對應的英文語音同義字元。

瞭解對映

在第 81 頁的「以連續語音為基礎之輸入法的對映」中的影像，顯示英文代表字元與每個受支援目標字體中同義字碼點之間的對映。CONSONANT 分類是英文代表字元與字體的子音字母之間的對映。VOWEL 分類是英文代表字元與字體的母音字母之間的對映。OTHER 分類包括不具有子音字母和母音字母 (不會依據前後字元而改變形式) 特性之字元的對映。

關鍵字 CONSONANT、VOWEL 和 OTHER 也意味著這些字元是 Unicode 標準的一部分。SPECIAL CONSONANT、SPECIAL VOWEL 或 SPECIAL OTHER 等區段，表示雖然這些字元顯示子音字母、母音字母或其他字元的特性，但這些字元不是正式的 Unicode 標準並且是字型附屬的。這些字元會在 Unicode Private User Area 中指定字碼點值。Solaris UTF-8 語言環境支援這些字元，但此對映可能不適用於其他平台。

這些對映檔與您系統上的對映檔不同，但可以就本討論內容稍加編輯，以移除不需要的關鍵字。

在 VOWELS 和 SPECIAL VOWELS 區段中，會依據上下文來顯示相同英文代表字元的獨立形式或附屬形式。請參閱第 103 頁的「連續語音輸入法的運作方式」。

喀拉拉文字體包含特別的 'CHILLU' 區段，實際上它屬於 SPECIAL OTHER 分類。

以連續語音為基礎之輸入法的對映

下列圖解顯示英文與目標印度文字體中語音同義字元的現有對映。在您熟悉所使用之字體的所有對映之前，請使用這些圖例做為參閱。本文所提供的對映是直覺式的，因此您應該不必查找圖例即可輸入大部分的字元。

注意 – 在這些對映中，如果 '、' 和 'l' 等特殊字元是對映內容的一部分時，必須以 '\ ' 字元跳過。如果沒有，當用於表示相同 UTF-8 字元的代表字元超過一個以上時，'l' 字元會被視為分隔符號。

圖 4-1、圖 4-2 和圖 4-3 顯示英文與子音字母、母音字母和其他字元的孟加拉文對映。

子音字母

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	ছ	h	হ
chh	ছ	n	ন	rr\.	ৱ
j	জ	p	প	rh\.	ৱ
jh	ঝ	f ph	ফ	y\.	য়
ny yn	ঞ	b	ব	v	ব
T	ট	bh	ভ	V	ব
Th	ঠ	m	ম		
D	ড	y	য		

圖 4-1 孟加拉文字子音字母的對映

母音字母	附屬形式	獨立形式
a	অ	।
aa	আ	।
i	ই	ি
ee ii	ঈ	ী
u	উ	ু
oo uu	ঊ	ূ
r \ ^	ঋ	ৃ
rr \ ^	ঌ	্ল
n \ ^	঍	ঁ
nn \ ^	঎	্ন
e	এ	ে
ai	ঐ	ৈ
o	ও	ে
au	ঔ	ৌ

圖 4-2 孟加拉文母音字母的對映

其他字元	
UM	ઁ
\ . N	ઃ
M	૨
H	ઃ
OU	૪
Rs	ઁ
Rs \ .	૪

圖 4-3 孟加拉文其他字元的對映

圖 4-4、圖 4-5 和圖 4-6 顯示英文與子音字母、母音字母和其他字元的古吉拉特文對映。

子音字母

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		

圖 4-4 古吉拉特文子音字母的對映

母音字母	附屬形式	獨立形式
a	ਅ	
aa	ਅ਼	।
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

圖 4-5 古吉拉特文母音字母的對映

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

圖 4-6 古吉拉特文其他字元的對映

圖 4-7、圖 4-8 和圖 4-9 顯示英文與子音字母、母音字母和其他字元的果魯穆其文對映。

子音字母

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		

圖 4-7 果魯穆其文字子音字母的對映

母音字母	附屬形式	獨立形式
a	ਅ	
aa	ਆ	ੴ
i	ਇ	ੴ
ee ii	ਈ	ੴ
u	ਉ	ੴ
oo uu	ਊ	ੴ
E	ਏ	ੴ
ai	ਐ	ੴ
O	ਓ	ੴ
au	ਔ	ੴ

圖 4-8 果魯穆其文母音字母的對映

其他字元	
um	ੴ
\.N	ੴ
UH	ੴ
AD	ੴ
IR	ੴ
UR	ੴ
OM	ੴ

圖 4-9 果魯穆其文其他字元的對映

圖 4-10、圖 4-11 和圖 4-12 顯示英文與子音字母、母音字母和其他字元的北印度文對映。

子音字母

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

圖 4-10 北印度文字子音字母的對映

母音字母	附屬形式	獨立形式
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	ॠ
O	ओ	ॡ
o	ओ	ॢ
au	औ	ॣ

圖 4-11 北印度文母音字母的對映

其他字元			
OM	ॐ		U\~
\.C	.		A\^
M	.		A\~
H	.		\
\.N	.		\ \
V\^	ॡ		EH
U\^	.		o

圖 4-12 北印度文其他字元的對映

圖 4-13、圖 4-14 和圖 4-15 顯示英文與子音字母、母音字母和其他字元的卡納達文對映。

子音字母

k Kh K	ᵏ ᵑ ᵑ	Dh	ᵑ	y	ᵑ
g	ᵑ	N	ᵑ	R r	ᵑ ᵑ
G gh	ᵑ ᵑ	t	ᵑ	rx rh	ᵑ ᵑ
\~G	ᵑ ᵑ	th	ᵑ	l	ᵑ ᵑ
c ch	ᵑ ᵑ	d	ᵑ	ll	ᵑ ᵑ
C CH	ᵑ ᵑ	dh	ᵑ	w v	ᵑ ᵑ
j	ᵑ	n	ᵑ	S sh	ᵑ ᵑ
J jh	ᵑ ᵑ	p	ᵑ	Sh	ᵑ ᵑ
\~J	ᵑ ᵑ	p ph	ᵑ ᵑ	s	ᵑ
T	ᵑ	b	ᵑ	\~h h	ᵑ ᵑ ᵑ ᵑ
Th	ᵑ ᵑ	B bh	ᵑ ᵑ	f	ᵑ ᵑ
D	ᵑ	m	ᵑ		

圖 4-13 卡納達文字音字母的對映

母音字母	附屬形式	獨立形式
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

圖 4-14 卡納達文母音字母的對映

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

圖 4-15 卡納達文其他字元的對映

圖 4-16、圖 4-17 和圖 4-18 顯示英文與子音字母、母音字母和其他字元的喀拉拉文對映。

子音字母

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	特殊子音字母	
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		

圖 4-16 喀拉拉文子音字母的對映

母音字母	附屬形式	獨立形式
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	ൌ
特殊母音字母		
ou	ഔ	ൗ

圖 4-17 喀拉拉文母音字母的對映

其他字元		CHILLU
M	◌	n \ ~ ന്
H	ഃ	N \ ~ ണ്
rr \ ^	ഋ	l \ ~ ൽ
U	ൺ	L \ ~ ൾ
UU	൹	r \ ~ ൿ

圖 4-18 喀拉拉文其他字元的對映

圖 4-19 和圖 4-20 顯示英文與子音字母和母音字母的坦米爾文對映。

子音字母

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

圖 4-19 坦米爾文字音字母的對映

母音字母	附屬形式	獨立形式
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
其他字元		
H	ஹ	

圖 4-20 坦米爾文母音字母的對映

圖 4-21、圖 4-22 和圖 4-23 顯示英文與子音字母、母音字母和其他字元的安得拉邦文對映。

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

圖 4-21 安得拉邦文子音字母的對映

母音字母	附屬形式	獨立形式
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

圖 4-22 安得拉邦文母音字母的對映

其他字元	
\ . N	८
M	०
H	ः
OU	ॆ
LM	ॆ
RR	ॆ
Nu	ॆ

圖 4-23 安得拉邦文其他字元的對映

連續語音輸入法的運作方式

針對每一個印度文字體，‘virama’ 或等號與子音字母的組合，會給出一個形式不完整 (或準備組合的形式) 的子音字母。每當鍵入子音字母相對應的多重鍵組合，子音字母 + virama 形式會輸出並符號化已準備好進行組合的字元。

初始輸入的子音字母會採用原本的不完整形式，一旦後面接著母音字母時，則會成為完整音節或子音字母的變化形式。

兩個連續的子音字母會維持準備好組合的不完整形式。不完整形式可以經由配置引擎當作已組合的單一字元來轉換或維持其可用於每一種語言之語意構造的獨立形式。

以任何母音字母做為字詞的開頭或後面接著其他母音字母時，會以獨立形式顯示。母音字母後面立即接著一個子音字母時，會採用附屬形式。

在任何環境中都不會改變其外觀形狀的字元就是其他字元。這些字元既不是子音字母也不是母音字母。

不做為字元一部分的數字和其他標點符號，是採取一對一對映。

透過這些原則，剖析器 (已寫入上述原則) 會將輸入字元剖析為這些不同分類，並輸出語言特定的 Unicode 字碼點。連續語音輸入法引擎不處理字元配置或轉譯，而是由系統中的其他模組來處理。

泰文本土化

目前 Solaris 環境支援三種泰文輸入層次與四種泰文鍵盤配置。

泰文輸入法

此發行版本支援下列泰文輸入法。在泰文資訊技術標準中，指定這些輸入法做為字元順序檢查。

1. 直接通過層次，沒有輸入檢查
2. 基本輸入檢查層次
3. 詳細輸入檢查層次

與舊版 Solaris 相同，此發行版本的預設是直接通過層次，沒有輸入檢查。

您可以使用 F2 功能鍵，從目前的輸入層次切換到下一個輸入層次。

泰文鍵盤配置

泰文輸入法支援四種不同的鍵盤配置。

- Kedmanee (TIS820-2531) 鍵盤配置。此 Kedmanee 配置是為打字機設計的，不適用於電腦鍵盤。由於打字機鍵盤上的按鍵有限，因此這個配置無法使用部分泰文特殊字元。TIS820-2531 在電腦鍵盤上採用 Kedmanee 配置。



- TIS820-2538 鍵盤配置。此增強型 Kedmanee 配置是 TIS820-2531 配置的更新版本，其中包含原本無法在舊版 Kedmanee 配置中使用的部分泰文特殊字元。TIS820-2538 是目前唯一由泰國工業標準局發佈的泰文鍵盤配置標準。



- Pattajoti 鍵盤配置。此 Pattajoti 配置也是針對打字機設計的，但具有較佳的手指受力分佈區域。



- 可配置的鍵盤配置。使用者可定義泰文輸入法的鍵盤配置。

泰文輸入法輔助視窗

此泰文輸入法輔助視窗支援下列功能與公用程式：

- 輸入層次切換。您可以按一下輔助板上的輸入層次按鈕，以選擇直接通過、基本或詳細來做為您的輸入層次。
- 泰文虛擬鍵盤。您可以按一下鍵盤按鈕來顯示泰文虛擬鍵盤，以輸入泰文字元。

第 5 章

UTF-8 語言環境支援概述

本章提供 UTF-8 語言環境支援的概述。本章包含以下主題：

- 第 107 頁的「Unicode 概述」
- 第 108 頁的「Unicode 語言環境：en_US.UTF-8 支援」
- 第 110 頁的「關於桌面輸入法」
- 第 140 頁的「系統環境」
- 第 144 頁的「字碼轉換」
- 第 145 頁的「DtMail 支援」
- 第 147 頁的「程式設計環境」

Unicode 概述

Unicode 是通用的字元編碼標準，用於顯示電腦處理的文字。Unicode 完全相容於國際標準 ISO/IEC 10646-1:2000 和 ISO/IEC 10646-2:2001，並且包含與 ISO/IEC 10646 相同的所有字元和編碼點。Unicode 標準提供其他關於字元和其用法的資訊。所有與 Unicode 相符的實施方法同樣符合 ISO/IEC 10646。

Unicode 提供了一致性的方法來進行多語一般文字的編碼，並有助於國際文字檔案的交換。處理多語文字的電腦使用者、商務人士、語言學家、研究員和科學家等都發現，「Unicode 標準」能大幅簡化他們的工作。經常使用數學符號和其他技術性字元的數學家 and 技術人員也發現「Unicode 標準」確實相當便利。

Unicode 透過 17 個 16 位元的平面最多可支援 1,114,112 個字碼點。每個平面可支援 65,536 個不同的字碼點。

在 Unicode 可以支援的上百萬個字碼點之中，版本 4.0 目前在平面 0、1、2 和 14 定義了 96,382 個字元。平面 15 和 16 用於個人使用字元，也就是使用者自行定義的字元。15 和 16 平面可合併支援總計 131,068 個使用者自行定義字元。

下列任何一種字元編碼方案都可以用來對 Unicode 進行編碼：

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

UTF-8 是一種可變長度的 Unicode 編碼格式，能有效地保存 ASCII 字元編碼值。此格式的使用方法如同 Solaris Unicode 語言環境中的檔案碼一般。

UTF-16 為 16 位元的 Unicode 編碼格式。在 UTF-16 中，達到 65,535 的字元將編碼為單一 16 位元值。對映超過 65,535 到 1,114,111 之間的字元則編碼為成對的 16 位元值 (代用值)。

UTF-32 為固定長度的 21 位元 Unicode 編碼格式，通常出現在 32 位元容器或資料類型中。此格式在 Solaris Unicode 語言環境中係做為程序碼 (寬字元碼) 使用。

如需關於「Unicode 標準」、ISO/IEC 10646 和其各種代表格式的詳細資訊，請參閱下列資源：

- Unicode 協會的「The Unicode Standard, Version 4.0」
- ISO/IEC 10646-1:2000，資訊技術通用的多重八位元組字元集 (UCS) - Part 1：架構和基本多語平面
- ISO/IEC 10646-2：資訊技術通用的多重八位元組字元集 (UCS) - Part 2：字體和符號的輔助多語平面、CJK 表意文字的輔助平面、特殊目的平面
- Unicode 協會的網站位於 <http://www.unicode.org/>。

Unicode 語言環境：en_US.UTF-8 支援

Unicode/UTF-8 語言環境支援 Unicode 4.0。en_US.UTF-8 語言環境透過使用 UTF-8 做為其字碼集的方法，來提供多字體的處理支援。此語言環境負責處理多重字體中的輸入和輸出文字，其為 Solaris 作業系統中第一個擁有此能力的語言環境。其他 UTF-8 語言環境的功能與 en_us.UTF-8 的功能類似。以下關於 en_US.UTF-8 的討論也可以套用到這些語言環境中。

注意 – UTF-8 是 Unicode/ISO/IEC 10646-1 的檔案系統安全「通用字元集轉換格式」，由 X/Open-Uniform Joint Internationalization Working Group (XoJIG) 於 1992 年制定，並在 1996 年由 ISO 和 IEC 認可做為 ISO/IEC 10646-1:1993 的第 2 修正案。Unicode 協會、國際標準組織 (ISO) 以及國際電子電機委員會 (IEC) 採用這個標準做為 Unicode 4.0 和 ISO/IEC 10646-1 的一部分。

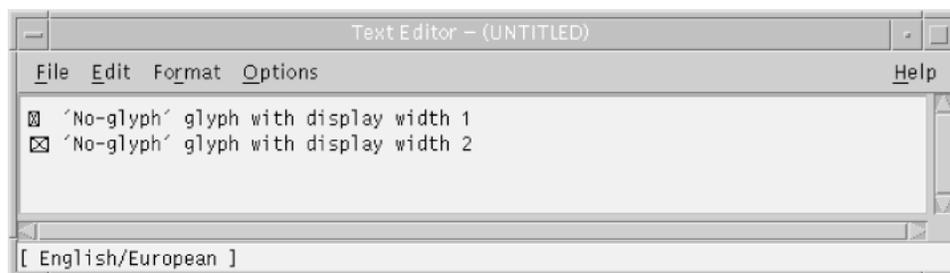
Solaris 環境中的 Unicode 語言環境支援定義於 Unicode 4.0 以及 ISO/IEC 10646-1、10646-2 之中每一個字碼點值的處理方式。支援的字體包括泛歐字體和亞洲字體以及阿拉伯文、希伯來文、印度文和泰文等複合文字版面配置的字體。

注意 – 部分 Unicode 語言環境包含額外的 Kanji 或 Hanzi 文字影像，特別是亞洲語言環境。

由於字型資源的限制，目前 Solaris Unicode 語言環境只包含下列字元集中的字元文字影像：

- ISO 8859-1 (大部分西歐語言，例如英文、法文、西班牙文、德文)
- ISO 8859-2 (大部分中歐語言，例如捷克文、波蘭文、匈牙利文)
- ISO 8859-4 (斯堪的納維亞文和波羅的海文)
- ISO 8859-5 (俄文)
- ISO 8859-6 (阿拉伯文，包含許多表示格式的字元文字影像)
- ISO 8859-7 (希臘文)
- ISO 8859-8 (希伯來文)
- ISO 8859-9 (土耳其文)
- TIS 620.2533 (泰文，包含許多表示格式的字元文字影像)
- ISO 8859-15 (具有 euro 記號的西歐語系)
- GB 2312-1980 (簡體中文)
- JIS X 0201-1976，JIS X 0208-1990 (日文)
- KSC 5601-1992 附錄 3 (韓文)
- GB 18030 (簡體中文)
- HKSCS (繁體中文，香港)
- Big5 (繁體中文，台灣)
- IS 13194.1991，亦為 ISCII (北印度文，包含許多表示格式的字元文字影像)

若您想檢視的字元在 en_US.UTF-8 語言環境中並沒有相對應的文字影像，則語言環境將會顯示 no-glyph 的文字影像，如以下圖例所示：



在安裝時該語言環境將為可選項，並指定為系統預設語言環境。

64 位元或 32 位元的 Solaris 系統都提供相同等級的 en_US.UTF-8 語言環境支援。

注意 – Motif 和 CDE 桌面應用程式及程式庫都支援 en_US.UTF-8 語言環境。不過，XView™ 和 OLIT 程式庫不支援 en_US.UTF-8 語言環境。

關於桌面輸入法

CDE 使用 Xm 工具組，提供在國際化應用程式中鍵入本土化輸入的功能。可啓用 XmText [Field] widgets 以建立與每個語言環境輸入法之間的介面。由於某些語言環境的文字撰寫方式為由右到左、由上到下等等，因此輸入法已進行國際化。在相同的應用程式中，您可以使用套用數種字型的不同輸入法。

預先編輯區可顯示已預先編輯的字串。可以在四種模式中寫入文字：

- OffTheSpot
- OverTheSpot (預設)
- Root
- 無

在 OffTheSpot 模式中，其位置就位於狀態區右邊主視窗的正下方。在 OverTheSpot 模式中，預先編輯區就位於游標點的位置。在 Root 模式中，預先編輯區和狀態區是與用戶端視窗分隔開來的。

如需詳細資訊，請參閱「VendorShell(3X) 線上說明手冊」中的 XmNpreeditType 資源說明。

注意 – 在目前 Solaris 環境中，亞洲當地的輸入法共分為簡體/繁體中文、日文和韓文。這些是除了 Unicode 語言環境目前的多字體輸入法之外的方法。

第 111 頁的「使用輸入模式」包含已選輸入法、輸入法的使用以及如何切換輸入法的說明。

字體選擇和輸入模式

Solaris Unicode 語言環境支援多種字體。每個 Unicode 語言環境都有 14 個輸入模式。

- 英文/歐洲
- 斯拉夫文
- 希臘文

- 阿拉伯文
- 希伯來文
- 泰文
- 日文
- 韓文
- 簡體中文
- 繁體中文
- 繁體中文 (香港)
- 印度文
- Unicode 十六進位和八進位碼輸入法
- 查表輸入法

使用輸入模式

您可以使用 Compose 鍵組合或透過輸入模式選擇視窗來切換至特定的輸入模式。若要使用輸入模式選擇視窗，請按一下應用程式視窗左下角的狀態區。輸入模式選擇視窗顯示於下圖中。

