



국제 언어 환경 설명서

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

부품 번호: 819-0401-10
2005년 1월

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 모든 권리는 저작권자의 소유입니다.

이 제품 또는 문서는 저작권에 의해 보호되고 사용권에 따라 사용, 복사, 배포 및 디컴파일은 제한됩니다. 이 제품이나 문서의 어떤 부분도 Sun 및 그 라이선스 허여자의 사전 서면 승인 없이 어떤 형태로든 어떤 수단을 통해서든 복제해서는 안 됩니다. 글꼴 기술을 포함한 타사 소프트웨어에 대한 저작권 및 사용권은 Sun 공급업체에 있습니다.

이 제품의 일부는 University of California에서 승인된 Berkeley BSD 시스템에 기초합니다. UNIX는 미국 및 다른 국가에서 X/Open Company, Ltd.를 통해 독점적으로 사용권이 부여되는 등록 상표입니다.

Sun, Sun Microsystems, Sun 로고, docs.sun.com, AnswerBook, AnswerBook2, SunOS는 미국 및 기타 국가에서 Sun Microsystems, Inc.의 상표 또는 등록 상표입니다. 및 Solaris는 미국 및 기타 국가에서의 Sun Microsystems, Inc. 등록 상표, 상표 또는 서비스 상표입니다. 모든 SPARC 상표는 사용 허가를 받았으며 미국 및 다른 국가에서 SPARC International, Inc.의 상표 또는 등록 상표입니다. SPARC 상표를 사용하는 제품은 Sun Microsystems, Inc.가 개발한 구조를 기반으로 하고 있습니다. Netscape Navigator는 미국 및 다른 국가에서 Netscape Communications Corporation의 상표 또는 등록 상표입니다. PostScript는 일부 관할 지역에 등록된 Adobe Systems, Incorporated의 상표 또는 등록 상표입니다.

Sun Microsystems, Inc.는 사용자 및 사용 허가자를 위해 OPEN LOOK 및 Sun™ GUI(그래픽 사용자 인터페이스)를 개발했습니다. Sun은 컴퓨터 업계를 위한 시각적 또는 그래픽 사용자 인터페이스(GUI)의 개념을 연구 개발한 Xerox사의 선구적인 노력을 높이 평가하고 있습니다. Sun은 Xerox 및 Xerox 그래픽 사용자 인터페이스(GUI)의 비독점적 라이선스를 가지며, 이 라이선스는 OPEN LOOK GUI를 구현하고 Sun의 서면 라이선스 계약을 준수하는 Sun 정식 사용자에게 적용됩니다.

연방 정부 취득: 상용 소프트웨어-정부 사용자는 표준 사용권 및 규정을 준수해야 합니다.

설명서는 "있는 그대로" 제공되며, 상품성, 특정 용도에 대한 적합성 또는 비침해에 대한 묵시적인 보증을 비롯한 일체의 명시적 또는 묵시적 조건, 표현 및 보증 책임이 없습니다. 단, 이러한 내용이 법적으로 무효한 경우는 제외합니다.



041012@10082



목차

머리말 13

1 Solaris 국제화 개요	17
Solaris 국제화 아키텍처 정보	17
새로운 국제화 및 지역화 기능	18
국제화 및 지역화의 개요	21
국제화 기본 절차	21
Solaris 인터페이스의 지역화 기능	22
로켈이란?	23
C 로켈 - 기본 로켈	24
전체 및 부분 로켈	24
로켈의 영향을 받는 동작	25
로켈 범주	25
지역화용 로켈 범주 사용	26
시간 형식	26
날짜 형식	27
숫자 형식	27
국제적 통화 형식	28
언어 단어 및 글자 차이	30
단어 분리자	30
정렬 순서	31
문자 세트	31
키보드 차이점	33
용지 크기의 차이	34