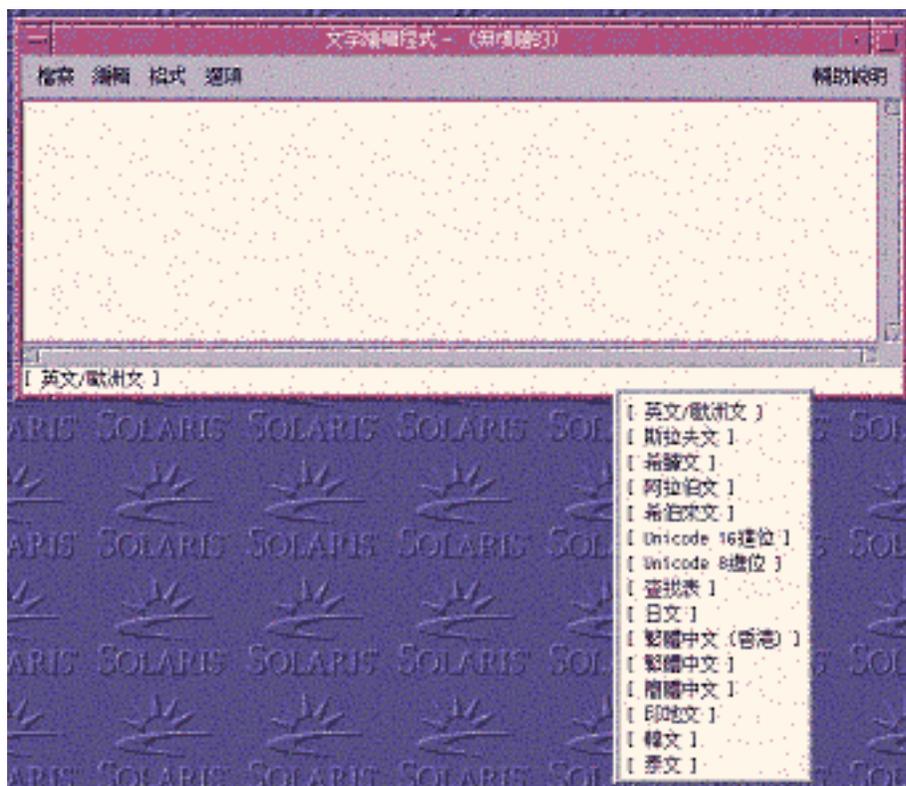


圖 5-1 輸入模式選擇視窗

輸入模式切換鍵序列

您可以使用列於表 5-1 中的鍵序列來將目前的輸入模式變更為新的輸入模式。使用這些鍵序列的唯一限制就是，如果您正處於任何一種亞洲輸入模式中，就必須同時按住 **Control** 鍵和空格鍵以切換回英文/歐洲輸入模式。一旦進入英文/歐洲輸入模式之後，您就可以使用鍵序列隨意在任何其他輸入模式中進行切換。

下列的鍵序列將顯示如何從英文/歐洲輸入模式切換至斯拉夫文輸入模式：

1. 按住 **Compose** 鍵。
2. 按住 **C** 鍵後放開。
3. 按住 **C** 鍵。

表 5-1 輸入模式切換鍵序列

鍵序列	輸入模式
Control-空格鍵	英文/歐洲
Compose c c	斯拉夫文
Compose g g	希臘文
Compose a r	阿拉伯文
Compose h h	希伯來文
Compose t t	泰文
Compose h i	印度文
Compose i n	印度文
Compose j a	日文
Compose k o	韓文
Compose s c	簡體中文
Compose t c	繁體中文
Compose h k	繁體中文 (香港)
Compose u o	Unicode 八進位碼輸入法
Compose u h	Unicode 十六進位碼輸入法
Compose l l	查表輸入法

英文/歐洲輸入模式

英文/歐洲輸入模式包含英文字母加上具有區別發音符號的字元 (例如 á、è、î、õ 和 ü) 以及歐洲字體的字元 (例如 ĵ、Œ、ı)。

此輸入模式是所有應用程式的預設模式。輸入模式顯示於 GUI 應用程式視窗的左下角。

若要插入具有區別發音符號的字元或 Latin-1、Latin-2、Latin-4、Latin-5 和 Latin-9 中的特殊字元，您必須依照以下範例所述輸入 Compose 鍵序列。

若要顯示 Ä 字元：

1. 按住 Compose 鍵後放開。
2. 同時按住 Shift 和 A 鍵。放開 Shift-A。
3. 按住 " 鍵後放開。

若要顯示 ĵ 字元：

1. 按住 Compose 鍵後放開。
2. 按住 ? 鍵後放開。
3. 按住 ? 鍵後放開。

若您的鍵盤上沒有 Compose 鍵，您可以同時按住 Control 鍵和 Shift 鍵以模擬該鍵的操作。

您可以使用下列任一種輸入序列來輸入所在語言環境的歐元貨幣符號 (Unicode value U+20AC)：

- AltGraph 和 E
- AltGraph 和 4
- AltGraph 和 5

使用這些輸入序列時，您必須同時按住兩個鍵。若您的鍵盤上沒有 AltGraph 鍵，您可以使用某些替代的 euro 符號輸入序列，例如 Compose e = 或 Compose c = 。

以下表格顯示出 Solaris 作業系統中最常用的 Latin-1、Latin-2、Latin-3、Latin-4、Latin-5 和 Latin-9 字體輸入的組合序列。

以下表格列出了常用的 Latin-1 Compose 鍵序列。

表 5-2 常用 Latin-1 Compose 鍵序列

按一下 Compose，然後按住再放開	然後按住之後放開	結果
空格鍵	空格鍵	不中斷空間
s	1	上標 1
s	2	上標 2
s	3	上標 3
!	!	倒置的驚嘆號
x	o	貨幣符號 ¢
p	!	段落符號 ¶
/	u	mu u
'	"	尖音符號 ´
,	, (逗號)	C 加尾形符號 Ç
	"	分音符號 ¨
-	^	長音符號 ˆ
o	o	度數符號 °
x	x	乘號 x

表 5-2 常用 Latin-1 Compose 鍵序列 (續)

按一下 Compose，然後按住再 放開	然後按住之後放開	結果
+	-	加減號 ±
-	-	連字號 -
-	:	除號 ÷
-	a	序數 (陰性) ^a
-	o	序數 (陽性) ^o
-	, (逗號)	Not 符號 ¬
.	.	中點 ·
1	2	普通分數 ½
1	4	普通分數 ¼
3	4	普通分數 ¾
<	<	左向雙箭頭引號 «
>	>	右向雙箭頭引號 »
?	?	倒置的問號 ¿
A	' (後引號)	A 加抑音符 À
A	' (單引號)	A 加尖音符 Á
A	*	A 加上環 Å
A		A 加分音符 Ä
A	^	A 加抑揚音符 Â
A	~	A 加波浪符號 Ã
A	E	AE 雙元音符號 Æ
C	, (逗號)	C 加尾形符號 Ç
C	o	版權符號 ©
D	-	大寫 eth ð
E	' (後引號)	E 加抑音符 È
E	'	E 加尖音符 É
E		E 加分音符 Ê
E	^	E 加抑揚音符 Ê
I	' (後引號)	I 加抑音符 Ì

表 5-2 常用 Latin-1 Compose 鍵序列 (續)

按一下 Compose，然後按住再放開	然後按住之後放開	結果
I	'	I 加尖音符 í
I	"	I 加分音符 ï
I	^	I 加抑揚音符 î
L	-	英鎊符號 £
N	~	N 加波浪符號 Ñ
O	` (後引號)	O 加抑音符 Ò
O	'	O 加尖音符 Ó
O	/	O 加斜線 Ø
O	"	O 加分音符 Ö
O	^	O 加抑揚音符 Ô
O	~	O 加波浪符號 Õ
R	o	註冊商標 ®
T	H	Thorn þ
U	` (後引號)	U 加抑音符 Ù
U	'	U 加尖音符 Ú
U		U 加分音符 Ü
U	^	U 加抑揚音符 Û
Y	'	Y 加尖音符 ý
Y	-	日圓符號 ¥
a	` (後引號)	a 加抑音符 à
a	'	a 加尖音符 á
a	nls=gb2312*	a 加上環 â
a		a 加分音符 ä
a	~	a 加波浪符號 ã
a	^	a 加抑揚音符 â
a	e	ae 雙元音符號 æ
c	, (逗號)	c 加尾形符號 ç
c	/	分值符號 ¢

表 5-2 常用 Latin-1 Compose 鍵序列 (續)

按一下 Compose，然後按住再 放開	然後按住之後放開	結果
c	o	版權符號 ©
d	-	eth ð
e	' (後引號)	e 加抑音符 è
e	'	e 加尖音符 é
e		e 加分音符 ë
e	^	e 加抑揚音符 ê
i	' (後引號)	i 加抑音符 ì
i	'	i 加尖音符 í
i		i 加分音符 ï
i	^	i 加抑揚音符 î
n	~	n 加波浪符號 ñ
o	' (後引號)	o 加抑音符 ò
o	'	o 加尖音符 ó
o	/	o 加斜線 ø
o		o 加分音符 ö
o	^	o 加抑揚音符 ô
o	~	o 加波浪符號 õ
s	s	德文中的雙 s ß 也就是高半音的 S
t	h	thorn þ
u	' (後引號)	u 加抑音符 ù
u	'	u 加尖音符 ú
u		u 加分音符 ü
u	^	u 加抑揚音符 û
y	'	y 加尖音符 ý
y		y 加分音符 ÿ
l	l	斷槓 ¡

以下表格列出了常用的 Latin-2 Compose 鍵序列。

表 5-3 常用的 Latin-2 Compose 鍵序列

按一下 Compose，然後按住再放開	按住後放開	結果
k	k	kra
A	—	A 加長音符號
E	—	E 加長音符號
E	.	E 字上加一點
G	,	G 加尾形符號
I	—	I 加長音符號
I	~	I 加波浪符號
I	a	I 加 ogonek
K	,	K 加尾形符號
L	,	L 加尾形符號
N	,	N 加尾形符號
O	—	O 加長音符號
R	,	R 加尾形符號
T		T 加一筆畫
U	~	U 加波浪符號
U	a	U 加 ogonek
U	—	U 加長音符號
N	N	Eng
a	—	a 加長音符號
e	—	e 加長音符號
e	.	e 字上加一點
g	,	g 加尾形符號
i	—	i 加長音符號
i	~	i 加波浪符號
i	a	i 加 ogonek
k	,	k 加尾形符號
l	,	l 加尾形符號
n	,	n 加尾形符號

表 5-3 常用的 Latin-2 Compose 鍵序列 (續)

按一下 Compose，然後按住再放開	按住後放開	結果
o	—	o 加長音符號
r	,	r 加尾形符號
t		t 加一筆畫
u	~	u 加波浪符號
u	a	u 加 ogonek
u	—	E 加長音符號
n	n	eng

以下表格列出了常用的 Latin-3 Compose 鍵序列。

表 5-4 常用的 Latin-3 Compose 鍵序列

按一下 Compose，然後按住再放開	按住後放開	結果
C	>	C 加抑揚音符
C	.	C 字上加一點
G	>	G 加抑揚音符
G	.	G 字上加一點
H	>	H 加抑揚音符
J	>	j 加抑揚音符
S	>	S 加抑揚音符
U	u	U 加短音符
c	>	c 加抑揚音符
c	.	c 字上加一點
g	>	g 加抑揚音符
g	.	g 字上加一點
h	>	h 加抑揚音符
j	>	j 加抑揚音符
s	>	s 加抑揚音符
u	u	u 加短音符

以下表格列出了常用的 Latin-4 Compose 鍵序列。

表 5-5 常用的 Latin-4 Compose 鍵序列

按一下 Compose，然後按住再放開	按住後放開	結果
k	k	kra
A	–	A 加長音符號
E	–	E 加長音符號
E	◦	E 字上加一點
G	、	G 加尾形符號
I	–	I 加長音符號
I	~	I 加波浪符號
I	a	I 加 ogonek
K	、	K 加尾形符號
L	、	L 加尾形符號
N	、	N 加尾形符號
O	–	O 加長音符號
R	、	R 加尾形符號
T	l	T 加一筆畫
U	~	U 加波浪符號
U	a	U 加 ogonek
U	–	U 加長音符號
N	N	Eng
a	–	a1 加長音符號
e	–	e 加長音符號
e	◦	e 字上加一點
g	、	g 加尾形符號
i	–	i 加長音符號
i	~	i 加波浪符號
i	a	i 加 ogonek
k	、	k 加尾形符號

表 5-5 常用的 Latin-4 Compose 鍵序列 (續)

按一下 Compose，然後按住再放開	按住後放開	結果
l	、	l 加尾形符號
n	、	n 加尾形符號
o	_	o 加長音符號
r	、	r 加尾形符號
t		t 加一筆畫
u	~	u 加波浪符號
u	a	u 加 ogonek
u	_	E 加長音符號
n	n	eng

以下表格列出了常用的 Latin-5 Compose 鍵序列。

表 5-6 常用的 Latin-5 Compose 鍵序列

按一下 Compose，然後按住再放開	按住後放開	結果
G	u	G 加短音符
I	o	l 字上加一點
g	u	g 加短音符
i	o	i 字去掉上面的點

以下表格列出了常用的 Latin-9 Compose 鍵序列。

表 5-7 常用 Latin-9 Compose 鍵序列

按一下 Compose，然後按住再放開	按住後放開	結果
o	e	連音符 oe
O	E	連音符 OE
Y	“	Y 分音符

如果您使用具有重音固定鍵的鍵盤，請使用下列 Compose 鍵序列。「dead_acute」和這一類的按鍵名稱，源自於 X11 已註冊的 X_dead_acute 的實鍵名稱，其列示於 `/usr/openwin/include/X11/keysymdef.h`。SunFA_Circum 和這一類的按鍵名稱，源自於 Sun 定義的 X11 實鍵名稱，例如 `/usr/openwin/include/X11/Sunkeysym.h` 中顯示的 SunXK_FA_Circum。

表 5-8 以重音固定鍵為基礎的 Compose 鍵序列

按住後放開	按住後放開	結果
dead_grave	空格鍵	抑音符
dead_acute	撇號	尖音符
dead_acute	空格鍵	撇號
dead_diaeresis	雙引號	分音符
dead_diaeresis	空格鍵	分音符
dead_circumflex	空格鍵	加抑揚音符的重音
dead_circumflex	斜線	垂直線
dead_circumflex	0	度數符號
dead_circumflex	1	上標 1
dead_circumflex	2	上標 2
dead_circumflex	3	上標 3
dead_circumflex	句號	中點
dead_circumflex	驚嘆號	斷槓
dead_circumflex	減號	長音符號
dead_circumflex	底線	長音符號
dead_cedilla	逗號	尾形符號
dead_cedilla	減號	Not 符號
dead_tilde	空格鍵	波浪號
dead_grave	A	A 加抑音符
dead_acute	A	A 加尖音符
dead_circumflex	A	A 加抑揚音符
dead_tilde	A	A 加波浪號
dead_diaeresis	A	A 加分音符
dead_grave	a	a 加抑音符
dead_acute	a	a 加尖音符
dead_circumflex	a	a 加抑揚音符
dead_tilde	a	a 加波浪號
dead_diaeresis	a	a 加分音符

表 5-8 以重音固定鍵為基礎的 Compose 鍵序列 (續)

按住後放開	按住後放開	結果
dead_cedilla	C	C 加尾形符號
dead_cedilla	c	c 加尾形符號
dead_grave	E	E 加抑音符
dead_acute	E	E 加尖音符
dead_circumflex	E	E 加抑揚音符
dead_diaeresis	E	E 加分音符
dead_grave	e	e 加抑音符
dead_acute	e	e 加尖音符
dead_circumflex	e	e 加抑揚音符
dead_diaeresis	e	e 加分音符
dead_grave	I	I 加抑音符
dead_acute	I	I 加尖音符
dead_circumflex	I	I 加抑揚音符
dead_diaeresis	I	I 加分音符
dead_grave	i	i 加抑音符
dead_acute	i	i 加尖音符
dead_circumflex	i	i 加抑揚音符
dead_diaeresis	i	i 加分音符
dead_tilde	N	N 加波浪號
dead_tilde	n	n 加波浪號
dead_grave	O	O 加抑音符
dead_acute	O	O 加尖音符
dead_circumflex	O	O 加抑揚音符
dead_tilde	O	O 加波浪號
dead_diaeresis	O	O 加分音符
dead_grave	o	o 加抑音符
dead_acute	o	o 加尖音符
dead_circumflex	o	o 加抑揚音符
dead_tilde	o	o 加波浪號

表 5-8 以重音固定鍵為基礎的 Compose 鍵序列 (續)

按住後放開	按住後放開	結果
dead_diaeresis	o	o 加分音符
dead_cedilla	S	S 加尾形符號
dead_cedilla	s	s 加尾形符號
dead_grave	U	U 加抑音符
dead_acute	U	U 加尖音符
dead_circumflex	U	U 加抑揚音符
dead_diaeresis	U	U 加分音符
dead_grave	u	u 加抑音符
dead_acute	u	u 加尖音符
dead_circumflex	u	u 加抑揚音符
dead_diaeresis	u	u 加分音符
dead_acute	Y	Y 加尖音符
dead_acute	y	y 加尖音符
dead_diaeresis	y	y 加分音符
SunFA_Grave	空格鍵	抑音符
SunFA_Grave	A	A 加抑音符
SunFA_Grave	a	a 加抑音符
SunFA_Grave	E	E 加抑音符
SunFA_Grave	e	e 加抑音符
SunFA_Grave	I	I 加抑音符
SunFA_Grave	i	i 加抑音符
SunFA_Grave	O	O 加抑音符
SunFA_Grave	o	o 加抑音符
SunFA_Grave	U	U 加抑音符
SunFA_Grave	u	u 加抑音符
SunFA_Acute	撇號	尖音符
SunFA_Acute	空格鍵	撇號
SunFA_Acute	A	A 加尖音符
SunFA_Acute	a	a 加尖音符

表 5-8 以重音固定鍵為基礎的 Compose 鍵序列 (續)

按住後放開	按住後放開	結果
SunFA_Acute	C	C 加尖音符
SunFA_Acute	c	c 加尖音符
SunFA_Acute	E	E 加尖音符
SunFA_Acute	e	e 加尖音符
SunFA_Acute	I	I 加尖音符
SunFA_Acute	i	i 加尖音符
SunFA_Acute	L	L 加尖音符
SunFA_Acute	l	l 加尖音符
SunFA_Acute	N	N 加尖音符
SunFA_Acute	n	n 加尖音符
SunFA_Acute	O	O 加尖音符
SunFA_Acute	o	o 加尖音符
SunFA_Acute	R	R 加尖音符
SunFA_Acute	r	r 加尖音符
SunFA_Acute	S	S 加尖音符
SunFA_Acute	s	s 加尖音符
SunFA_Acute	U	U 加尖音符
SunFA_Acute	u	u 加尖音符
SunFA_Acute	Y	Y 加尖音符
SunFA_Acute	y	y 加尖音符
SunFA_Acute	Z	Z 加尖音符
SunFA_Acute	z	z 加尖音符
SunFA_Cedilla	逗號	尾形符號
SunFA_Cedilla	減號	Not 符號
SunFA_Cedilla	C	C 加尾形符號
SunFA_Cedilla	c	c 加尾形符號
SunFA_Cedilla	G	G 加尾形符號
SunFA_Cedilla	g	g 加尾形符號
SunFA_Cedilla	K	K 加尾形符號

表 5-8 以重音固定鍵為基礎的 Compose 鍵序列 (續)

按住後放開	按住後放開	結果
SunFA_Cedilla	k	k 加尾形符號
SunFA_Cedilla	L	L 加尾形符號
SunFA_Cedilla	l	l 加尾形符號
SunFA_Cedilla	N	N 加尾形符號
SunFA_Cedilla	n	n 加尾形符號
SunFA_Cedilla	R	R 加尾形符號
SunFA_Cedilla	r	r 加尾形符號
SunFA_Cedilla	S	S 加尾形符號
SunFA_Cedilla	s	s 加尾形符號
SunFA_Cedilla	T	T 加尾形符號
SunFA_Cedilla	t	t 加尾形符號
SunFA_Circum	空格鍵	加抑揚音符的重音
SunFA_Circum	0	度數符號
SunFA_Circum	1	上標 1
SunFA_Circum	2	上標 2
SunFA_Circum	3	上標 3
SunFA_Circum	驚嘆號	斷槓
SunFA_Circum	減號	長音符號
SunFA_Circum	底線	長音符號
SunFA_Circum	句號	中點
SunFA_Circum	斜線	垂直線
SunFA_Circum	A	A 加抑揚音符
SunFA_Circum	a	a 加抑揚音符
SunFA_Circum	C	C 加抑揚音符
SunFA_Circum	c	c 加抑揚音符
SunFA_Circum	E	E 加抑揚音符
SunFA_Circum	e	e 加抑揚音符
SunFA_Circum	G	G 加抑揚音符
SunFA_Circum	g	g 加抑揚音符

表 5-8 以重音固定鍵為基礎的 Compose 鍵序列 (續)

按住後放開	按住後放開	結果
SunFA_Circum	H	H 加抑揚音符
SunFA_Circum	h	h 加抑揚音符
SunFA_Circum	I	I 加抑揚音符
SunFA_Circum	i	i 加抑揚音符
SunFA_Circum	J	J 加抑揚音符
SunFA_Circum	j	j 加抑揚音符
SunFA_Circum	O	O 加抑揚音符
SunFA_Circum	o	o 加抑揚音符
SunFA_Circum	S	S 加抑揚音符
SunFA_Circum	s	s 加抑揚音符
SunFA_Circum	U	U 加抑揚音符
SunFA_Circum	u	u 加抑揚音符
SunFA_Diaeresis	雙引號	分音符
SunFA_Diaeresis	空格鍵	分音符
SunFA_Diaeresis	A	A 加分音符
SunFA_Diaeresis	a	a 加分音符
SunFA_Diaeresis	E	E 加分音符
SunFA_Diaeresis	e	e 加分音符
SunFA_Diaeresis	I	I 加分音符
SunFA_Diaeresis	i	i 加分音符
SunFA_Diaeresis	O	O 加分音符
SunFA_Diaeresis	o	o 加分音符
SunFA_Diaeresis	U	U 加分音符
SunFA_Diaeresis	u	u 加分音符
SunFA_Diaeresis	y	y 加分音符
SunFA_Diaeresis	Y	Y 加分音符
SunFA_Tilde	空格鍵	波浪號
SunFA_Tilde	A	A 加波浪號
SunFA_Tilde	a	a 加波浪號

表 5-8 以重音固定鍵為基礎的 Compose 鍵序列 (續)

按住後放開	按住後放開	結果
SunFA_Tilde	N	N 加波浪號
SunFA_Tilde	n	n 加波浪號
SunFA_Tilde	O	O 加波浪號
SunFA_Tilde	o	o 加波浪號

阿拉伯文輸入模式

若要切換到阿拉伯文輸入模式，請按 **Compose a r** 或從輸入模式選擇視窗中選取阿拉伯文輸入模式。如需使用輸入模式選擇視窗的詳細資訊，請參閱第 111 頁的「使用輸入模式」。

下圖顯示了阿拉伯文鍵盤的配置。

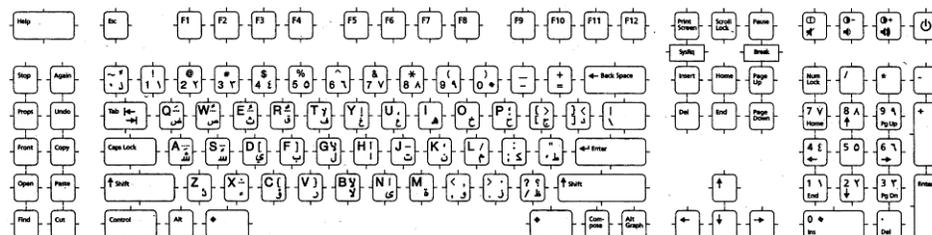


圖 5-2 阿拉伯文鍵盤

斯拉夫文輸入模式

若要切換到斯拉夫文輸入模式，請按 **Compose c c** 或從輸入模式選擇視窗中選取斯拉夫文輸入模式。如需使用輸入模式選擇視窗的詳細資訊，請參閱第 111 頁的「使用輸入模式」。

斯拉夫文 (俄文) 鍵盤配置如下圖所示。

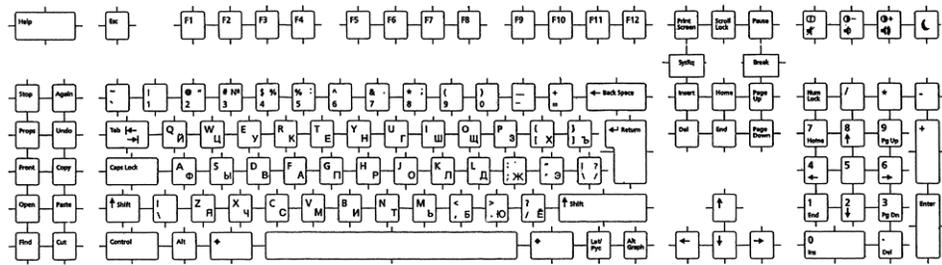


圖 5-3 斯拉夫文 (俄文) 鍵盤

當您切換至斯拉夫文輸入模式之後，便無法輸入英文或歐洲文字。若要切換到英文/歐洲輸入模式，請同時按 **Control**—**空格鍵**或從狀態區的「輸入模式選擇視窗」中選取英文/歐洲輸入模式。請參閱第 111 頁的「使用輸入模式」。

您也可以輸入相關輸入模式切換按鍵序列，切換至其他輸入模式。

希臘文輸入模式

若要切換到希臘文輸入模式，請按 **Compose g g** 或從輸入模式選擇視窗中選取希臘文輸入模式。如需使用輸入模式選擇視窗的詳細資訊，請參閱第 111 頁的「使用輸入模式」。

當您切換至希臘文輸入模式之後，便無法輸入英文或歐洲文字。若要切換到英文/歐洲輸入模式，請同時按 **Control** 和 **空格鍵**或從狀態區的輸入模式選擇視窗中選取英文/歐洲輸入模式。下圖顯示了希臘文 Euro 鍵盤的配置。

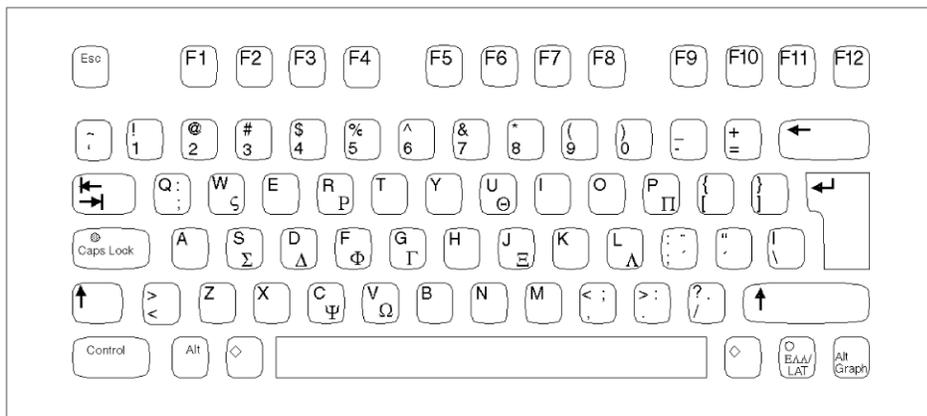


圖 5-4 希臘文 Euro 鍵盤

下圖顯示希臘文 UNIX 鍵盤。

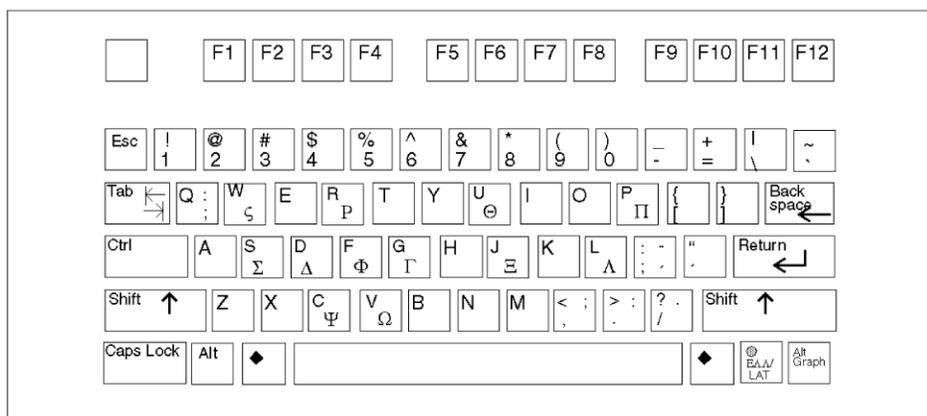


圖 5-5 希臘文 UNIX 鍵盤

下圖顯示希臘文輸入模式支援的 Compose 鍵序列。部分 Compose 鍵序列是以重音固定鍵為起始鍵。「ordfemenine」是陰性序數指示鍵的縮寫。

表 5-9 希臘文輸入模式的 Compose 鍵序列

按住後放開	按住後放開	結果
分號	a	小寫 Greek_alpha 加 tonos
分號	e	小寫 Greek_epsilon 加 tonos

表 5-9 希臘文輸入模式的 Compose 鍵序列 (續)

按住後放開	按住後放開	結果
分號	h	小寫 Greek_eta 加 tonos
分號	i	小寫 Greek_iota 加 tonos
分號	o	小寫 Greek_omicron 加 tonos
分號	y	小寫 Greek_upsilon 加 tonos
分號	v	小寫 Greek_omega 加 tonos
分號	A	大寫 Greek_alpha 加 tonos
分號	E	大寫 Greek_epsilon 加 tonos
分號	H	大寫 Greek_eta 加 tonos
分號	I	大寫 Greek_iota 加 tonos
分號	O	大寫 Greek_omicron 加 tonos
分號	Y	大寫 Greek_upsilon 加 tonos
分號	V	大寫 Greek_omega 加 tonos
dead_acute	Greek_alpha	小寫 Greek_alpha 加 tonos
dead_acute	Greek_epsilon	小寫 Greek_epsilon 加 tonos
dead_acute	Greek_eta	小寫 Greek_eta 加 tonos
dead_acute	Greek_iota	小寫 Greek_iota 加 tonos
dead_acute	Greek_omicron	小寫 Greek_omicron 加 tonos
dead_acute	Greek_upsilon	小寫 Greek_upsilon 加 tonos
dead_acute	Greek_omega	小寫 Greek_omega 加 tonos
dead_acute	Greek_ALPHA	大寫 Greek_alpha 加 tonos
dead_acute	Greek_EPSILON	大寫 Greek_epsilon 加 tonos
dead_acute	Greek_ETA	大寫 Greek_eta 加 tonos
dead_acute	Greek_IOTA	大寫 Greek_iota 加 tonos
dead_acute	Greek_OMICRON	大寫 Greek_omicron 加 tonos
dead_acute	Greek_UPSILON	大寫 Greek_upsilon 加 tonos
dead_acute	Greek_OMEGA	大寫 Greek_omega 加 tonos
dead_acute	a	小寫 Greek_alpha 加 tonos
dead_acute	e	小寫 Greek_epsilon 加 tonos
dead_acute	h	小寫 Greek_eta 加 tonos

表 5-9 希臘文輸入模式的 Compose 鍵序列 (續)

按住後放開	按住後放開	結果
dead_acute	i	小寫 Greek_iota 加 tonos
dead_acute	o	小寫 Greek_omicron 加 tonos
dead_acute	y	小寫 Greek_upsilon 加 tonos
dead_acute	v	小寫 Greek_omega 加 tonos
dead_acute	A	大寫 Greek_alpha 加 tonos
dead_acute	E	大寫 Greek_epsilon 加 tonos
dead_acute	H	大寫 Greek_eta 加 tonos
dead_acute	I	大寫 Greek_iota 加 tonos
dead_acute	O	大寫 Greek_omicron 加 tonos
dead_acute	Y	大寫 Greek_upsilon 加 tonos
dead_acute	V	大寫 Greek_omega 加 tonos
冒號	i	小寫 Greek_iota 加 dialytika
冒號	y	小寫 Greek_upsilon 加 dialytika
冒號	I	大寫 Greek_iota 加 dialytika
冒號	Y	大寫 Greek_upsilon 加 dialytika
dead_diaeresis	i	小寫 Greek_iota 加 dialytika
dead_diaeresis	y	小寫 Greek_upsilon 加 dialytika
dead_diaeresis	I	大寫 Greek_iota 加 dialytika
dead_diaeresis	Y	大寫 Greek_upsilon 加 dialytika
dead_diaeresis	Greek_iota	小寫 Greek_iota 加 dialytika
dead_diaeresis	Greek_upsilon	小寫 Greek_upsilon 加 dialytika
dead_diaeresis	Greek_IOTA	大寫 Greek_iota 加 dialytika
dead_diaeresis	Greek_UPSILON	大寫 Greek_upsilon 加 dialytika
分號	分號	希臘文 tonos
冒號	冒號	分音符/dialytika

表 5-9 希臘文輸入模式的 Compose 鍵序列 (續)

按住後放開	按住後放開	結果
ordfeminine	0	加減符號
ordfeminine	1	節符號 (§)
ordfeminine	2	上標 2
ordfeminine	3	上標 3
ordfeminine	5	斷槓
ordfeminine	6	版權符號
ordfeminine	7	Not 符號
ordfeminine	8	連字號
ordfeminine	9	度數符號
ordfeminine	連字號	普通分數二分之一
ordfeminine	反斜線	英鎊符號
ordfeminine	左大括號	修飾元字母反轉逗號
ordfeminine	右大括號	修飾元字母撇號
ordfeminine	左括號	左向雙箭頭引號
ordfeminine	右括號	右向雙箭頭引號
SunFA_Acute	a	小寫 Greek_alpha 加 tonos
SunFA_Acute	e	小寫 Greek_epsilon 加 tonos
SunFA_Acute	h	小寫 Greek_eta 加 tonos
SunFA_Acute	i	小寫 Greek_iota 加 tonos
SunFA_Acute	o	小寫 Greek_omicron 加 tonos
SunFA_Acute	y	小寫 Greek_upsilon 加 tonos
SunFA_Acute	v	Greek_omega 加 tonos
SunFA_Acute	A	大寫 Greek_alpha 加 tonos
SunFA_Acute	E	大寫 Greek_epsilon 加 tonos
SunFA_Acute	H	大寫 Greek_eta 加 tonos
SunFA_Acute	O	大寫 Greek_omicron 加 tonos
SunFA_Acute	I	大寫 Greek_iota 加 tonos
SunFA_Acute	Y	大寫 Greek_upsilon 加 tonos
SunFA_Acute	V	大寫 Greek_omega 加 tonos

表 5-9 希臘文輸入模式的 Compose 鍵序列 (續)

按住後放開	按住後放開	結果
SunFA_Acute	Greek_alpha	小寫 Greek_alpha 加 tonos
SunFA_Acute	Greek_epsilon	小寫 Greek_epsilon 加 tonos
SunFA_Acute	Greek_eta	小寫 Greek_eta 加 tonos
SunFA_Acute	Greek_iota	小寫 Greek_iota 加 tonos
SunFA_Acute	Greek_omega	小寫 Greek_omega 加 tonos
SunFA_Acute	Greek_omicron	小寫 Greek_omicron 加 tonos
SunFA_Acute	Greek_upsilon	小寫 Greek_upsilon 加 tonos
SunFA_Acute	Greek_ALPHA	大寫 Greek_alpha 加 tonos
SunFA_Acute	Greek_EPSILON	大寫 Greek_epsilon 加 tonos
SunFA_Acute	Greek_ETA	大寫 Greek_eta 加 tonos
SunFA_Acute	Greek_IOTA	大寫 Greek_iota 加 tonos
SunFA_Acute	Greek_OMICRON	大寫 Greek_omicron 加 tonos
SunFA_Acute	Greek_UPSILON	大寫 Greek_upsilon 加 tonos
SunFA_Acute	Greek_OMEGA	大寫 Greek_omega 加 tonos
SunFA_Diaeresis	i	小寫 Greek_iota 加 dialytika
SunFA_Diaeresis	y	小寫 Greek_upsilon 加 dialytika
SunFA_Diaeresis	I	大寫 Greek_iota 加 dialytika
SunFA_Diaeresis	Y	大寫 Greek_upsilon 加 dialytika
SunFA_Diaeresis	Greek_iota	小寫 Greek_iota 加 dialytika
SunFA_Diaeresis	Greek_upsilon	小寫 Greek_upsilon 加 dialytika
SunFA_Diaeresis	Greek_IOTA	大寫 Greek_iota 加 dialytika
SunFA_Diaeresis	Greek_UPSILON	大寫 Greek_upsilon 加 dialytika

表 5-10 希臘文輸入模式中的三鍵 Compose 鍵序列

按住後放開	按住後放開	按住後放開	結果
分號	冒號	y	小寫 Greek_upsilon 加 dialytika 和 tonos

表 5-10 希臘文輸入模式中的三鍵 Compose 鍵序列 (續)

按住後放開	按住後放開	按住後放開	結果
冒號	分號	y	小寫 Greek_upsilon 加 dialytika 和 tonos
分號	冒號	i	小寫 Greek_iota 加 dialytika 和 tonos
冒號	分號	i	小寫 Greek_iota 加 dialytika 和 tonos
dead_acute	dead_diaeresis	y	小寫 Greek_upsilon 加 dialytika 和 tonos
dead_diaeresis	dead_acute	y	小寫 Greek_upsilon 加 dialytika 和 tonos
dead_acute	dead_diaeresis	i	小寫 Greek_iota 加 dialytika 和 tonos
dead_diaeresis	dead_acute	i	小寫 Greek_iota 加 dialytika 和 tonos
dead_acute	dead_diaeresis	Greek_upsilon	小寫 Greek_upsilon 加 dialytika 和 tonos
dead_diaeresis	dead_acute	Greek_upsilon	小寫 Greek_upsilon 加 dialytika 和 tonos
dead_acute	dead_diaeresis	Greek_iota	小寫 Greek_iota 加 dialytika 和 tonos
dead_diaeresis	dead_acute	Greek_iota	小寫 Greek_iota 加 dialytika 和 tonos
SunFA_Acute	SunFA_Diaeresis	i	小寫 Greek_iota 加 dialytika 和 tonos
SunFA_Diaeresis	SunFA_Acute	i	小寫 Greek_iota 加 dialytika 和 tonos
SunFA_Acute	SunFA_Diaeresis	y	小寫 Greek_upsilon 加 dialytika 和 tonos
SunFA_Diaeresis	SunFA_Acute	y	小寫 Greek_upsilon 加 dialytika 和 tonos
SunFA_Acute	SunFA_Diaeresis	Greek_iota	小寫 Greek_iota 加 dialytika 和 tonos
SunFA_Diaeresis	SunFA_Acute	Greek_iota	小寫 Greek_iota 加 dialytika 和 tonos
SunFA_Acute	SunFA_Diaeresis	Greek_upsilon	小寫 Greek_upsilon 加 dialytika 和 tonos

表 5-10 希臘文輸入模式中的三鍵 Compose 鍵序列 (續)

按住後放開	按住後放開	按住後放開	結果
SunFA_Diaeresis	SunFA_Acute	Greek_upsilon	小寫 Greek_upsilon 加 dialytika 和 tonos

表 5-11 希臘文輸入模式中的四鍵 Compose 鍵序列

按住後放開	按住後放開	按住後放開	按住後放開	結果
分號	冒號	分號	冒號	希臘文 dialytika tonos
冒號	分號	冒號	分號	希臘文 dialytika tonos