2	일반 국제화 기능	35
	코드 세트 독립 지원	35
	CSI 접근 방법	36
	CSI화 된 명령	36
	CSI 활성화 라이브러리	37
	로켈 데이터베이스	37
	프로세스 코드 형식	38
	멀티바이트 지원 환경	38
	동적으로 링크된 응용프로그램	39
	변경된 인터페이스	39
	ctype 매크로	40
	libc의 국제화 API	41
	genmsg 유틸리티	48
	사용자 정의 및 사용자 확장 가능 코드 변환	49
	국제화된 도메인 이름(IDN) 지원	49
3	Solaris 환경에서 지역화	53
	지역화를 위한 소프트웨어 지원	53
	Solaris 로켈 패키지 요약	53
	지원되는 로켈	54
	로켈용 복수 키 구성 시퀀스	60
	Solaris 환경에서의 키보드 지원	61
	SPARC 시스템에서 키보드 변경하기	63
	Intel 시스템에서 키보드 변경하기	66
	키보드 레이아웃 도표	66
	새로운 Solaris 키보드 소프트웨어 지원	74
	▼ 에스토니아어 유형 6 USB 키보드 지원에 액세스하는 방법	74
	▼ 캐나다 프랑스어 유형 6 USB 키보드 지원에 액세스하는 방법	74
	▼ 폴란드 프로그래머 유형 5 키보드 지원에 액세스하는 방법	75
4	지원되는 아시아권 로켈	77
	일본어 지역화	77
	일본어 로켈	77
	일본어 문자 세트	77
	일본어 글꼴	78
	일본어 입력 시스템	79
	▼ ATOK 입력 방법을 사용하는 방법	79

일본어 단말기용 단말기 설정	79
일본어 iconv 모듈	80
사용자 정의 문자 지원	80
부분 로케일과 전체 로케일의 차이	80
인도어 지역화	81
▼인도어 입력 방법을 사용하는 방법	81
인도어 키보드	82
매핑의 이해	85
연속 음성 기반 입력 방법에 대한 매핑	85
연속 음성 입력 방법이 작동하는 방법	107
태국어 지역화	108
태국어 입력 방법	108
태국어 키보드 레이아웃	108
태국어 입력 방식 보조 창	110
5 UTF-8 로케일 지원의 개요	111
유니코드의 개요	111
유니코드 로케일: en_US.UTF-8 지원	112
데스크탑 입력 방법에 대하여	114
스크립트 선택 및 입력 모드	115
입력 방식 액세스하기	115
입력 모드 스위치 키 시퀀스	116
영어/유럽어 입력 모드	117
아랍어 입력 모드	132
키릴 문자 입력 모드	133
그리스어 입력 모드	134
히브리어 입력 모드	142
일본어 입력 모드	143
한국어 입력 모드	144
간체 한자 입력 모드	144
정체 한자 입력 모드	144
정체 한자(홍콩) 입력 모드	145
유니코드 16진법 입력 모드	145
테이블 조회 입력 모드	146
시스템 환경	146
로케일 환경 변수	146
TTY 환경 설정	147
코드 변환	150

DtMail 지원	151
프로그래밍 환경	154
X 응용프로그램과 함께 사용되는 FontSet	154
CDE/Motif 응용프로그램의 글꼴 목록 정의	154
6 Complex Text Layout	157
CTL 기술의 개요	157
CTL 구조의 개요	158
X 라이브러리 기반 응용프로그램에 대한 CTL 지원	158
XOC 자원	159
CTL 기술 지원을 위한 Motif 변경 사항	159
XmNlayoutDirection 자원	160
XmStringDirection 자원	161
XmRendition 자원	161
XmText and XmTextField 자원	162
XmTextFieldGetLayoutModifier 자원	166
XmTextGetLayoutModifier 자원	166
XmTextFieldSetLayoutModifier 자원	167
XmTextSetLayoutModifier 자원	167
XmStringDirectionCreate 자원	167
UIL 인자	168
CTL 응용프로그램 개발하기	168
레이아웃 방향 제어	168
자원 파일에서 렌더 테이블 만들기	172
수평 탭	173
마우스 선택	174
키보드 선택	175
텍스트 자원 및 기하학	175
이식 관련 지침	176
7 mp를 사용한 인쇄 필터 강화	177
UTF-8에 대한 인쇄	177
mp 인쇄 필터 강화 개요	178
로컬 특정 글꼴 구성 파일 mp.conf와 함께 mp 사용하기	178
로컬 특정 포스트스크립트 프롤로그 파일과 함께 mp 사용하기	179
Xprt(X 인쇄 서버) 클라이언트로 mp 사용하기	179
mp.conf구성 파일을 사용하는 지역화	180