希伯來文輸入模式

若要切換到希伯來文輸入模式，請按 **Compose h h** 或從輸入模式選擇視窗中選取希伯來文輸入模式。如需使用輸入模式選擇視窗的詳細資訊，請參閱第 111 頁的「使用輸入模式」。

下圖顯示了希伯來文鍵盤的配置。

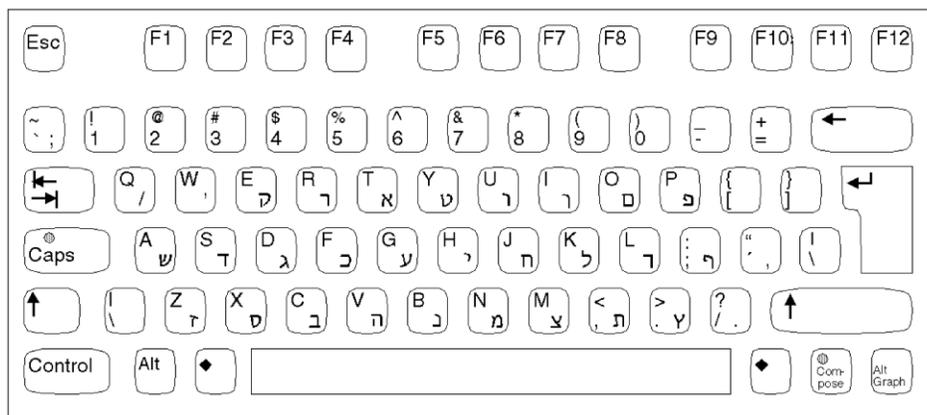


圖 5-6 希伯來文鍵盤

若要使用當地的繁體中文 (香港) 輸入系統，您必須在系統上安裝一或多個繁體中文 (香港) 語言環境。

Unicode 十六進位輸入模式

若要切換到 Unicode 十六進位輸入模式，請按 **Compose u h** 或從輸入模式選擇視窗中選取 Unicode 十六進位輸入模式。若要切換到八進位數字系統，請按 **Compose u o** 或選取 Unicode 八進位輸入模式。如需使用輸入模式選擇視窗的詳細資訊，請參閱第 111 頁的「使用輸入模式」。

若要使用這些輸入模式，您必須知道字元的十六進位或八進位字碼點值。請參閱「*The Unicode Standard, Version 4.0*」，以取得字碼點值與字元之間的對映。

如果您正在 Unicode 十六進位碼輸入模式中，則若您要輸入一個字元就要輸入四個十六進位數字。下面是一些示範的十六進位值：

- 00A1 為倒置的驚嘆號
- 03B2 為希臘文小字母的第二個
- AC00 為韓文的 Hangul 音節
- 30A1 為日文片假名字母 A
- 4E58 為統一漢字字元

您可以在十六進位數字中使用大寫和小寫的 A、B、C、D、E 和 F。如果您偏好使用八進位數字系統，而不願使用十六進位數字系統，可以輸入八進位數字 0 到 7。若輸入了錯誤的數字，可以使用 **Delete** 鍵或 **Backspace** 鍵來刪除。

查表輸入模式

若要切換到查表輸入模式，請按 **Compose l l** 或從輸入模式選擇視窗中選取查表輸入模式。如需使用輸入模式選擇視窗的詳細資訊，請參閱第 111 頁的「使用輸入模式」。

第二個查表視窗顯示以群組方式顯示的候選字，每次最多可顯示 80 個候選字。請按 **Control n** 顯示下一組候選字或按 **Control p** 顯示上一組候選字。

系統環境

本節將說明語言環境變數、TTY 環境設定、32 位元和 64 位元 STREAMS 模組以及終端機支援。

語言環境變數

請確認您的系統上已安裝 en_US.UTF-8 語言環境。若要檢查目前語言環境的各種設定，請使用 locale 公用程式。

```
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=
```

若要使用 en_US.UTF-8 語言環境的桌面環境，請先選擇該語言環境。請先在一個 TTY 環境中，設定 LANG 環境變數為 en_US.UTF-8 來選擇語言環境，請參考下列 C-shell 範例：

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

請確定您未設定 LC_ALL、LC_COLLATE、LC_CTYPE、LC_MESSAGES、LC_NUMERIC、LC_MONETARY 和 LC_TIME 種類或是已設定為 en_US.UTF-8。這些種類一經設定，其將會置換較低優先權的 LANG 環境變數。如需環境變數架構的詳細資訊，請參閱「setlocale(3C) 線上說明手冊」。

您也可以從 CDE 桌面啟動 en_US.UTF-8 環境。在 CDE 登入螢幕的 Options -> Language 功能表中選擇 en_US.UTF-8。

TTY 環境設定

根據您所使用的終端機或終端機模擬器，您可能需要在系統上安裝某些字碼集特定的 STREAMS 模組。

如需 STREAMS 模組和一般使用串流的詳細資訊，請參閱「*STREAMS Programming Guide*」。

下表列出了在終端機環境中，en_US.UTF-8 語言環境所支援的 64 位元 STREAMS 模組。如需詳細資訊，請參閱「*Solaris 64-bit Developer's Guide*」。

表 5-12 en_US.UTF-8 所支援的 STREAMS 模組

32 位元 STREAMS 模組	描述
/usr/kernel/strmod/sparcv9/u8lat1	UTF-8 和 ISO8859-1 之間的字碼轉換 STREAMS 模組 (西歐語系)
/usr/kernel/strmod/sparcv9/u8lat2	UTF-8 和 ISO8859-2 之間的字碼轉換 STREAMS 模組 (東歐)
/usr/kernel/strmod/sparcv9/u8koi8	UTF-8 和 KOI8-R (斯拉夫文) 之間的字碼轉換 STREAMS 模組

注意 – 從 Solaris 10 發行版本開始，不再支援 32 位元核心的 SPARC sun4u 平台。表 5-12 僅適用於 32-位元核心的 x86 平台。如需詳細資訊，請參閱「版本說明」。

下表列出了 en_US.UTF-8 所支援的 64 位元 STREAMS 模組。

表 5-13 en_US.UTF-8 所支援的 64 位元 STREAMS 模組

64 位元 STREAMS 模組	描述
/usr/kernel/strmod/sparcv9/u8lat1	UTF-8 和 ISO8859-1 之間的字碼轉換 STREAMS 模組 (西歐)
/usr/kernel/strmod/sparcv9/u8lat2	UTF-8 和 ISO8859-2 之間的字碼轉換 STREAMS 模組 (東歐)
/usr/kernel/strmod/sparcv9/u8koi8	UTF-8 和 KOI8-R (斯拉夫文) 之間的字碼轉換 STREAMS 模組

▼ 如何載入 STREAMS 核心模組

1. 以 root 使用者身份，確認您執行的是 64 位元 Solaris 或 32 元 Solaris 系統。

```
system# isainfo -v
```

- 64 位元 Solaris 系統會傳回下列資訊：

```
64 位元 sparcv9 應用程式
32 位元 sparc 應用程式
```

- 32 位元 Solaris 系統會傳回下列資訊：

```
32 位元 sparc 應用程式
```

- 32 位元 x86 系統會傳回下列資訊：

```
32 位元 i386 應用程式
```

2. 確認您的系統是否已載入 STREAMS 模組。

```
system# modinfo | grep 模組名稱
```

若您已安裝如 `u8lat1` 等 STREAMS 模組，其輸出將會顯示如下：

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

3. 如果該模組尚未載入，請使用 `modload(1M)` 指令來載入它。

- 在 32 位元系統上，請鍵入：

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

- 在 64 位元系統上，請鍵入：

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

核心已載入適當的 `u8lat1` STREAMS 模組。現在可以將它推入到串流。

▼ 如何卸載 STREAMS 核心模組

1. 以 `root` 使用者身份，確認已載入核心模組。

例如，若要確認已載入 `u8lat1`，請鍵入：

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

2. 請使用 `modunload(1M)` 指令來卸載核心。

例如，若要卸載 `u8lat1` 模組，請鍵入：

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

▼ 如何設定 Latin-2 終端機和 STREAMS 模組

1. 請使用 `strchg(1M)`，如第二個指令行所示。

```
system% cat > tmp/mystreams
ttcompat
ldterm
u8lat1
ptem
^D
system% strchg -f /tmp/mystreams
```

在使用 `strchg(1)` 之前，請確認您是 `root` 使用者或裝置擁有人。

2. 執行 `strconf` 指令以測試目前的配置。

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

3. 執行 `strchg` 指令以重新設定原始配置。

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

dtterm、xterm 和可輸入/輸出 UTF-8 字元的終端機

不同於舊版 Solaris 作業系統，支援 UTF-8 字碼集輸入和輸出的 `dtterm` 和 `xterm` 終端機模擬器及其他終端機，在其串流中不需要任何額外的 STREAMS 模組。如果使用 `stty(1)` 公用程式來設定終端機環境，則 `ldterm` 模組將與字碼集無關，並且支援 Unicode/UTF-8。

若要為 Unicode 語言環境設定適當的終端機環境，請使用 `stty(1)` 公用程式。

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

若要查詢目前的設定，請使用 `stty` 公用程式的 `-a` 選項，如下所示：

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

注意 – 由於 `/usr/ucb/stty` 尚未國際化，請使用 `/bin/stty` 來代替。

Latin-1、Latin-2 或 KOI8-R 的終端機支援

針對僅支援 Latin-1 (ISO8859-1)、Latin-2 (ISO8859-2) 或 KOI8-R 的終端機，您必須進行以下的 STREAMS 配置：

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

此配置僅適用於支援 Latin-1 的終端機。針對 Latin-2 終端機，請將 STREAMS 模組 `u8lat1` 替換為 `u8lat2`。針對 KOI8-R 終端機，請將模組替換為 `u8koi8`。

請確定您已經在核心程式中載入該 STREAMS 模組。

儲存 `~/.cshrc` 中的設定

假設必要的 STREAMS 模組已與核心一起載入，您可以直接在 `.cshrc` 檔案 (C shell 範例) 中儲存下列指令行：

```
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
```

一旦您將這些指令行儲存至 `.cshrc` 檔案中，每次當您使用 STREAMS 模組時就不再需要輸入所有的指令。請注意，第二 `_EOF` 的起始位置必須位於檔案的第一個欄位。

字碼轉換

Unicode 語言環境透過 `iconv` 和 `sdtconvtool` 公用程式，支援在多個國家的主要字碼集之中新增多種字碼轉換。

`geniconvtbl` 公用程式能在目前 Solaris 環境中啓用使用者自行定義的字碼轉換。您可以藉由 `iconv(1)` 和 `iconv(3)` 來使用由 `geniconvtbl` 公用程式所建立的使用者自行定義字碼轉換。如需此公用程式的詳細資訊，請參閱「`geniconvtbl(1)` 線上說明手冊」和「`geniconvtbl(4)` 線上說明手冊」。

可套用至 `iconv`、`iconv_open` 和 `sdtconvtool` 的可用 `fromcode` 和 `toctype` 名稱，列示在附錄 A 中的表格。如需 `iconv` 字碼轉換的詳細資訊，請參閱「`iconv(1)` 線上說明手冊」和「`sdtconvtool(1)` 線上說明手冊」。如需可用字碼轉換的詳細資訊，請參閱「`iconv(5)` 線上說明手冊」。另請參閱附錄 A。

注意 – UCS-2、UCS-4、UTF-16 和 UTF-32 為所有 Unicode/ISO/IEC 10646 表示法格式，它們認可 Unicode 4.0 和 ISO/IEC 10646-1:2000 標準中定義的位元組次序記號 (BOM) 字元 (如果字元出現於字元串流的開頭)。其他像是 UCS-2BE、UCS-4BE、UTF-16BE 和 UTF-32BE 為所有固定寬度的 Unicode/ISO/IEC 10646 表示法格式，它們並不會認可 BOM 字元，且會假定大尾數法的位元組次序。另一方面，UCS-2LE、UCS-4LE、UTF-16LE 和 UTF-32LE 則會假定小尾數法的位元組次序。這些格式也不會認可 BOM 字元。

如需 ISO8859-* 和 KO18-* 相關字體及語言的資訊，請參閱 <http://czyborra.com/charsets/iso8869.html>。

DtMail 支援

由於字體範圍擴大了，執行於 en_US.UTF-8 語言環境的 Solaris DtMail 支援下列 MIME 名稱的字元集：

- US-ASCII (7 位元 US ASCII)
- UTF-8 (UCS 轉換格式 8 位元)
- UTF-7 (UCS 轉換格式 7 位元)
- ISO-8859-1 (Latin-1)
- ISO-8859-2 (Latin-2)
- ISO-8859-3 (Latin-3)
- ISO-8859-4 (Latin-4)
- ISO-8859-5 (Latin/斯拉夫文)
- ISO-8859-6 (Latin/阿拉伯文)
- ISO-8859-7 (Latin/希臘文)
- ISO-8859-8 (Latin/希伯來文)
- ISO-8859-9 (Latin-5)
- ISO-8859-10 (Latin-6)
- ISO-8859-13 (Latin-7/波羅的海文)
- ISO-8859-14 (Latin-8/塞爾特文)
- ISO-8859-15 (Latin-9)
- ISO-8859-16 (Latin-10)
- KOI8-R (斯拉夫文)
- ISO-2022-JP and EUC-JP (日文)
- ISO-2022-KR and EUC-KR (韓文)
- ISO-2022-CN (簡體中文)
- ISO-8859-13 (Latin-7/波羅的海文)
- ISO-8859-14 (Latin-8/Celtic)
- KOI8-U (Cyrillic/烏克蘭文)
- Shift_JIS (Shift JIS 中的日文)
- GB2312 (EUC 中的簡體中文)
- TIS-620 (泰文)
- UTF-16 (UCS 轉換格式 16 位元)
- UTF-16BE (UTF-16 大尾數法)
- UTF-16LE (UTF-16 小尾數法)
- Windows-1250
- Windows-1251
- Windows-1252
- Windows-1253
- Windows-1254
- Windows-1255
- Windows-1256
- Windows-1257
- Windows-1258
- Big5 (繁體中文)
- UTF-32 (UCS 轉換格式 32 位元)

- UTF-32BE (UTF-32 大尾數法)
- UTF-32LE (UTF-32 小尾數法)

此支援功能可讓使用者在 Dtmail 單一實例中，檢視以世界上任何區域的字元集所編碼的各種電子郵件。DtMail 可藉由檢視電子郵件所提供的 MIME 字元集和內容傳輸編碼來為接收到的電子郵件解碼。提供支援 Windows-125x MIME 字元集的功能。

如果要傳送郵件，您就必須指定收信人郵件使用者代理 (郵件用戶端) 可讀取的 MIME 字碼集或是使用 en_US.UTF-8 語言環境所提供的預設 MIME 字碼集。您可以按 **Control Y** 或按一下「格式」功能表按鈕，然後按一下「變更字元集」按鈕，以在「新訊息」視窗中切換外送郵件的字元集。位於「傳送」按鈕頂端的下一個可用的字元集名稱將顯示在左下角。

若電子郵件訊息標頭或訊息內文所包含的字元無法由指定的 MIME 字元集來表示，系統會自動將字元集切換至能表示任何字元的 UTF-8。

若訊息只包含了 7 位元 US-ASCII 字元集中的字元，您電子郵件的預設 MIME 字元集為 US-ASCII。任何郵件使用者代理者都可解譯這些電子郵件訊息，而不會造成字元或資訊的漏失。

若訊息中包含混合字體的字元，則預設的 MIME 字元集為 UTF-8。所有 UTF-8 的 8 位元字元都是以可列印參閱來進行編碼。如需 MIME、已註冊的 MIME 字元以及可列印參閱編碼的詳細資訊，請參閱 RFCs 2045、2046、2047、2048、2049、2279、2152、2237、1922、1557、1555 和 1489。

使用 X 應用程式的 FontSet

如需使用 X 應用程式之 FontSet 的資訊，請參閱第 108 頁的「Unicode 語言環境：en_US.UTF-8 支援」。

Solaris 桌面環境中的每個字元集都有其相關聯的字型集。

下列是目前 Solaris 環境所支援的 Latin-1 字型清單：

```
-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
```

如需 CDE 常用字型別名，包括 `-dt-interface user-*` 和 `-dt-application-*` 別名的資訊，請參閱「*Common Desktop Environment: Internationalization Programmer's Guide*」。

在 `en_US.UTF-8` 語言環境中，`utf` 也包含在區域環境的一般字型別名之內，並為 X 編輯字型描述名稱樣式欄位中的附加屬性。因此，為了取得適當的字型集，請參閱下列範例，在字型集建立中加入其他的樣式：

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

CDE/Motif 應用程式中的 FontList 定義

和 FontSet 定義一樣，應用程式的 X`mFontList` 資源定義也應該包含由語言環境所支援的附加樣式屬性。

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

第 6 章

複合文字版面配置

複合文字版面配置 (CTL) 擴充，可以使 Motif API 支援邏輯和實體文字間所需的繁複轉換過程的寫入系統。阿拉伯文、希伯來文和泰文等語言需要這樣的轉換。CTL Motif 提供字元外形，例如連音符、分音符和音節順序。另外還提供靜態與動態文字視窗元件轉換的支援以及動態文字視窗元件的雙向文字能力和調整功能。因為文字的產生是透過解譯層來處理，因此可以輕易地擴充其他視窗元件程式庫來支援 CTL。

本章包含以下主題：

- 第 149 頁的「CTL 技術概述」
- 第 150 頁的「CTL 架構概述」
- 第 150 頁的「基於 X Library 的應用程式的 CTL 支援」
- 第 150 頁的「XOC 資源」
- 第 151 頁的「支援 CTL 技術的 Motif 變更」
- 第 160 頁的「開發 CTL 應用程式」

CTL 技術概述

若要使用這些新功能，使用者必須擁有「可攜式版面配置服務 (PLS)」程式庫及適當的語言引擎。CTL 使用 PLS 做為語言引擎的介面，並且在產生文字之前，使用此語言引擎來轉換文字。支援 CTL 的應用程式，必須包含在 CTL 文件中所述的其他資源。

特別是，XomCTL 支援下列由基礎語言環境附屬的 PLS 模組轉換所提供的複雜語言外形及重新排序功能：

- 位置變化
- 連字 (多變一) 及字元組合 (一變多)
- 變音符
- 雙向
- 對稱交換
- 數字外形

- 字串驗證

CTL 架構概述

此 CTL 架構已編製如圖 6-1 所示。Dt Apps 位於堆疊頂端，使用 Motif CTL 功能來產生文字。Motif 會使用 PLS 依次與特定語言環境語言引擎介面溝通，並執行轉換來支援位置變化、數字外形設計等功能。

CTL 架構使用特定語言環境引擎來支援新的語言。換句話說，您可以在不改變 Motif 或 Dt Apps 的情況下增加泰文及越南文支援。

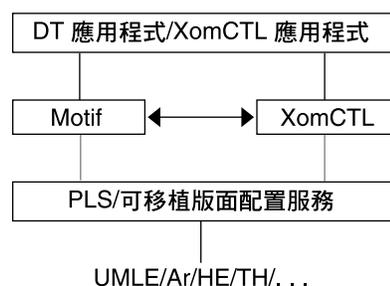


圖 6-1 CTL 架構

基於 X Library 的應用程式的 CTL 支援

XomCTL (X 程式庫輸出模組中的複合文字版面配置支援) 可使所有純 X Windows 應用程式 (例如 X 基礎的終端機模擬器) 具備 CTL 支援。XomCTL 提供包含了 X11 無字型支援的完整功能之 Open Source XI18N 實施。

XOC 資源

在目前 Solaris 環境中，提供下列 XOC 資源：

XNText

讓使用者能夠設定需要執行 CTL 作業的文字緩衝區

XNTextLayoutNumGlyphs	將影像編號提供給文字緩衝區中的文字
XNTextLayoutModifier	與 Motif 的 XmNLayoutModifier 相同
XNTextLayoutProperty	與 PLS 屬性、輸入至輸出以及輸出至輸入相同
XNTextLayoutMapInpToOut	與 PLS 屬性、輸入至輸出以及輸出至輸入相同
XNTextLayoutMapOutToInp	與 PLS 屬性、輸入至輸出以及輸出至輸入相同

請參閱 X/Open 或「PLS 可攜式版面配置服務」規格書，以取得這些資源的說明。

支援 CTL 技術的 Motif 變更

下列 Motif 的變更支援 CTL 技術：

XmNlayoutDirection	控制物件版面配置
XmStringDirection	指定系統顯示字串字元的方向
XmRendition	將新的虛擬資源增加至 XmRendition
XmText 和 XmTextField	影響與 XmRendition 關聯之文字的版面配置行為
XmTextFieldGetLayoutModifier	傳回產生版面配置物件的版面配置修飾元字串
XmTextGetLayoutModifier	傳回因視窗元件而產生的目前版面配置物件設定
XmTextFieldSetLayoutModifier	設定產生版面配置物件所必需的版面配置修飾元值
XmTextSetLayoutModifier	修改因視窗元件產生的版面配置物件設定
XmStringDirectionCreate	建立複合字串

XmNlayoutDirection 資源

XmNlayoutDirection 資源會控制物件版面配置。此資源以下述方式與 LayoutObject 的方向值交互影響。

如需 XmNlayoutDirection 的概述，請參閱 Motif 「*Programmer's Guide*」(2.1 版) 第 11.3 章節，其中特別說明了 XmStringDirection 與 XmNlayoutDirection 之間的交互關係。

決定版面配置方向

當 `XmNlayoutDirection` 指定為 `XmDEFAULT_DIRECTION` 時，視窗元件的版面配置方向會在建立時從管理虛擬 XOC 中設定。如果是動態文字 (`XmText` 和 `XmTextField`)，視窗元件會使用與 `XmRendition` 相關聯的管理虛擬 XOC。如果是靜態文字 (`XmList`、`XmLabel` 和 `XmLabelG`)，版面配置方向是從指定方向的第一個複合字串元件設定的。此種規格以兩種方式的其中一種出現：

- `XmSTRING_COMPONENT_LAYOUT_PUSH` 或 `XmSTRING_COMPONENT_DIRECTION` 類型的元件。
- `XmSTRING_COMPONENT_LOCALE_TEXT`、`XmSTRING_COMPONENT_WIDECHAR_TEXT` 或 `XmSTRING_COMPONENT_TEXT` 類型的元件，這些元件源自於相關聯的 `XmRendition` 和 `LayoutObject`。

如果 `XmNlayoutDirection` 未指定為 `XmDEFAULT_DIRECTION` 且 `XmNlayoutModifier @ls orientation` 值未在版面配置修飾元字串中明確指定，則 `XmNlayoutDirection` 值會傳遞到 XOC 及其 `LayoutObject`。

同時明確指定 `XmNlayoutDirection` 和 `XmNlayoutModifier @ls orientation` 值會混淆行爲。`XmNlayoutDirection` 控制視窗元件物件版面配置，而 `XmNlayoutModifier @ls orientation` 值則控制版面配置變換。

請參閱「*CAE Specification: Portable Layout Services: Context-dependent and Directional Text*」(The Open Group：Feb 1997；ISBN 1-85912-142-X；document number C616)，以取得具移植性功能的說明，具移植性功能是用來處理依賴於上下文且為雙向文字的轉換，以做為現存 POSIX 語言環境模型的邏輯擴充。此文件是專為想要提供複合文字語言支援的系統和應用程式設計者所設計的。

XmStringDirection 資源

`XmStringDirection` 是用來指定系統顯示字串字元之方向的資料類型。

針對任何不具有指定字串方向之元件的複合字串 (`XmString`)，`XmNlayoutDirection` 資源會設定預設產生方向。因此，如果要設定版面配置方向，您必須為 `XmNlayoutDirection` 資源設定適當的值。您不須建立特定方向元件的複合字串。當應用程式產生了 `XmString`，應用程式應該檢查字串建立時是否有明確方向 (`XmStringDirection`)。如果字串沒有提供方向元件，應用程式應該檢查目前視窗元件資源的 `XmNlayoutDirection` 值，並將該值視為 `XmString` 的預設產生方向。

XmRendition 資源

CTL 將下表所列新的虛擬資源增加到 `XmRendition`。虛擬資源的說明將接續在表格之後。

表 6-1 XmRendition 中的新資源

姓名	類別/類型	存取	預設值
XmNfontType	XmCFontType/XmFontType	CSG	XmAS_IS
XmNlayoutAttrObject	XmClayoutAttrObject/String	CG	NULL
XmNlayoutModifier	XmClayoutModifier/String	CSG	NULL

XmNfontType

指定產生的字型物件類型。若是 CTL，資源值必須是 XmFONT_IS_XOC 值。如果此值不相符，便會忽略 XmNlayoutAttrObject 和 XmNlayoutModifier 資源。

如果此資源的值是 XmFont_IS_XOC 且 XmNfont 資源未指定，那麼在建立時，XmNfontName 資源的值會轉換到由 XmNlayoutAttrObject 資源指定之語言環境或目前語言環境中的 XOC 物件。此外，XmNlayoutModifier 資源的值會傳遞到任何與 XOC 相關聯的版面配置物件。

XmNlayoutAttrObject

指定版面配置 AttrObject 引數。此資源是用來建立與此 XmRendition 相關聯之 XOC 相關聯的版面配置物件。請參考版面配置服務 m_create_layout() 規格，以瞭解此字串的語法和語義。如需版面配置修飾元方向輸出值和 XmNlayoutDirection 視窗元件資源之間交互影響的說明，請參閱 XmNfontType 的說明。

XmNlayoutModifier

指定要傳遞到 XOC 的 XmRendition 所使用之版面配置物件的版面配置值。如需此字串的語法和語義，請參閱「CAESpecification」。

使用 XmRendition{Retrieve,Update} 來設定此資源，會導致字串被傳遞到與此產生相關聯之 XOC 相關聯的版面配置物件。您可以使用此機制來動態地配置版面配置服務。如果變更了 Orientation、Context、TypeOfText、TextShaping 或 ShapeCharset，可能會導致意外的行為。

其他版面配置行為

XmNlayoutModifier 會影響與 XmRendition 相關聯之文字的版面配置行為。例如，如果數字的版面配置預設處理是 NUMERALS_NOMINAL，但您藉由將 XmNlayoutModifier 設定為 @ls numerals=nominal:national 或 @ls numerals=:national 來變更為 NUMERALS_NATIONAL。

版面配置值可分類成下列群組：

- 編碼描述 - TypeOfText、TextShaping、ShapeCharset (以及語言環境字碼集)
TypeOfText 基本上是呈區段排列的，且以不透明區塊展示。透過產生物件動態地修改這些值通常是沒有意義的，且肯定會造成無法預期的後果。
- 版面配置行為 - Orientation、Context、ImplicitAlg、Swapping 以及 Numerals。Orientation 和 Context 不可以動態方式修改。您可以安全地修改 ImplicitAlg、Swapping 和 Numerals。

- 編輯行為 - CheckMode。

XmText 和 XmTextField 資源

Xm CTL 在 Motif 2.0 CStext 視窗元件之後擴充了 XmText 和 XmTextField，其方法是增加可經由視覺化圖案來操作之移動和刪除動作的平行集。標準 Motif 2.1 Text 和 TextField 的邏輯順序和實際順序並無不同：*next* 和 *forward* 表示“到右邊”，*previous* 和 *backward* 表示“到左邊”。然而，CStext 會做出適當區別，並以完全實體名稱 (例如，*left-character()*、*delete-right-word()* 等) 來定義新的動作集。這些動作的常式都被定義成可對視窗元件的 *XmNlayoutDirection* 產生回應，且會呼叫正確的 *next-* 或 *previous-* 動作。

Xm CTL 延伸較 CStext 延伸略為複雜。Xm CTL 延伸不會對視窗元件的整體方向有所回應，但是會對虛擬 XOC (包括中性穩定) 所指定游標周圍的實際字元特定方向產生回應。

新的資源名稱可讓您控制選取策略、提供產生標籤和控制對齊。

新的 Xm CTL 動作集大致上是 {Character, Word}、{Left, Right} 和 {Move, Delete, Kill} 的交叉作用產物。下表將列出動作集。

表 6-2 Xm CTL 中的新資源

名稱	類別/類型	存取	預設值
XmNrenditionTag	XmCRenditionTag/XmRString	CSG	XmFONTLIST_DEFAULT_TAG
XmNalignment	XmCAalignment/XmRAalignment	CSG	XmALIGNMENT_BEGINNING
XmNeditPolicy	XmCEditPolicy/XmREditPolicy	CSG	XmEDIT_LOGICAL

XmNrenditionTag

指定 XmNrenderTable 資源中 XmRendition 的產生標記供視窗元件使用。

XmNalignment

指定視窗元件中使用的文字對齊方向。僅支援 XmALIGNMENT_END 和 XmALIGNMENT_CENTER。

XmNeditPolicy

指定供視窗元件使用的編輯策略，XmEDIT_LOGICAL 或 XmEDIT_VISUAL。在 XmEDIT_VISUAL 的情況下，選取、游標移動和刪除都是以視覺化的樣式呈現。設定此資源也會將一般的鍵盤移動和刪除事件解譯成新的「視覺」動作清單或現有的邏輯動作。

字元方向動作常式

forward-cell() 和 *backward-cell()* 動作查詢指定方向的字元方向。如果方向是從左至右，這些動作會呼叫對應的 *next-/forward-* 或 *previous-/ backward-* 變數。

字元方向的其他行爲

動作會使用巢狀層的「版面配置服務」變換 `OutToInp` 和 `Property` 緩衝區來決定字元方向。因此視窗元件的行爲取決於每個語言環境的特定轉換。如果 `OutToInp` 中的資訊，特別是 `Property` 緩衝區中的資訊不正確，視窗元件可能不會正常運作。尤其當特定語言環境的模組落在此規格之外的範圍時，同樣文字、應用程式、資源值和 `LayoutObject` 配置的雙向編輯行爲在每個平台上會有所不同。

視覺模式動作會導致以顯示格爲基礎的行爲顯示出來。邏輯模式動作會導致以邏輯字元爲基礎的行爲。例如，`delete-right-character()` 作業會刪除與顯示顯示格對應的輸入緩衝區字元。也就是說，輸入緩衝區字元的整個 `LayoutObject` 變換“property”位元組「new cell indicator」是 1，而所有隨後字元的「new cell indicator」是 0。

如需 `Property` 緩衝區的詳細資訊，請參閱「*CAE Specification*」中的 `m_transform_layout()` 規格。

同樣的，就 `backward-character()` 而言，插入點會在輸入緩衝區中往後移動一個字元，而游標會重現在對應相關輸出緩衝區字元的可視位置。因此，有好幾個按鍵必須移過複合顯示格。當插入點移過像是分音符或連音符片段 (其「new cell indicator」是 0) 的輸入緩衝區字元時，實際上並未變更游標的顯示位置。

此行爲表示刪除動作會從邏輯/輸入緩衝區端執行或是從實體/輸出緩衝區端的顯示格層執行。嚴格、實際的字元對應刪除並沒有任何模式，因爲輸入緩衝區和輸出緩衝區之間並沒有一對一的對應關係。例如，已知的實體字元只能代表邏輯字元的一部份。

XmText 動作常式

下列清單說明 `XmText` 動作常式。

`left-character(extend)`

如果 `XmNeditPolicy` 是 `XmEDIT_LOGICAL`，且呼叫它時沒有使用引數，則插入游標會邏輯式地往後移動一個字元。如果插入游標是在行頭，插入游標就會移到前一行的最後一個字元 (如果有的話)。否則，插入游標位置就不會變更。

如果 `XmNeditPolicy` 是 `XmEDIT_VISUAL`，那麼游標就會移到游標位置的左側。如果插入游標是在行頭，那它會移到前一行的最後一個字元 (如果有的話)。

如果使用了 `extend` 引數來呼叫 `left-character()`，則插入游標會移動，延伸目前的選取範圍，就像沒有使用引數時一樣。

`left-character()` 動作以合理的 `XmCR_MOVING_INSERT_CURSOR` 值產生對 `XmNmotionVerifyCallback` 程序的呼叫。如果使用 `extend` 引數來呼叫的話，這個動作就會產生對 `XmNgainPrimaryCallback` 程序的呼叫。如需詳細資訊，請參閱「*Motif Programmer's Reference*」中的回呼說明。

`right-character(extend)`

如果 `XmNeditPolicy` 是 `XmEDIT_LOGICAL`，且呼叫它時沒有使用任何引數，則插入游標會邏輯式地往前移動一個字元。如果插入游標是在行尾，此動作會將插入游標移到下一行的開頭 (如果有的話)。

如果 `XmNeditPolicy` 是 `XmEDIT_VISUAL`，那麼游標就會移到游標位置的右側。如果插入游標是在行尾，那插入游標會移到下一行的開頭 (如果有的話)。

如果使用 `extend` 引數來呼叫，`XmNeditPolicy` 會移動插入游標，延伸目前的選取範圍，就像沒有使用引數時一樣。

`right-character()` 動作以合理的 `XmCR_MOVING_INSERT_CURSOR` 值產生對 `XmNmotionVerifyCallback` 程序的呼叫。如果使用 `extend` 引數來呼叫的話，這個動作就會產生對 `XmNgainPrimaryCallback` 程序的呼叫。如需詳細資訊，請參閱「*Motif Programmer's Reference*」中的回呼說明。

`right-word(extend)`

如果 `XmNeditPolicy` 是 `XmEDIT_LOGICAL`，且呼叫它時沒有使用任何引數，則插入游標會移到接續文字的起始字元 (如果有的話)。否則，游標會移到目前文字的結尾。如果插入游標位於行尾或是一行的最後一個字，游標就會移到下一行的第一個字 (如果有的話)。否則，游標會移到目前文字的結尾。

如果 `XmNeditPolicy` 是 `XmEDIT_VISUAL`，且未使用引數呼叫，插入游標會移到行尾右側或之後第一個空白字元之後的非空白字元。

如果使用 `extend` 引數來呼叫，引數會移動插入游標，延伸目前的選取範圍，就像沒有使用引數時一樣。

`left-word()` 動作以合理的 `XmCR_MOVING_INSERT_CURSOR` 值產生對 `XmNmotionVerifyCallback` 程序的呼叫。如果使用 `extend` 引數來呼叫的話，這個動作就會產生對 `XmNgainPrimaryCallback` 程序的呼叫。如需詳細資訊，請參閱「*Motif Programmer's Reference*」中的回呼說明。

`delete-left-character()`

如果 `XmNeditPolicy` 是 `XmEDIT_LOGICAL`，它就相當於 `delete-previous-char()`。如果 `XmNeditPolicy` 是 `XmEDIT_VISUAL`，那麼在一般模式中，如果選取範圍是非 `null`，便會刪除此選取範圍。否則，此動作會刪除插入游標左邊的字元。在增加模式中，如果選取範圍是非 `null`，而游標和選取範圍並未切斷連接，且 `XmNpendingDelete` 已設為 `True`，則此動作會刪除選取範圍。否則，此動作會刪除插入游標左邊，對選取範圍產生影響的字元。

`delete-left-character()` 動作以合理的 `XmCR_MODIFYING_TEXT_VALUE` 值產生對 `XmNmodifyVerifyCallback` 程序的呼叫，並以合理的 `XmCR_VALUE_CHANGED` 值產生對 `XmNvalueChangedCallback` 程序的呼叫。

`delete-right-character()`

如果 `XmNeditPolicy` 是 `XmEDIT_VISUAL`，它就相當於 `delete-next-character()`。如果 `XmNeditPolicy` 是 `XmEDIT_VISUAL`，那麼在一般模式中，如果選取範圍是非 `null`，它就會刪除選取範圍。否則，它會刪除插入游標右邊的字元。在增加模式中，如果出現非 `null` 的選取範圍，且游標和選取範圍並未切斷連接，則 `XmNpendingDelete` 會設為 `True`，選取範圍也會被刪除。否則，它會刪除插入游標右邊的字元。這個動作會對選取範圍產生影響。

`delete-right-character()` 動作以合理的 `XmCR_MODIFYING_TEXT_VALUE` 值產生對 `XmNmodifyVerify-Callback` 程序的呼叫，並以合理的 `XmCR_VALUE_CHANGED` 值產生對 `XmNvalue-ChangedCallback` 程序的呼叫。

某些以顯示格為基礎的常式是為支援字元的複合、連音和分音片段而執行。換句話說，兩個或兩個以上的字元可以用佔用了一個顯示格的單一文字影像來代表。

下列清單說明 XmText 顯示格動作常式。

backward-cell (extend)

將插入游標往回移一個顯示格。如果 XmNeditPolicy 是 XmEDIT_LOGICAL，則插入游標就會移到目前顯示格的前一個顯示格開頭 (如果有的話)。否則，游標會移到目前顯示格的開頭。

如果 XmNeditPolicy 是 XmEDIT_VISUAL，那麼游標就會移到左側顯示格的開頭 (如果有的話)。prev-cell() 動作以合理的 XmCR_MOVING_INSERT_CURSOR 值產生對 XmNmotionVerifyCallback 程序的呼叫。如果使用 extend 引數來呼叫的話，這個動作就會產生對 XmNgainPrimaryCallback 程序的呼叫。如需詳細資訊，請參閱「*Motif Programmer's Reference*」中的回呼說明。

forward-cell (extend)

將插入游標移到下一個顯示格的開頭 (如果有的話)。否則，此動作會將游標移到顯示格結尾。如果 XmNeditPolicy 是 XmEDIT_LOGICAL，那麼游標會往前移一個顯示格。

如果 XmNeditPolicy 是 XmEDIT_VISUAL，則游標會移到游標位置右邊顯示格的開頭 (如果有的話)；否則，它會移到目前顯示格結尾。forward-cell() 動作以合理的 XmCR_MOVING_INSERT_CURSOR 值產生對 XmNmotionVerifyCallback 程序的呼叫。如果使用 extend 引數來呼叫的話，這個動作就會產生對 XmNgainPrimaryCallback 程序的呼叫。如需詳細資訊，請參閱「*Motif Programmer's Reference*」中的回呼說明。

XmTextFieldGetLayoutModifier 資源

XmTextFieldGetLayoutModifier() 會傳回反映產生版面配置物件狀態的版面配置修飾元字串。

XmTextFieldGetLayoutModifier() 的語法是：

```
#include <Xm/TextF.h>
```

字串 XmTextFieldGetLayoutModifier(視窗元件 視窗元件)

XmTextFieldGetLayoutModifier() 會存取目前因視窗元件產生的版面配置物件設定的值。如要使用簡易函式變更版面配置物件修飾元的值，XmTextFieldGetLayoutModifier 函式會傳回版面配置物件的完整狀態，而不只是變更的值。

XmTextFieldGetLayoutModifier() 以字串值的形式傳回版面配置物件修飾元的值。

XmTextGetLayoutModifier 資源

XmTextGetLayoutModifier() 會傳回反映產生版面配置物件狀態的版面配置修飾元字串。

XmTextGetLayoutModifier() 的語法是：

```
#include <Xm/Text.h>
String XmTextGetLayoutModifier(視窗元件 視窗元件)
```

XmTextGetLayoutModifier 會存取目前因視窗元件產生的版面配置物件設定的值。如果使用簡易函式變更版面配置物件修飾元的值，XmTextGetLayoutModifier 函式會傳回版面配置物件的完整狀態，而不只是變更的值。

XmTextGetLayoutModifier 以字串值的形式傳回版面配置物件修飾元的值。

XmTextFieldSetLayoutModifier 資源

XmTextFieldSetLayoutModifier() 會設定版面配置修飾元值，該值會變更版面配置物件的產生行爲。

XmTextFieldSetLayoutModifier() 的語法是：

```
#include <Xm/TextF.h> \
void XmTextFieldSetLayoutModifier(視窗元件 \
視窗元件 string layout_modifier)
```