- ▼ 프린터 상주 글꼴을 추가하는 방법 184
- ▼ 공유 객체 파일을 작성하는 방법 185
- 프롤로그 파일 추가 및 사용자 정의 186
 - 포스트스크립트 파일 사용자 정의 186
 - .xpr Files 189

A iconv 코드 변환 195

- 색인 223

표

표 1-1	법정 UTF-8 바이트 시퀀스	20
표 1-2	국제 시간 형식	26
표 1-3	국제 날짜 형식	27
표 1-4	국제 숫자 규약	28
표 1-5	국제 통화 규약	28
표 1-6	유로 통화를 지원하는 사용자 로케일	29
표 1-7	독일어 로케일과 관련 LC_MONETARY 피연산자	30
표 1-8	공통 국제 페이지 크기	34
표 2-1	메시징 libc의 기능	41
표 2-2	코드 libc의 변환	41
표 2-3	정규 libc의 표현식	42
표 2-4	넓은 libc의 문자 클래스	42
표 2-5	libc의 로케일 수정 및 질의	42
표 2-6	libc의 로케일 데이터 질의	42
표 2-7	libc의 문자 분류 및 음역	43
표 2-8	libc의 문자 조합	44
표 2-9	libc의 통화 형식	44
표 2-10	libc의 날짜 및 시간 형식	44
표 2-11	멀티바이트 libc의 처리	45
표 2-12	libc의 넓은 문자 및 문자열 처리	45
표 2-13	libc의 형식화된 넓은 문자 입력 및 출력	46
표 2-14	libc의 넓은 문자열	47
표 2-15	libc의 넓은 문자 입력 및 출력	47
표 2-16	iconv 코드 변환	50
표 3-1	아시아 로케일	54
표 3-2	오스트랄라시아 로케일	55
표 3-3	중앙 아메리카 로케일	56

표 3-4	중앙 유럽 로케일	56
표 3-5	동유럽 로케일	56
표 3-6	중동 로케일	57
표 3-7	북 아프리카 로케일	57
표 3-8	북미 로케일	58
표 3-9	북부 유럽 로케일	58
표 3-10	남미 로케일	59
표 3-11	남유럽 로케일	59
표 3-12	서유럽 로케일	60
표 3-13	Compose 키로 만든 구분 기호	60
표 3-14	지역 키보드 지원	61
표 3-15	유형 4, 5 및 5c 키보드용 레이아웃	63
표 4-1	일본어 비트맵 글꼴	78
표 4-2	일본어 트루 타입 글꼴	79
표 5-1	입력 모드 스위치 키 시퀀스	117
표 5-2	공통 Latin-1 Compose 키 시퀀스	118
표 5-3	공통 Latin-2 Compose 키 시퀀스	122
표 5-4	공통 라틴-3 Compose 키 시퀀스	123
표 5-5	공통 Latin-4 Compose 키 시퀀스	124
표 5-6	공통 Latin-5 Compose 키 시퀀스	125
표 5-7	공통 라틴-9 Compose 키 시퀀스	126
표 5-8	악센트 사용 불능 키 기반 Compose 키 시퀀스	126
표 5-9	그리스어 입력 모드의 Compose 키 시퀀스	135
표 5-10	3개 키를 통한 그리스어 Compose 키 시퀀스 입력 모드	141
표 5-11	4개 키를 통한 그리스어 Compose 키 시퀀스 입력 모드	142
표 5-12	en_US.UTF-8이 지원하는 STREAMS 모듈	147
표 5-13	en_US.UTF-8이 지원하는 64비트 STREAMS 모듈	147
표 6-1	XmRendition의 새로운 자원	161
표 6-2	Xm CTL의 새로운 자원	163
표 6-3	UIL	168
표 7-1	선택적 키워드/값 쌍	183
표 7-2	STARTCOMMON/ENDCOMMON 키워드 값	192
표 A-1	사용 가능한 유니코드 관련 iconv 코드 변환 모듈	195
표 A-2	사용 가능한 유니코드 및 IBM/Microsoft EBCDIC 및 PC 코드 페이지 관련 iconv 코드 변환 모듈	217
표 A-3	UTF-8에 대한 사용 가능한 iconv 코드 변환 - IBM 및 Microsoft EBCDIC/PC 코드 페이지	219