XmTextFieldSetLayoutModifier 會修改與視窗元件相關聯的版面配置物件產生設定。當您使用此簡易函式設定版面配置物件修飾元時，只有輸入參數中指定的屬性會改變。其餘的屬性則維持不變。

XmTextSetLayoutModifier 資源

XmTextSetLayoutModifier() 會設定版面配置修飾元值，該值會變更版面配置物件的產生行爲。

XmTextSetLayoutModifier() 的語法是：

```
#include <Xm/Text.h>
void XmTextSetLayoutModifier(視窗元件 視窗元件, 字串 layout_modifier)
```

XmTextSetLayoutModifier 修改與視窗元件相關聯的版面配置物件產生設定。當您使用此簡易函式設定版面配置物件修飾元時，只有輸入參數中指定的屬性會改變；其餘的屬性則維持不變。

XmStringDirectionCreate 資源

XmStringDirectionCreate 會建立 複合字串。

XmTextSetLayoutModifier() 的語法是：

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

XmStringDirectionCreate 會使用單一元件及給定值的方向來建立複合字串。此外，針對任何不具有指定字串方向之元件的複合字串 (XmString)，XmNlayoutDirection 資源會設定預設產生方向。因此，如果要設定版面配置方向，您必須為 XmNlayoutDirection 資源設定適當的值。您不需建立具有特定方向元件的複合字串。

當應用程式產生了 XmString，應用程式應該檢查字串建立時是否有明確方向 (XmStringDirection)。如果沒有提供方向元件給應用程式，應用程式應該檢查目前視窗元件資源的 XmNlayoutDirection 值，並將該值視為 XmString 的預設產生方向。

UIL 引數

下表顯示 UIL 引數名稱及其類型。

表 6-3 UIL

UIL 引數名稱	引數類型
XmNlayoutAttrObject	字串型
XmNlayoutModifier	字串型
XmNrenditionTag	字串型
XmNalignment	整型
XmNeditPolicy	整型

開發 CTL 應用程式

下列章節說明如何開發 CTL 應用程式。

控制版面配置方向

您可以儲存複合字串的方向，這樣資料結構就可以用於描述從左至右方向的文字 (例如，英文、西班牙文、法文和德文)或是從右至左方向文字 (例如，希伯來文和阿拉伯文)。在 Motif 應用程式中，您可以從 `VendorShell` 或 `MenuShell`，使用 `XmNlayoutDirection` 資源來設定版面配置方向。「管理員」及「原始」視窗元件和 `Gadget` 一樣也擁有 `XmNlayoutDirection` 資源。預設值是得自於具有相同資源的最接近祖先。

在使用 `XmText` 視窗元件的情況下，您必須指定垂直方向以及水平方向。將 `layoutDirection` 設定為 `XmRIGHT_TO_LEFT`，可以讓字串方向成為從右至左，但是游標會垂直向下。如果垂直方向很重要，而且您需要由上至下的對齊方式，請確認指定了 `XmRIGHT_TO_LEFT_TOP_TO_BOTTOM`。這個設定會指定先從右至左，再由上至下來配置元件，並產生期望的行為。

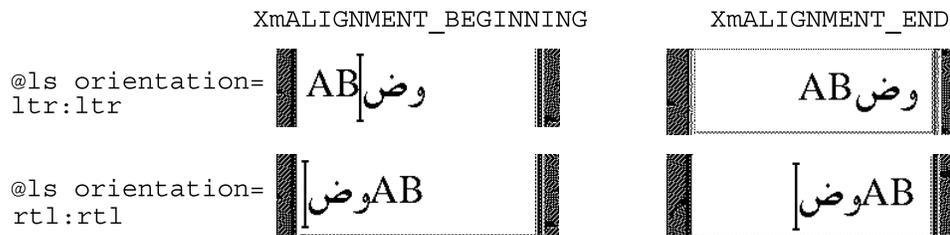
`XmText` 和 `TextField` 視窗元件的行為同樣受到 `XmRendition` 的 `XmNalignment` 和 `XmNlayoutModifier` 資源的影響。這些資源 (除了 `XmNlayoutDirection`) 會一起控制「文字」視窗元件的版面配置行為。這個行為將說明於圖 6-2。

該圖所使用的輸入字串為：

A B و ض

下圖 `XmNlayoutModifier` 字串 `@ls orientation=` 的設定值會顯示於左邊欄位。

版面配置方向：XmLEFT_TO_RIGHT



版面配置方向：XmRIGHT_TO_LEFT

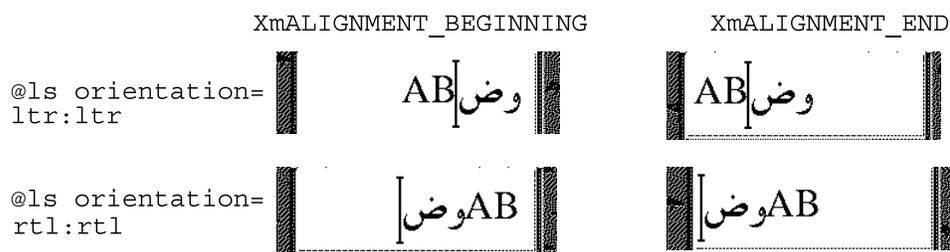


圖 6-2 版面配置方向

如圖所示，XmNalignment 會指示文字配合版面配置方向向右對齊或向左對齊。XmNlayoutModifier 會將文字分成幾個段落，並依據方向值將它們由左至右或由右至左排列。換言之，如果 XmNlayoutDirection 是 XmRIGHT_TO_LEFT，且 XmNalignment 值是 XmALIGNMENT_BEGINNING，該字串就會向右對齊。

範例 6-1 建立 Rendition

下列的程式碼，使用其標記是「ArabicShaped」的 Rendition，建立一個其 XmNlabelString 是 XmCHARSET_TEXT 類型的 XmLabel。此 Rendition 是使用「ar」（相對應到阿拉伯文語言環境的語言環境名稱）的 XmNlayoutAttrObject 以及為輸出緩衝區指定 NUMERALS_CONTEXTUAL 的 Numerals 值和「iso8859-6」的 ShapeCharset 值之版面配置修飾元字串來建立的。

語言環境特定的版面配置模組會將其本身的輸入文字，轉換為使用 16 位元 Unicode 字碼集進行編碼的實體字元輸出緩衝區。因為已經指定明確版面配置語言環境，這段文字會獨立於執行階段語言環境設定適當地產生。在此範例中，輸入是使用 ISO 8859-6 來編碼。

```
int n;
Arg args[10];
Widget w;
XmString labelString;
XmRendition rendition;
```

範例 6-1 建立 Rendition (續)

```
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,
                             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,
                              s, n);
renderTable =
    XmRenderTableAddRenditions(NULL, &rendition, 1, XmREPLACE_MERGE);
XtVaSetValues(w, XmNrenderTable, renderTable, NULL);
```

範例 6-2 編輯 Rendition

下列程式碼使用單一 Rendition 來建立 TextField 視窗元件和 RenderTable。XmNlayoutAttrObject 和 XmNlayoutModifier 兩個虛擬資源都未被指定，因此將預設為 NULL。這個值表示與 Rendition 有關聯的版面配置物件屬於預設語言環境，若其存在的話。

如果要順利執行這個範例，語言環境必須設定成具有整個字碼集是 ISO 8859-6，且其語言環境特定版面配置模組可以支援 IMPLICIT_BASIC 演算法。此 Rendition 的 LayoutObject 的 ImplicitAlg 值已被修改，修改方式是透過 Rendition 的 XmNlayoutModifier 虛擬資源。

```
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);
```

範例 6-2 編輯 Rendition (續)

```
....
n = 0;
XtSetArg(args[n], XmNlayoutModifier, "@ls implicitalg=basic");
n++;
XmRenditionUpdate(rendition, args, n);
```

在資源檔案中建立 Render 表格

Rendition 以及 render 表格應該在資源檔案中，針對適當國際化的應用程式進行指定。當您將 render 表格指定於一個檔案時，程式的二進位碼就可因應指定語言環境的特殊需求而獨立產生，且可輕易地依照該語言環境需求完成自訂。

Render 表格會以下列語法指定於資源檔案中：*resource_spec*: [*tag* [, *tag*] *

其中 *tag* 是某些適合 XmNtag rendition 資源的字串。

這一行將建立初始 render 表格，其將包含一個或更多個指定的 rendition。這些 rendition 將附加指定標記：

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

下列範例將說明與 XmRendition (可以使用資源檔案設定) 相關聯的 CTL 資源。fontType 必須設定為 FONT_IS_XOC，才能使版面配置物件生效。layoutModifier (使用 @ls 指定) 已經由 rendition 物件傳遞至版面配置物件。

如需可以使用 layoutModifier 在版面配置上設定之資源的清單，請參閱「CAE Specification: Portable Layout Services: Context-dependent and Directional Text」，The Open Group：Feb 1997；ISBN 1-85912-142-X；document number C616。

範例 6-3 在應用程式中建立 Render 表格

在建立 render 表格之前，應用程式必須先建立至少一個該表格中的一個 rendition。XmRenderTableAddRenditions() 函式一如其名所示，也是使用新的 rendition 來強化 render 表格。若要建立新的 render 表格，請呼叫 XmRenderTableAdd s() 函式，並以 NULL 引數來替換現有的 render 表格。

下列程式碼將使用 rendition，將 XmNfontType 設定為 XmFONT_IS_XOC，來建立 render 表格。

```
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
```

範例 6-3 在應用程式中建立 Render 表格 (續)

```
        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=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);
```

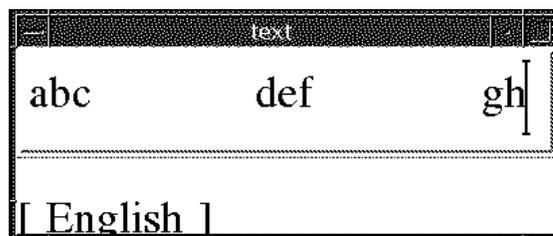
水平定位點

複合字串可以包含用於控制文字位置的定位點字元。若要為顯示目的解譯這些字元，視窗元件會提交定位停駐點清單給產生，以使複合字串生效。然而，動態視窗元件的 `TextField` 和 `XmText`，並不使用產生的定位點資源。而是使用 $8*$ (字元 0 的寬度) 公式來計算定位點寬度。

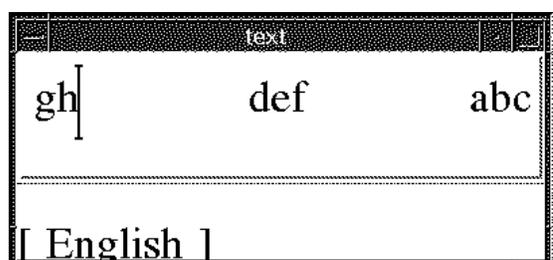
定位點的測量是從複合字串顯示的左邊界開始的距離。如果版面配置方向是從右至左，這個距離就會從右邊界開始計算。不管文字方向為何 (從右到左的阿拉伯文或是從左到右的英文)，此定位點都會依照版面配置方向 (`XmNlayoutDirection`) 的指定，從右邊或是左邊插入一個空白字元。

定位點後面的文字永遠會對齊於定位點停止位置。定位點停止位置是從視窗元件起始處開始計算，其依次受到 `XmNlayoutDirection` 的影響。下圖將說明定位點的行為及其行為與文字方向性的互動方式以及與視窗元件的 `XmNlayoutDirection` 的互動方式。

這個說明的輸入是 `abc\tdef\tgh`。



版面配置方向: XmLEFT_TO_RIGHT



版面配置方向: XmRIGHT_TO_LEFT

圖 6-3 定位點行為

滑鼠選取

使用者以滑鼠按鈕 1 做了主要選取。按下此按鈕取消選取任何現存的選取，並將插入游標和控點移動到使用者在文字中按下按鈕的所在位置。一邊按住同時拖曳滑鼠 1，便可選取所有位於控點和指標位置間的文字，來取消選取該範圍以外的任何文字。

所選取文字會受到 `XmNeditPolicy` 資源影響，此資源可以設定為 `XmEDIT_LOGICAL` 或 `XmEDIT_VISUAL`。如果 `XmNeditPolicy` 設定為 `XmEDIT_LOGICAL`，且所選取文字為雙向，則此選取文字看起來就不會是連續的，而是一群區段的集合。邏輯緩衝區中的文字與該顯示並不具有一對一的關聯性。

因此，雙向文字之邏輯字元的連續緩衝區，不會產生連續字元串流。相反地，如果 `XmNeditPolicy` 設定為 `XmEDIT_VISUAL`，所選取文字看起來是連續的，但在邏輯緩衝區中則被分為區段。因此在相同游標點上的序列選取、刪除和雙向文字插入，不會產生相同字串。

鍵盤選取

可以使用滑鼠的選取作業也同樣可以使用鍵盤。Shift 和方向鍵的組合可執行文字選取動作。

所選取文字會受到 `XmNeditPolicy` 資源影響，此資源可以設定為 `XmEDIT_LOGICAL` 或 `XmEDIT_VISUAL`。如果 `XmNeditPolicy` 已設定為 `XmEDIT_LOGICAL`，且所選文字是雙向的，所選文字看起來就是不連續的。因為在邏輯緩衝區中的文字與該顯示並不具有一對一的關聯性，因此，雙向文字之邏輯字元的連續緩衝區，不會產生連續字元串流。

相反地，如果 `XmNeditPolicy` 設定為 `XmEDIT_VISUAL`，所選取文字看起來是連續的，但在邏輯緩衝區中則被分為區段。因此在相同游標點上的序列選取、刪除和雙向文字插入，不會產生相同字串。

文字資源與幾何學

下列是與幾何學有關的文字資源：

- 視窗元件用來選取字型或字型集以及其他顯示文字屬性的 `render` 表格 `XmNrenderTable`。
`Text` 和 `Textfield` 視窗元件只能使用與字型相關的產生資源，像是 `XmNfontType`。這些視窗元件也可以指定版面配置物件的屬性，例如 `XmNlayoutAttrObject`。這些視窗元件通常包含語言環境識別碼和 `XmNlayoutModifier`，後者會將指定的版面配置值，傳遞到與 XOC 關聯的「版面配置物件」，而 XOC 與 `XmRendition` 關聯。
- 資源 (`XmNwordWrap`)，可以指定當文字比視窗元件寬的時候，文字行是否會在文字邊界中斷。
在字的邊界中斷一行將不會在文字中插入新行。針對像是阿拉伯文的草寫體語言，如果文字長度大於視窗元件長度，文字會自動換行到下一行。但是，下一行第一個字元的成形，將與邏輯緩衝區中的先前字元無關。

移植指令

啓用「複合文字版面配置 (CTL)」功能的新 Motif 程式庫，儲放於 `/usr/dt/lib/libXm.so.4`。如果應用程式連結至 `libXm.so.3` 便無法支援 CTL。
`ldd app_name` 顯示應用程式所連結的程式庫。若要移植現有應用程式以便啓用 CTL，請執行下列步驟：

1. 新增 `-DSUN_CTL` 至 `Makefile`。
這個旗標很重要，且包括了可支援 CTL 的必要資料結構。此值應該在編譯期間設定。
2. 重新編譯現有的應用程式。
這個重新編譯將自動與啓用 CTL 的 Motif 程式庫 `libXm.so.4` 進行連結。
3. 新增 `XmText.translations` 資源至應用程式資源檔案中。如果沒有這些資源，就無法啓動語言環境的版面配置引擎。
4. 請參考附加至文件的範例應用程式。

注意 – 請使用在 `fontName` 資源中可用且適於您的語言環境之字型名稱。

舉例說來，如果您希望在 `XmTextField` 或 `XmText` 視窗元件中可以移動顯示格架構字元 (泰文)，請依照下列方式設定相關聯視窗元件的轉譯：

```
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

藉由 mp 所進行的列印過濾器增強

本章說明 mp 公用程式的列印增強功能。本章將討論下列主題：

- 第 169 頁的「UTF-8 的列印」
- 第 170 頁的「mp 列印過濾器增強概述」
- 第 171 頁的「使用 mp.conf 配置檔進行本土化」
- 第 177 頁的「語言環境相依 prolog 檔案」
- 第 177 頁的「新增及自訂 prolog 檔案」
- 第 177 頁的「PostScript 檔案自訂」
- 第 180 頁的「.xpr 檔案」

UTF-8 的列印

增強型 mp 列印過濾器可以列印多種輸入檔案格式，包括在目前 Solaris 環境中以 UTF-8 寫成的平面文字檔案。此列印過濾器使用 Solaris 系統上可用的 TrueType 與 Type 1 可縮放字型以及 X11 點陣圖字型。此過濾器也可以使用印表機內建字型，並扮演 X 列印伺服器用戶端的角色。

這個公用程式的輸出是標準的 PostScript™，並且可以發送到任一部 PostScript 印表機。當設定為 X 列印伺服器用戶端，且列印伺服器可支援 mp 時，mp 公用程式也可以輸出任何頁面描述語言。

若要使用該公用程式，請鍵入下列指令：

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

由於 mp 接受下列的 stdin 串流，因此也可以將公用程式當作過濾器來使用：

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

您可以將公用程式設定為行列式印表機的字列過濾器。例如，下列的指令序列會告知印表機服務 LP，lp1 印表機只接受 mp 格式的檔案。這個指令也會將 lp1 印表機安裝在 /dev/ttya 通訊埠上。如需詳細資訊，請參閱「lpadmin(1M) 線上說明手冊」。

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

您可以使用 `lpfilter(1M)` 來新增如下所列的過濾器公用程式：

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

該指令會告知 LP，可透過名為 `pathname` 的過濾器描述檔取得一個轉換程式 (在此例中指的是 `mp`)。 `pathname` 包含下列資訊：

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

過濾器可使用 `/usr/bin/mp` 將預設類型檔案輸入轉換為 PostScript 輸出。

若要列印 UTF-8 文字檔案，請使用下列指令：

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

如需詳細資訊，請參閱「`mp(1)` 線上說明手冊」。

mp 列印過濾器增強概述

目前 Solaris 發行版本中的 `mp` 列印過濾器已經過增強。為了在語言環境中產生可列印國際性文字的輸出檔，最新的 `mp` 可在內部以三種不同模式運作。可用的模式為：

- 使用語言環境特定的字型配置檔 `mp.conf`
- 使用語言環境特定的 PostScript prolog 檔 `prolog.ps`
- 以 Xprt (X 列印伺服器) 用戶端的方式運作

以下章節說明何時使用特定的列印方法，以及 `mp` 針對這些列印方法使用哪種配置和支援檔案。

以語言環境特定的字型配置檔 `mp.conf` 配合 `mp` 來使用

若是沒有在指令行中指定 `-D` 或 `-P` 選項，此列印方法即為預設值，除非 `prolog.ps` 檔案存在於 `/usr/openwin/lib/locale/$LANG/print` 或是 `/usr/lib/lp/locale/$LANG/mp` 目錄中。該 `prolog.ps` 檔案強制 `mp` 使用檔案中的 PostScript 內嵌式字型。即使 `prolog.ps` 存在於語言環境中，使用 `-M` 選項會忽略 `prolog.ps` 檔，並使用 `mp.conf` 檔案 (若存在的話) 做為替代。

此方法使用 `/usr/lib/lp/locale/$LANG/mp/mp.conf` 字型配置檔。您可能不需要變更此檔案，除非需要使用替代字型列印。此檔案可以使用 TrueType、Type 1 或 pcf 字型來配置。

藉由語言環境特定的 PostScript Prologue 檔案來使用 mp

`/usr/lib/lp/locale/C/` 目錄包含一般用於此列印模式的 `.ps` 列印頁面版面配置檔案。第 177 頁的「新增及自訂 prolog 檔案」中提供如何自訂這些檔案的說明。

若是指令行中沒有指定 `-D` 或 `-P` 選項，並且存在

`/usr/openwin/lib/locale/$LANG/print/prolog.ps`，那麼 `prolog.ps` 檔案即前置至輸出。根據 `.ps prolog` 頁的列印樣式，版面配置檔案亦前置至輸出。

此列印方法只利用 PostScript 字型檔。第 177 頁的「新增及自訂 prolog 檔案」中說明 `prolog.ps` 的檔案自訂。

將 mp 視為 Xprt (X 列印伺服器) 用戶端來使用

以 `mp` 做為 `Xprt` 用戶端，以使 `mp` 可列印 `Xprt` 列印服務所支援之網路連線印表機的輸出。做為 `Xprt` 用戶端，`mp` 支援 PostScript 和多種版本的 PCL。

`Xprt` 用戶端依據下列規則來試圖連線到 `Xprt` 伺服器：

- 當 `-D printer_name@machine[:dispNum]` 或 `-P printer_name@machine[:dispNum]` 選項搭配 `mp` 指令一起使用，`mp` 會以 `printer_name` 嘗試連線到 `machine[:dispNum]` 上的 `Xprt` 列印服務。

如果上述連線到 `machine[:dispNum]` 的嘗試失敗或是指定給 `-D` 或 `-P` 的引數只有 `printer_name`，那麼 `mp` 指令會檢查支援 `printer_name` 引數之 `Xprt` 伺服器的 `XPSEVERLIST`。例如：

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

- 如果未找到任何使用上述規則的伺服器，`mp` 會檢查 `XPDISPLAY` 環境變數是否設定為 `machine[:dispNum]`。例如：

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

- 如果 `XPDISPLAY` 變數未設定或是無效狀態，則 `mp` 會嘗試連線到預設顯示：2100。如果預設顯示值也是無效的，`mp` 會出現錯誤訊息。

`/usr/lib/lp/locale/C/mp` 目錄包含 `Xprt` 用戶端的 `.xpr` 列印頁面範例版面配置檔案。這些範例檔是針對 300 dpi 印表機所建立的。如果目標印表機具有不同的 dpi 值，範例檔的 dpi 值會自動地轉換為與目標印表機相符的解析度。

使用 mp.conf 配置檔進行本土化

配置檔提供新增或變更字型項目或字型群組項目的彈性。

系統預設的配置檔是 `/usr/lib/lp/locale/$LANG/mp/mp.conf`，其中 `$LANG` 是執行列印之語言環境中的語言環境變數。您可以使用 `-u config.file path` 選項來指定個人使用的配置檔。

一個針對相容性而被編碼為字元的 *ligature* 或變數文字影像稱為**表示格式**。mp.conf 檔案主要是用來將語言環境中的中間字碼點，對映至用來列印該字碼點的字型編碼中的表示格式。

中間字碼點可以是寬字元或是「可攜式版面配置服務 (PLS)」層的輸出。若要進行「複合文字版面配置」列印，則中間字碼點必須是 PLS 輸出。由 mp 產生的預設中間字碼是 PLS 輸出。

目前支援的字型格式為可攜式編譯格式 (PCF)、TrueType 以及 Type1 格式。同時支援系統與印表機內建的 Type 1 字型。請注意下列關於 mp.conf 配置檔的格式與內容的資訊：

- 每行必須以有效的關鍵字 (指令) 起始。
- 關鍵字的引數必須與關鍵字出現在同一行。
- 以 # 字元當做起始的行將被視為註釋直到該行結尾。
- 以 0x 做為起始的數值引數將解譯為十六進位數。

mp.conf 檔案中不同的區段包括：

- 字型別名
- 字型群組定義
- 將語言環境中的中間字碼範圍對映至字型群組
- 將每個字型與共用物件產生關聯，而該物件會將中間字碼點對映至字型編碼中的表示格式。

字型別名

mp.conf 檔案中的字型別名區段是用來定義每個用於列印字型的別名。此區段中的每一行皆為如下的格式：

```
FontNameAlias font-alias-name font-type font-path
```

font-alias-name

為字型名稱加入別名的一般慣例是在後方加上表示該字體為羅馬字體、粗體、斜體或粗斜體的字母 (R、B、I 或 BI)，藉以指定字型的編碼/程序檔名稱。

例如，`/usr/openwin/lib/X11/fonts/75dpi/courR18.pcf.Z`，因為它是 iso88591 Roman 字型，所以可以指定 iso88591R 別名。

font-type

.pcf 的字型類型值是 PCF、Adobe Type1 的字型類型值是 Type1 以及 TrueType 的字型類型值是 TrueType。只有這三種字型可以在 mp.config 檔案中配置。

font-path

字型檔案的絕對路徑名稱。若是 Type1 印表機內建字型，則只需指定字型名稱，例如 Helvetica。

例如，

```
FontNameAlias prnHelveticaR Type1 Helvetica
```

字型群組定義

您可以結合相同類型的字型以構成字型群組。字型群組的格式如下列所示：

<i>keyword</i>	FontGroup。
<i>fontgroupname</i>	字型的群組名稱。
<i>GroupType</i>	字型類型。只為相同類型的字型建立字型群組 (PCF, Type1, TrueType)。
<i>Roman</i>	字型群組中的羅馬字型名稱。
<i>Bold</i>	字型群組中的粗體字型名稱。
<i>Italic</i>	字型群組中的斜體字型名稱。
<i>BoldItalic</i>	字型群組中的粗斜體字型名稱。

針對建立群組，只需要羅馬字型登錄。粗體、斜體以及粗斜體字型為選擇性的。例如，不同類型的字型是用來顯示郵件或新聞文章的標題行。若是只定義 Roman 字型，則此字型是用來代替其他字型。

對映區段

mp.conf 檔案的對映區段從中間字碼的範圍對映至語言環境中的字型群組。這個區段中每一行的格式如下所示：

<i>keyword</i>	MapCode2Font。
<i>range_start</i>	一個 4 位元十六進位值是以 0x 開頭，指出字碼範圍的起點以對映至一個或多個字型群組。
<i>range_end</i>	表示需要對映的字碼範圍結尾。如果此值是 '-'，則其中只有單一中間字碼點對映至目標字型。
群組	將列印其表示格式的 Type1、PCF 或 TrueType 字型群組。

關聯區段

mp.conf 檔案的關聯區段可讓每個字型與共用物件產生關聯，而該物件會將中間字碼點對映至字型編碼中的表示格式。這個區段中每一行的格式如下所示：

關鍵字	CnvCode2Font。
<i>font alias name</i>	為字型所定義的別名。
<i>mapping function</i>	可依序透過中間字碼格式取得該對映功能，並以字型編碼回傳，以取得文字影像索引，並繪製文字影像。

file path having mapping function 包含對映功能的 .so 檔案名稱。您可以使用 dumpcs 中的公用程式，來查找 EUC 語言環境的中間字碼集。

注意 – 目前 mp (1) 所使用的 TrueType 引擎，只能使用格式 4 和 PlatformID 3 cmap。您只能配置 Microsoft .ttf 檔案。此外，字元對映編碼必須是 TrueType 字型引擎的 Unicode 或符號才能正確地作業。由於 Solaris 環境中的大部分 .ttf 字型都遵循這些限制，因此您可以在 mp.conf 檔案中對映 Solaris 軟體的所有 TrueType 字型。

您可以建立一個對映到字型的共用物件，以與 PCF type1 X 邏輯字型描述 (XLFD) 相應。然後您可以建立從中間字碼範圍對映到 XLFD 所指定之編碼的共用物件。例如：

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

相對應的 PCF 字型為：

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

由於此字型是以 ISO 8859-8 編碼的，因此共用物件必須在中間字碼和相應的 ISO 8859-8 字碼點之間對映。

若是 TrueType 字型使用 XLFD：

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

擁有相對應的字型：

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

您應該在中間字碼與 Unicode 之間對映，因為上一個 TrueType 字型的 cmap 編碼為 Unicode。在這個 TrueType 字型的範例中，假設 en_US.UTF-8 語言環境中與希伯來字元 (由 PLS 層所產生) 相對應的中間字碼範例為 0xe50000e9。由於該字型為 Unicode 編碼，因此當您傳送 0xe50000e9 時，其輸出會對應於 Unicode 中的表示格式，您可依此方式設計相對應 .so 模組中的功能。這裡的範例是 0x000005d9。

對映功能的函式原型應該為：

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

下列是可在 mp.conf: 檔案中使用的可選關鍵字/值對。

```
PresentationForm      WC/PLSOutput
```

預設值為 PLSOutput。若是使用者指定 WC，則所產生的中間字碼點為寬字元。針對 CTL 列印，應該使用此預設值。

如果該語言環境為非 CTL 的語言環境且其鍵盤值是 PLSOutput，則該值會被忽略，mp 也會產生寬字元字碼做為替代。

如果語言環境支援 CTL，您可以使用列於下表的可選關鍵字/值對。這些變數可以假設表格中間欄中所給定的任何可能值。

表 7-1 可選關鍵字/值對

可選關鍵字	可選值	預設值
Orientation	ORIENTATION_LTR/ ORIENTATION_RTL/ ORIENTATION_CONTEXTUAL	ORIENTATION_LTR
Numerals	NUMERALS_NOMINAL/ NUMERALS_NATIONAL/ NUMERALS_CONTEXTUAL	NUMERALS_NOMINAL
TextShaping	TEXT_SHAPED/ TEXT_NOMINAL/ TEXT_SHFORM1/ TEXT_SHFORM2/ TEXT_SHFORM3/ TEXT_SHFORM4	TEXT_SHAPED

▼ 如何新增印表機內建字型

下列程序中的範例，將展示如何在配置檔中加入新的 PCF、TrueType 或 Type1 印表機內建字型。

請完成此程序，以更換目前所配置的字型。在頭兩個步驟中，用來顯示範圍 0x00000021 - 0x0000007f 中字元的 PCF 字型，由 TrueType 字型來取代。

1. 在增加新的字型之前，請先在配置檔案中查看與目前配置的字型相對應的各種元件。

```
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，您可以將

`/usr/openwin/lib/locale/ja/X11/fonts/TT/HG-MinchoL.ttf` 字型對映到 `en_US.UTF-8` 語言環境。由於 `HG-MinchoL.ttf` 是 Unicode TrueType 字型檔，因此使用 `.so` 模組對映功能即可直接傳回內送的 `ucs-2` 字碼點。

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

a. 將此對映儲存到 `ttfjis0201.c` 檔案。

b. 建立共用物件檔案。

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

2. 若要對映 PCF 檔案，例如

`/usr/openwin/lib/locale/ja/X11/fonts/75dpi/gotmrk20.pcf.Z`，請檢查 `/usr/openwin/lib/locale/ja/X11/fonts/75dpi/fonts.dir` 檔案中與 XLFD 相應的下列編碼：

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

a. 若是 `jisx0201` 編碼方式，請準備好從 `ucs-2` 對映至 `jisx0201` 的共用物件。取得對映表，以建立 `.so` 模組。若是 Unicode 語言環境，請在 ftp.unicode.org/pub/MAPPINGS/ 目錄中尋找對映至 Unicode 的字元集。

b. 使用這些對映來撰寫 `xu2jis0201.c` 檔案：

```
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. 當您建立對映檔時，請包含所有 `usc-2` 至 `jisx0201` 的實例。

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

▼ 如何建立共用物件檔案

下列程序中的範例，將展示如何建立共用物件檔案。

1. 若要新增字型，請編輯下列範例中與 `mp.conf` 檔案中區段相應的行。

此範例顯示如何新增 TrueType 字型。`.so` 路徑會指向 `xu2jis0201.so` 檔案。

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