그림

그림 1-1	Solaris Operating System에서 로캘의 기능 및 구조	21
그림 2-1	IDN 대 ACE 변환	49
그림 2-2	ACE 대 IDN 변환	50
그림 3-1	아랍어 키보드	66
그림 3-2	벨기에 키보드	66
그림 3-3	키릴어(러시아어) 키보드	66
그림 3-4	덴마크어 키보드	67
그림 3-5	핀란드어 키보드	67
그림 3-6	프랑스어 키보드	67
그림 3-7	독일어 키보드	68
그림 3-8	이탈리아어 키보드	68
그림 3-9	일본어 키보드	68
그림 3-10	한국어 키보드	69
그림 3-11	네덜란드 (네덜란드어) 키보드	69
그림 3-12	노르웨이어 키보드	69
그림 3-13	포르투갈어 키보드	70
그림 3-14	스페인어 키보드	70
그림 3-15	스웨덴어 키보드	70
그림 3-16	스위스 (프랑스어) 키보드	71
그림 3-17	스위스 (독일어) 키보드	71
그림 3-18	정체 한자 키보드	71
그림 3-19	터키어 F 키보드	72
그림 3-20	터키어 Q 키보드	72
그림 3-21	영국 키보드	72
그림 3-22	미국 키보드	73
그림 3-23	미국/UNIX 키보드	73
그림 4-1	Bengali 자음용 맵	86

그림 4-2	Bengali 모음용 맵	86
그림 4-3	Bengali 기타용 맵	87
그림 4-4	Gujarati 자음용 맵	88
그림 4-5	Gujarati 모음용 맵	89
그림 4-6	Gujarati 기타용 맵	90
그림 4-7	Gurmukhi 자음용 맵	91
그림 4-8	Gurmukhi 모음용 맵	92
그림 4-9	Gurmukhi 기타용 맵	93
그림 4-10	힌두어 자음용 맵	94
그림 4-11	힌두어 모음용 맵	94
그림 4-12	힌두어 기타용 맵	95
그림 4-13	Kannada 자음용 맵	96
그림 4-14	Kannada 모음용 맵	97
그림 4-15	Kannada 기타용 맵	98
그림 4-16	Malayalam 자음용 맵	99
그림 4-17	Malayalam 모음용 맵	100
그림 4-18	Malayalam 기타용 맵	101
그림 4-19	타밀어 자음용 맵	102
그림 4-20	타밀어 모음용 맵	103
그림 4-21	Telugu 자음용 맵	105
그림 4-22	Telugu 모음용 맵	105
그림 4-23	Telugu 기타용 맵	106
그림 5-1	입력 모드 선택 창	115
그림 5-2	아랍어 키보드	132
그림 5-3	키릴 문자(러시아) 키보드	133
그림 5-4	그리스 유럽 키보드	134
그림 5-5	그리스어 UNIX 키보드	134
그림 5-6	히브리어 키보드	142
그림 5-7	일본어 키보드	143
그림 5-8	한국어 키보드	144
그림 5-9	DtMail 새 메시지 창	152
그림 6-1	CTL 구조	158
그림 6-2	레이아웃 방향	169
그림 6-3	탭 이동 동작	173

머리말

국제 언어 환경 설명서는 Solaris™ 운영 체제 (Solaris OS)에 새롭게 도입된 국제화 기능에 대하여 소개합니다. 다양한 언어와 문화적 규약을 지원하는 국제적인 소프트웨어 제품을 제작하기 위한 현재 Solaris 릴리스의 사용법에 관한 중요한 정보가 포함되어 있습니다.

또한, 설명서는 이 릴리스의 국제화 기능에 대한 추가 정보를 포함하는 다른 문서에 대한 포인터를 제공합니다.

주 - 이 설명서의 운영 체제와 관련된 모든 정보는 Solaris 운영 체제와 관련이 있습니다.

이 머리말은 다음 절로 구성됩니다.