```
CnvCode2Font jis0201R _tffjis0201 <..so path>
```

注意 – 若要新增 PCF 字型，請將關鍵字由 TrueType 變更為 PCF。

2. 使用變更後的 `mp.conf` 檔案來呼叫 `mp` 指令，以使用新字型來列印範圍 `0x0020-0x007f`。

若要對映其他的日文字元範圍，您可以使用同一個 `so` 檔案來達成。例如，您可以對映範圍 `0x0000FF61-0x0000FF9F`。

注意 – 若要維持向後相容性，您可以使用 `/usr/openwin/lib/locale/$LANG/print/prolog.ps` 檔案在目前的語言環境中建立輸出。當您使用 `prolog.ps` 檔案時，不需要任何配置檔。

您可以在 `/usr/lib/lp/locale/en_US.UTF-8/mp` 目錄中找到 `mp.conf` 範例檔。

新增及自訂 `prolog` 檔案

`prolog` 檔案可以區分為兩個主要種類：

- PostScript `prolog` 檔案 (`.ps`)
- X print server client `prolog` 檔案 (`.xpr`)

PostScript 檔案自訂

PostScript 檔案縮減為下列種類：

- 共用 `prolog` 檔案
- 列印版面配置 `prolog` 檔案

語言環境相依 `prolog` 檔案

`prolog.ps` 檔案的目的是設定非一般的字型。應用程式使用這些預先定義的 PostScript 字型名稱來列印。針對「桌面設定行事曆」管理員和 `mp`、`prolog` 檔案至少必須定義下列字型名稱：

- `LC_Times-Roman`
- `LC_Times-Bold`

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

以下範例將使用這些字型來列印指定的特定本土字元集：

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

Solaris 本土化工具包為日文環境提供 `prolog.ps` 範例檔案。此外，您可以在 `/usr/openwin/lib/locale/ja/print/` 目錄中找到這個檔案。

下列範例將顯示如何在現有的 `prolog.ps` 檔案中增加或變更複合字型。

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

例如，您可以定義名為 `LC_Base-Font` 的複合字型。`LC_Base-Font` 可能是 *Foo-Fine* 的複合字型，其中包含語言環境字元集和 *Base-Font*。新增或變更字型並不需要深入瞭解 PostScript 程式設計的知識。

研讀範例版本是建立 `prolog.ps` 檔案的最佳方法。在 `prolog.ps` 範例中，必須寫入兩個常式：`makecodeset12` 和 `makeEUCfont`。`makecodeset12` 常式會設定本機字型編碼方式的資訊。此常式可能因語言環境而異。`makeEUCfont` 常式會合併基本字型和語言環境字型來形成複合字型。`prolog` 檔案的建立者應該擁有良好的 PostScript 知識，以便寫入 `makecodeset12` 和 `makeEUCfont`。

`prolog.ps` 檔案支援僅保留供向後相容性使用。請勿為了產生語言環境的列印輸出而建立新的 `prolog.ps` 檔案。請使用 `mp.conf` 做為替代。

`prolog.ps` 檔案的路徑是

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

共用的 PostScript prolog 檔案

共用的 `prolog` 檔案是 `mp.common.ps`。

每個其他頁面版面配置 `prolog` 檔案都需要包含此檔案。

`mp.common.ps` 檔案位於 `/usr/lib/lp/locale/C/mp/` 目錄中。這個檔案包含一個 PostScript 常式，此常式可將字型從標準編碼重新編碼為 ISO 8859-1 編碼。可呼叫列印版面配置 `prolog` 檔案中的 `.reencodeISO` 常式來變更字型的編碼。通常此 `prolog` 檔案不需要任何自訂。若是您建立屬於自己的 `prolog` 檔案，請將環境變數 `MP_PROLOGUE` 設定為指向包含修改後 `prolog` 檔案的目錄。

列印版面配置 prolog 檔案

列印版面配置 prolog 檔案和 mp.*.ps 檔案，都包含控制列印頁面版面配置的常式。除了以使用者名稱、列印日期以及頁碼來給定列印頁面的頁首和頁尾之外，這些 prolog 檔案還可以提供其他的資訊。例如，prolog 檔案可給定有效的列印區域大小以及列印的橫向和直向模式。

列印版面配置 prolog 檔案是：

- 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

需要在每一個 prolog 檔案中定義標準函式集。當起始新的列印頁面、列印頁面結束或新的一行結束時，即呼叫這些函式。這些函式的執行定義了列印輸出的列印屬性。

下列 PostScript 變數是由 mp 二進位檔在執行時間定義的。所有的列印版面配置檔案皆可以使用這些變數來列印動態資訊，像是 user name、subject 以及 print time。從變數取得的這項資訊通常會出現在列印頁面的頁首或頁尾。

<i>User</i>	執行 mp 的使用者名稱，從系統 passwd 檔案中取得。
<i>MailFor</i>	變數是用來保留要列印文章的類型名稱。此變數的可能值為： <ul style="list-style-type: none">■ 清單針對 - 當輸入為文字檔案時■ 郵件針對 - 當輸入為郵件檔案時■ 文章起始 - 當輸入為來自新聞群組的文章時
<i>Subject</i>	從郵件與新聞標頭處取得的主題。您可以使用 -s 選項，以強制加入主題至郵件或新聞檔案中，也可以加入至一般的文字檔案。
<i>Timenow</i>	出現在頁首或頁尾中的列印時間。此資訊是從 localtime() 函式取得。

在列印版面配置 prolog 檔案中執行下列的函式。這些函式皆可以使用子函式。

<i>endpage</i>	用法：page_number endpage 當到達列印頁面的底部時即呼叫。這個函式會復原頁面的圖形內容並送出 showpage。在部分 prolog 檔案中，頁首和頁尾資訊是以逐頁模式顯示，而非以逐欄模式顯示。您可以執行此函式來呼叫可顯示頁首和頁尾灰階菱形的子函式。
----------------	---

newpage	用法：page_number newpage 當新頁面開始時會執行常式或指令。這些常式所具備的功能包括設定橫向列印模式、儲存列印圖形內容以及翻譯頁面座標。
endcol	用法：page_number col_number endcol 此函式是用來顯示頁首和頁尾資訊、移動到新的列印位置等等。

若要增加新的列印版面配置 prolog 檔案，您需要在列印版面配置 prolog 檔案中明確定義下列變數。

<i>NumCols</i>	列印頁面的欄數。預設值為 2。
<i>PrintWidth</i>	以英吋計算的列印區域寬度。預設值為 6。
<i>PrintHeight</i>	列印區域的高度 (英吋)。預設值為 9。

.xpr 檔案

這些檔案的預設儲存位置是 /usr/lib/lp/locale/C/mp/。 .xpr 檔案相對應至每一個 PostScript prolog 版面配置檔案，除了 mp.common.ps. 以外。您可以藉由定義 MP_PROLOGUE 環境變數，來定義替代的 prolog 目錄。

這些檔案以關鍵字/值對的方式作業。以 # 起始的行將視為註釋。除非明確敘述否則以空白隔開不同的記號。每個 .xpr 檔案的三個主要區段，乃是藉由下列關鍵字對當做界限：

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

某些關鍵字/值對可以用在這三個區域中。下列章節將針對每個區域作說明。

STARTCOMMON/ENDCOMMON 關鍵字

所有出現在 STARTCOMMON 關鍵字之後並在 ENDCOMMON 關鍵字之前的關鍵字/值對，定義了列印頁面的一般屬性。關鍵字的數個不同有效值是由斜線 (/) 區分開來。

ORIENTATION 0/1
0 表示以縱向列印，1 表示以橫向列印。

PAGELength *unsigned-integer*
表示每一個邏輯頁面行數的值。

LINELENGTH *unsigned-integer*
表示每行單一欄位字元數的值。

NUMCOLS *unsigned-integer*
每一實體頁面的邏輯頁面數。

HDNGFONTSIZE *unsigned-integer*
以十分之一點計算的標題字型點大小。

BODYFONTSIZE *unsigned-integer*
以十分之一點計算的內文字字型點大小。

PROLOGDPI *unsigned-integer*
目前的 .xpr 檔案建立時所使用的每英寸點數標度。

YTEXTBOUNDARY *unsigned-integer*
此 y 座標可建立頁面或邏輯頁面 (欄位) 中的文字列印邊界。此邊界是用來當做額外的檢查，以便檢視文字列印是否發生在預期的區域內。當您從相對應的字型中取得字元高度資訊時可能會出現錯誤，因此在進行「複合文字版面配置」和 EUC 列印時將會需要此邊界。

STARTTEXT *unsigned-integer unsigned-integer*
當真實文字從實體頁面中的第一頁邏輯頁面開始列印時，x/y 點為十分之一點。

PAGESTRING 0/1
1 表示需要將頁面字串附加在標題中的頁面編號之前。
0 表示只顯示頁面編號。

EXTRAHDNGFONT *font string 1, font string 2, ... font string n*
字型字串是 X 邏輯字型描述。將關鍵字 EXTRAHDNGFONT 從逗號分隔字型名稱清單分隔開來的記號是引號 " 字元，不是空格或定位字元。在列印標題時，這些字型的優先順序將高於內建的字型。通常，EXTRABODYFONT 是用來指定配置在 /usr/openwin/server/etc/XpConfig/C/print/models/<model name>/fonts 目錄中的印表機內建字型。
該 fonts.dir 檔案包含印表機內建字型的 XLFD。
在 .xpr 檔案中，通常會指定字型，如下列範例所示：
"-monotype-Gill Sans-Regular-r-normal- *-%d-*-*p-0-iso8859-2"
若出現 %d，則將由 mp 取代為 .xpr 檔案中目前的標題字型點大小。x 解析度和 y 解析度是由 * 指定的。如果可以的話，將平均寬度欄位設定為 0，以指示選擇可縮放字型。您也可以提供更多指定的字型名稱。

EXTRABODYFONT *font string 1, font string 2, ... font string n*
除了這些字型是用來列印頁面內文之外，其餘與 EXTRAHDNGFONT 相同。

XDISPLACEMENT *signed/unsigned int*
針對頁面內容在 x 軸方向的移位，提供 x 軸座標位移以套用至頁面。此位移可以是 +ve 或 -ve 值。

YDISPLACEMENT *signed/unsigned int*
除了 y 軸中發生移位之外，其餘與 x 置換相同。

當您處理一些頁邊寬度並不符合標準的印表機，而需要移動頁面中的列印內容時，這兩個關鍵字將非常有用。

STARTPAGE/ENDPAGE 關鍵字

此區段中的關鍵字值對是以 STARTPAGE 以及 ENDPAGE 關鍵字當做邊界。此區段包含欲套用至實體頁面的繪圖與標題資訊。一張實體頁面可包含許多的邏輯頁面，但包含在這些關鍵字中的所有繪圖常式只能套用至實體頁面一次。

有效的繪圖實體為 LINE 和 ARC。XDrawLine() 和 XDrawArc() 函式將以這些關鍵字的值執行。

此區段中的維度是以 PROLOGDPI 單元做對映。角是以度計算。

LINE x1 y1 x2 y2	x/y 無符號座標可定義一對用來連接到線上的點。
ARC x y width height angle1 angle2	x 軸與 y 軸皆為表示 arc 原點的無符號整數。寬度與高度為表示 arc 寬度與高度的無符號整數。
USERSTRINGPOS x y	無符號座標表示使用者資訊列印在標題的位置。
TIMESTRINGPOS x y	無符號座標表示列印次數在標題的位置。
PAGESTRINGPOS x y	無符號座標表示列印每個列印頁面的頁面字串位置。
SUBJECTSTRINGPOS x y	無符號座標表示頁面中列印主題的位置。

STARTFORCEDPAGE/ENDFORCEDPAGE 區段

當指定 -n 選項予 mp 時，所有 STARTPAGE/ENDPAGE 區段中指定的裝飾皆不列印。然而，即使指定了 -n 選項，也會列印所有包含在 STARTFORCEDPAGE/ENDFORCEDPAGE 區段中的內容。

STARTCOLUMN/ENDCOLUMN 區段

所有關鍵字皆與第 182 頁的「STARTPAGE/ENDPAGE 關鍵字」中說明的相同，除了此區段中的項目是套用 NUMCOLS 次數至實體頁面上。如果 NUMCOLS 為 3，則實體頁面的可列印區域會分割成三部份和線、弧或每個頁面出現三次的標題裝飾。

STARTFORCEDCOLUMN/ENDFORCEDCOLUMN 區段

當 -n 選項給定為 mp 時，所有在 STARTCOLUMN/ENDCOLUMN 區段中所給定的裝飾都不會列印。然而，即使指定了 -n 選項，也會列印所有包含在 STARTFORCEDCOLUMN/ENDFORCEDCOLUMN 區段中的內容。

建立新的 .xpr 檔案

當建立新的 .xpr prolog 檔案時，您只能指定預設值以外的值。

如果這些值不是在 .xpr 檔案中為 STARTCOMMON/ENDCOMMON 區段所指定的，則下表將列出不同關鍵字的 mp 程式預設值。

表 7-2 STARTCOMMON/ENDCOMMON 關鍵字值

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

以 STARTPAGE/ENDPAGE 和 STARTCOLUMN/ENDCOLUMN 做為邊界的其他兩個區段不需要預設值。

若要建立不含裝飾的頁面，請以直向格式在每個實體頁面上使用四個邏輯頁面。指定下列的區段以及值：

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

當您建立不含裝飾的頁面時，不需要指定下列兩個區段：

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

如果您並未在列印頁面上放置裝飾，則不需要這些參數。所有座標的預設值都是 300 dpi，除非您沒有指定 PROLOGDPI 關鍵字。若是目標印表機的解析度不同，.xpr 檔案將藉由程式按比例縮放以符合該解析度。

在建立 .xpr 檔案之前，您必須知道紙張的大小。針對 U.S. 紙張，8.5x11 英吋，針對解析度 300 dpi 的印表機，2550X3300 為整體大小。大多數的印表機無法從紙張的最左上角開始列印。相反的，部分邊界空格是圍繞著實體頁面而指定的。即使您嘗試從 0,0 開始列印，實際的列印動作並不會從頁面的最左上角開始。當您建立新的 .xpr 檔案時，請考慮此限制。

附錄 A

iconv 字碼轉換

本附錄列出了目前 Solaris 作業環境中與 Unicode 有關的可用字碼轉換模組。

表 A-1 與 Unicode 有關的可用 iconv 字碼轉換模組

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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
euJP	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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode 有關的可用 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
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 為新的 IBM 字碼頁名稱。目前 Solaris 環境也支援 UTF-8 和 UTF-EBCDIC 字碼頁之間的雙向轉換。

下表列出了可在目前 Solaris 環境中找到的 Unicode、IBM/Microsoft EBCDIC 和 PC iconv 字碼轉換模組。

表 A-2 與 Unicode、IBM/Microsoft EBCDIC 和 PC 字碼頁有關的 iconv 字碼轉換模組

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode、IBM/Microsoft EBCDIC 和 PC 字碼頁有關的 iconv 字碼轉換模組
(續)

來源字碼 (符號)	目的字碼 (符號)
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 與 Unicode、IBM/Microsoft EBCDIC 和 PC 字碼頁有關的 iconv 字碼轉換模組 (續)

來源字碼 (符號)	目的字碼 (符號)
UTF-8	CP1255
UTF-8	CP1256
UTF-8	CP1257
UTF-8	CP1258

下表列出了在將 IBM 和 Microsoft EBCDIC/PC 字碼頁轉換為 UTF-8 時，可用的 iconv 字碼轉換。

表 A-3 可用的 iconv 字碼轉換 - 從 IBM 和 Microsoft EBCDIC/PC 字碼頁轉換為 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 字碼轉換 - 從 IBM 和 Microsoft EBCDIC/PC 字碼頁轉換為 UTF-8
(續)

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

表 A-3 可用的 iconv 字碼轉換 - 從 IBM 和 Microsoft EBCDIC/PC 字碼頁轉換為 UTF-8
(續)

UTF-EBCDIC	UTF-8
CP866	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

編號和符號

16 位元 Unicode 3.0 字碼集, 161

A

APIs, 國際化, 40

C

C 語言環境, 24

CDE

語言環境支援, 108

輸入法, 110

Compose 鍵, 33

Compose 鍵序列

Latin-1, 114

Latin-2, 117

Latin-3, 119

Latin-4, 120

Latin-5, 121

Latin-9, 121

希臘文, 130

希臘文, 三鍵, 134

希臘文, 四鍵, 136

重音固定鍵, 121

.cshrc, STREAMS 模組設定, 143

ctype 巨集, 40

D

DtMail, MIME 字元集, 145

dtterm, 143

E

en_US.UTF-8, FontSet 定義, 148

en_US.UTF-8, 支援, 108

F

FontSet 定義, 148

G

genmsg 公用程式, 47-48

GMT 偏移, 26

H

Hangul, 31

Hanja, 31

I

iconv

日文字元字碼轉換, 76

字碼轉換, 50

iconv 轉換模組, EBCDIC/PC 字碼頁, 209-211
iconv 轉換模組, Unicode, 185-206
ISO Latin-1, 23
ISO8859, 字元支援, 109

K

Kedmanee 鍵盤, 104
增強型, 104

L

LANG 環境變數, 140
Latin-1, Compose 鍵序列, 114
Latin-2, Compose 鍵序列, 117
Latin-3, Compose 鍵序列, 119
Latin-4, Compose 鍵序列, 120
Latin-5, Compose 鍵序列, 121
Latin-9, Compose 鍵序列, 121
LC_ALL, 23
libc
API, 37-38
日期與時間格式, 44
多位元組處理, 44
字元分類和轉譯, 42
字元比較, 43
字碼轉換函式, 41
查詢語言環境, 42
修改和查詢語言環境, 42
訊息函式, 41
常規表示式, 41
貨幣格式, 44
寬字元和字串處理, 45
寬字元輸入與輸出, 46, 47
寬字元類別, 42
寬字串, 46
應用程式連結, 35-50

M

mbtwoc, 40-47
Motif
TextField, 160
UIL 引數, 159
XmNalignment, 154, 160, 161

Motif (續)

XmNeditPolicy, 154
XmNlayoutDirection, 151
XmNlayoutDirection, 160
XmNlayoutModifier, 160, 161
XmNrenditionTag, 154
XmRendition, 152, 160
XmStringDirection, 152
XmStringDirectionCreate, 159
XmText, 160
XmTextFieldGetLayoutModifier, 157
XmTextFieldSetLayoutModifier, 158
XmTextGetLayoutModifier, 158
XmTextSetLayoutModifier, 158
mp
PostScript 變數, 179
TrueType, 18
Xprt 用戶端, 171
列印過濾器, 169, 170
mp.conf 檔案, 171

P

Pattajoti 鍵盤, 105
POSIX 語言環境, 24
PostScript
prolog 檔案, 177
執行時間變數, 179
prolog 檔案, 177

S

setlocale 指令, 140
SPARC 鍵盤, 63
strconf 指令, 142
STREAMS
TTY 環境, 140
字碼轉換, 140
載入模組, 141
stty, 公用程式, 143
Sun Ray, USB Type 6 鍵盤, 19

T

TTY 環境, 設定, 140

TypeOfText, 153

U

UIL, 159

Unicode

十六進位輸入模式, 139

概述, 107

轉換模組, 185-206

UTC, 26

UTF-8, 支援, 108

X

X 列印伺服器 (Xprt), mp, 171

X 邏輯字型描述 (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 應用程式, 40

.xpr 檔案, 177

xterm, 143

土

土耳其文, 字元支援, 109

土耳其文 F 鍵盤, 70

土耳其文 Q 鍵盤, 70

中

中文

ㄅ ㄆ ㄇ ㄏ, 32

台灣, 32

注音, 32

拼音, 32

香港, 32

漢字, 32

中文 (續)

輸入法, 20

中國, 中華人民共和國, 32

中華人民共和國, 32

丹

丹麥文鍵盤, 65

公

公用程式

genmsg, 48

genmsg, 47-48

iconv, 49

stty, 143

語言環境, 140

日

日文

iconv 模組, 76

片假名, 31

平假名, 31

本土化, 73-76

字元支援, 109

字元集, 73-74

字型, 74-75

完整本土化套裝軟體, 76

漢字, 31

語言環境, 73

輸入法, 75

輸入模式, 137

日文鍵盤, 66

日期格式, 27

比

比利時文鍵盤, 64

片

片假名, 31

北

- 北印度文
 - 字元支援, 109
 - 梵文字母, 32

卡

- 卡納達文鍵盤, 79

可

- 可攜式版面配置服務 (PLS), 171

古

- 古吉拉特文鍵盤, 78

平

- 平假名, 31

本

- 本土化, 73-105
 - 定義, 21-23
 - 配置檔, 171

列

- 列印過濾器, 169

匈

- 匈牙利文, 字元支援, 109

印

- 印度, 輸入法, 77

多

- 多位元組, 轉換, 40-47
- 多位元組字元, 38

字

- 字元
 - 支援, 109
 - 外形, 149-167
 - 多位元組, 38
 - 轉換, 40-47
 - 字元編碼
 - Unicode, 107
 - UTF-16, 107
 - UTF-32, 107
 - UTF-8, 107
 - 字母, 30
 - 字型
 - 日文 TrueType, 75
 - 日文點陣圖, 74
 - 別名, 172
 - 對映, 172
 - 字碼集, 23
 - 字元支援, 109
 - 轉換, 144
 - 字碼集獨立性 (CSI)
 - ASCII 斜線, 36
 - Java 國際化, 36
 - NULL 位元組, 36
 - Shift-JIS 字碼集, 35-37
 - 多位元組字元, 38
 - 指令, 36-37
 - 動態連結應用程式, 38-39
 - 程式庫, 37
 - 程序碼格式, 38
 - 語言環境資料庫, 37-38
 - 檔案程式碼編碼方式, 36
 - 擴充式 UNIX 編碼 (EUC), 35-37
- 字碼轉換模組, 185-206, 207-209
 - 字體選擇, 110

西

- 西班牙文, 字元支援, 109
- 西班牙文鍵盤, 68

西歐
字元支援, 109

完
完整本土化套裝軟體, 日文, 76

希
希伯來文
字元支援, 109
意第緒文, 32
輸入模式, 136
希臘文
字元支援, 109
輸入模式, 129, 130
希臘文 Euro, 鍵盤, 130
希臘文 UNIX, 鍵盤, 130

系
系統程式庫, 連結應用程式至, 38-39

坦
坦米爾文鍵盤, 80

孟
孟加拉文鍵盤, 78

果
果魯穆其文鍵盤, 79

波
波羅的海文, 字元支援, 109
波蘭文, 字元支援, 109

法
法文鍵盤, 65

版
版面配置行爲, 153

芬
芬蘭文鍵盤, 65

阿
阿拉伯文
字元支援, 109
輸入模式, 128
阿拉伯文鍵盤, 63

俄
俄文, 字元支援, 109

拼
拼音, 32

指
指令, CSI, 36-37

查
查表, 輸入模式, 139

美
美國/UNIX 鍵盤, 71
美國鍵盤, 71

英

英文

- 字元支援, 109
- 輸入模式, 113
- 英國鍵盤, 70

夏

夏令時 (DST), 27

挪

挪威文鍵盤, 67

時

時區, 27
時間格式, 26

泰

泰文, 31
字元支援, 109
字元順序檢查, 104-105
輸入法, 104, 105
鍵盤 配置, 20

特

特拉古文 鍵盤, 80

紙

紙張大小, 一般大小, 33

訊

訊息目錄, 47-48

國

國際化

- ISO Latin-1, 23
- 定義, 21-23
- 國際化 API, 40
- 國際化網域名稱 (IDN), 49

捷

捷克文, 字元支援, 109

梵

梵文鍵盤, 78

終

終端機
支援, 143
設定選項, 143

荷

荷蘭 (荷蘭文) 鍵盤, 67

連

連結, 應用程式, 38-39
連續語音輸入法, 印度文字體, 103

喀

喀拉拉文鍵盤, 80

斯

斯拉夫文, 輸入模式, 128
斯拉夫文鍵盤, 64
斯堪的納維亞文, 字元支援, 109

程

程式庫, (CSI), 37

瑞

瑞士 (法文) 鍵盤, 69

瑞士 (德文) 鍵盤, 69

瑞典文鍵盤, 68

義

義大利文鍵盤, 66

葡

葡萄牙文鍵盤, 68

對

對映

北印度文字元, 90

卡納達文字元, 92

古吉拉特文字元, 84

安得拉邦文字元, 101

坦米爾文字元, 98

孟加拉文字元, 81

英文與印度文字體的語音同義字元, 81

喀拉拉文字元, 95

對應, 果魯穆其文字元, 87

漢

漢字, 31

語

語言環境

C, 24

Compose 鍵, 58

POSIX, 24

Solaris, 73-105

中東, 55

語言環境 (續)

中美洲, 54

中歐, 54

文化慣例, 25

文字分割元, 30

日文, 73

日期格式, 27

北非, 55

北美, 56

北歐, 56

字元集, 30

西歐, 58

完整, 24, 51

亞洲, 52

定義, 23

東歐, 54

南美, 56

南歐, 57

時間格式, 26

紙張大小, 33

排序, 30

貨幣, 21

貨幣格式, 28

部分, 24, 51

種類, 25

數字格式, 27

澳大拉西亞, 53

環境變數, 140

鍵盤差異, 33

語言環境環境變數, 140

寬

寬字元, 支援, 40-47

德

德文, 字元支援, 109

德文鍵盤, 66

複

複合文字版面配置 (CTL), 14, 149-167

Motif, 151-159

Motif 程式庫, 166-167

複合文字版面配置 (CTL) (續)

- XOC 資源, 150
- XOM, 150
- 分音符, 149-167
- 文字方向, 149-167
- 文字資源, 166
- 水平定位點, 164-165
- 版面配置方向, 160-163
- 建立 render 表格, 163-164
- 建立 Rendition, 161
- 架構, 150
- 連音符, 149-167
- 滑鼠選取, 165
- 編輯 Rendition, 162
- 鍵盤選取, 166

輸

輸入法

- ATOK 方法, 75
- 日文, 75
- 印度文, 81
- 印度文方法, 77
- 泰文, 105

輸入模式

- en_US.UTF-8 語言環境, 110
- Unicode 十六進位, 139
- 日文, 137
- 希伯來文, 136
- 希臘文, 129, 130
- 阿拉伯文, 128
- 查表, 139
- 英文, 113
- 斯拉夫文, 128
- 繁體中文, 138
- 韓文, 137
- 簡體中文, 138

靜

- 靜態連結, 38-39

應

應用程式

- FontSet/XmFontList 定義, 147
- XPG4, 40
- 連結至系統程式庫, 38-39

繁

繁體中文

- 字元支援, 109
- 輸入模式, 138
- 繁體中文鍵盤, 69

鍵

鍵盤

- CTL 選取, 166
- Kedmanee, 104
- Kedmanee 增強型, 104
- Pattajoti, 105
- Type 4、5 和 5c, 61
- type 6, 60
- 土耳其文 F, 70
- 土耳其文 Q, 70
- 丹麥文, 65
- 日文, 66, 137
- 比利時文, 64
- 卡納達文, 79
- 古吉拉特文, 78
- 在 SPARC 上變更鍵盤設定, 63
- 西班牙文, 68
- 希伯來文, 136
- 希臘文 Euro, 130
- 希臘文 UNIX, 130
- 坦米爾文, 80
- 孟加拉文, 78
- 果魯穆其文, 79
- 法文, 65
- 芬蘭文, 65
- 阿拉伯文, 63, 128
- 美國, 71
- 美國/UNIX, 71
- 英國, 70
- 挪威文, 67
- 特拉古文, 80
- 配置, 63, 78, 104

鍵盤 (續)

- 區域, 59
- 梵文, 78
- 荷蘭 (荷蘭文), 67
- 喀拉拉文, 80
- 斯拉夫文, 64
- 斯拉夫文 (俄文), 129
- 瑞士 (法文), 69
- 瑞士 (德文), 69
- 瑞典文, 68
- 義大利文, 66
- 葡萄牙文, 68
- 德文, 66
- 繁體中文, 69
- 韓文, 67, 138
- 變更為捷克文配置, 63
- 變更配置, 60

韓

韓文

- Hangul, 31
 - Hanja, 31
 - 字元支援, 109
 - 輸入模式, 137
- 韓文鍵盤, 67

簡

簡體中文

- 字元支援, 109
- 輸入模式, 138

轉

轉換

- Unicode iconv 模組, 185-206
- 使用者自行定義字碼集, 48