- 14 페이지 “이 책에 대하여”
- 14 페이지 “설명서 구성”
- 14 페이지 “관련 서적과 사이트”
- 15 페이지 “Sun 설명서 온라인 액세스”
- 15 페이지 “표기 규칙”
- 16 페이지 “명령에 나오는 셸 프롬프트의 예”

주 - 이 Solaris 릴리스는 SPARC® 및 x86 프로세스 아키텍처 제품군을 사용하는 시스템을 지원합니다.: UltraSPARC®, SPARC64, AMD64, Pentium, 및 Xeon EM64T. 지원되는 시스템은 <http://www.sun.com/bigadmin/hcl>의 *Solaris 10 Hardware Compatibility List*를 참조하십시오. 이 설명서는 플랫폼 유형간의 어떤 구현을 인용합니다.

이 설명서 안에서 용어 “x86”는 AMD64 또는 Intel Xeon/Pentium 제품군과 호환되는 프로세서를 사용한 64-bit 및 32-bit 시스템을 참조합니다. 지원되는 시스템은 *Solaris 10 Hardware Compatibility List*를 참조하십시오.

이 책에 대하여

이 설명서는 현재 Solaris 운영 체제의 국제적 응용 프로그램을 설계 및 지원하는 소프트웨어 개발자와 소프트웨어 관리자를 대상으로 합니다.

설명서는 독자가 C 프로그래밍 언어에 대한 실제적인 지식을 가지고 있다고 가정합니다.

설명서 구성

본 설명서의 각 장은 다음과 같이 구성되어 있습니다:

- 1 장은 현재 Solaris 릴리스에서 사용 가능한 새로운 국제화 및 지역화 기능에 대하여 설명합니다.
- 2 장은 코드 세트 독립(CSI), 로캘 데이터베이스, libc API 및 기타 국제화 기능에 대한 소개 정보를 제공합니다.
- 3 장은 현재 Solaris 운영 체제에서 사용하기 위해 지원되는 로캘, 글꼴 및 키보드에 대한 정보를 제공합니다.
- 4 장은 현재 Solaris 릴리스에서 제공되는 일본어, 힌디어 및 태국어 지역화 지원에 대하여 설명합니다.
- 5 장은 현재 Solaris 운영 체제에서 사용하기 위해 지원되는 사용 가능 입력 방법 및 코드 변환 기능에 대한 정보를 제공합니다.
- 6 장은 Complex Text Layout(CTL) 확장으로 Motif APIs가 논리적이고 물리적인 텍스트 표현 간의 복잡한 변형을 필요로 하는 쓰기 시스템을 지원할 수 있습니다. 복합 변형이 필요한 쓰기 체계는 아랍어, 히브리어 및 태국어입니다.
- 7 장은 mp 인쇄 필터에 대한 특정한 강조를 가진 인쇄 지원에 대하여 설명합니다.
- 부록 A에는 사용 가능한 iconv 변환 표가 있습니다..

관련 서적과 사이트

다음 책은 이 설명서에 설명하는 항목에 대한 추가 정보를 제공합니다.

- Solaris 국제화:
Tuthill, Bill 및 David Smallberg. *Creating Worldwide Software: Solaris International Developer's Guide* 제2판 Mountain View, California, Sun Microsystems Press, 1997.
이 책은 books@sun.com 및 <http://www.sun.com/books/>을 통해 사용 가능합니다.

니다. 이 책은 Solaris Operating System의 국제화 프로세스의 일반적 개요를 제공합니다.

- Solaris 공통 데스크탑 환경:
Solaris Common Desktop Environment: Programmer's Guide 또한 Solaris Documentation CD에 포함되어 있는 CDE 개발자 모음의 일부입니다.
- 중국어 및 한국어 Solaris 로컬:
 한국어 *Solaris 사용 설명서*
Simplified Chinese Solaris User's Guide
Traditional Chinese Solaris User's Guide
- OSF/Motif 응용 프로그램 개발:
OSF/Motif Programmer's Guide, Release 1.2, Englewood Cliffs, New Jersey, Prentice-Hall, 1993. 이 책은 OSF/Motif 응용 프로그램 프로그래밍 인터페이스를 사용하여 Motif 응용 프로그램을 작성하는 방법에 대한 Open Software Foundations (OSF) 설명서입니다.

Sun 설명서 온라인 액세스

docs.sun.comSM 웹 사이트에서 Sun 기술 관련 문서를 온라인으로 이용할 수 있습니다. 다음 주소에서 docs.sun.com 아카이브를 살펴보고 특정 서적 제목이나 주제에 대해 검색할 수 있습니다. URL은 <http://docs.sun.com>입니다.

표기 규칙

다음 표는 이 책에서 사용된 서체 변경 사항에 대하여 설명합니다.

표 P-1 표기 규칙

서체 또는 기호	의미	예
AaBbCc123	명령, 파일 및 디렉토리의 이름 등 컴퓨터 화면에 출력되는 내용입니다.	.login 파일을 편집하십시오. 모든 파일을 표시하려면 <code>ls -a</code> 명령을 사용하십시오. <code>machine_name% you have mail.</code>

표 P-1 표기 규칙 (계속)

서체 또는 기호	의미	예
AaBbCc123	화면 상의 컴퓨터 출력과는 반대로 사용자가 직접 입력하는 사항입니다.	machine_name% su Password:
<i>AaBbCc123</i>	명령줄에서 입력 위치 표시: 실제 이름이나 값으로 대체됩니다.	파일을 삭제하려면 rm <i>filename</i> 을 입력하십시오.
<i>AaBbCc123</i>	책 제목, 새로 나오는 단어나 용어, 강조 표시할 단어입니다.	사용자 설명서 의 6장을 읽으십시오. 이를 클래스 옵션이라고 합니다. 이 작업을 수행하려면 루트 여야 합니다.

명령에 나오는 셸 프롬프트의 예

다음 표는 C 셸, Bourne 셸 및 Korn 셸의 기본 시스템 프롬프트와 슈퍼유저 프롬프트입니다.

표 P-2 셸 프롬프트

셸	프롬프트
C 셸 프롬프트	machine_name%
C 셸 슈퍼유저 프롬프트	machine_name#
Bourne 셸 및 Korn 셸 프롬프트	\$
Bourne 셸 및 Korn 셸 슈퍼유저 프롬프트	#

Solaris 국제화 개요

이 장은 Solaris 국제화 및 지역화의 새로운 기능과 핵심 개념을 소개합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 18 페이지 “새로운 국제화 및 지역화 기능”
- 21 페이지 “국제화 및 지역화의 개요”
- 23 페이지 “로켈이란?”
- 26 페이지 “지역화용 로켈 범주 사용”
- 30 페이지 “언어 단어 및 글자 차이”
- 33 페이지 “키보드 차이점”
- 34 페이지 “용지 크기의 차이”

Solaris 국제화 아키텍처 정보

현재 Solaris 릴리스는 UTF-8 로켈에 대한 유니코드 4.0 지원, 향상된 키보드 지원 및 몇 가지 mp 인쇄 필터 향상을 포함하는 여러 가지 새로운 기능을 포함합니다.

Solaris 국제화 아키텍처는 응용 프로그램 및 전세계 언어 서비스를 위한 개발, 배치 및 관리를 용이하게 합니다. 단일한 다국어 제품이 39가지 언어와 162개 로켈 지원을 제공합니다. 또한, 태국어와 힌두어 스크립트에 대해 필요한 복잡한 텍스트 레이아웃에 대한 지원이 사용 가능합니다. 양방향 텍스트 기능이 아랍어와 히브리어에 대해서도 지원됩니다.

입력 방법, 문자 세트, 코드 세트 변환 및 기타 언어 관련 기능이 여러 Solaris 로켈에 대하여 지원됩니다. 표준 API에 따라 여러 언어 환경에서 응용 프로그램을 배치할 수 있습니다. Solaris 환경에서 언어 속성을 사용자 정의하고 변환기 테이블을 변경하거나 새로운 입력 방법 편집기를 사용자 정의할 수도 있습니다.

Solaris X 세계화 프레임워크용 소스 코드가 2000년 가을에 개방형 소스 커뮤니티로 릴리스되었습니다. 해당 릴리스를 사용하여 공통 참조 구현에 따라 국제적 응용 프로그램의 호환성과 상호 운영성을 향상할 수 있습니다. 세계화에 대한 코드 세트 독립적 접근

방법을 통해 여러 토착 언어와 유니코드 로케에서 작업할 수 있습니다. Solaris 프레임워크는 플랫폼에 걸친 확장 기능을 제공합니다. 풍부한 데이터 변환기 세트가 여러 코드화 와 다양한 제3의 플랫폼 사이의 상호 운영성을 보장합니다.

또한 Solaris 플랫폼은 다국적 기업들의 전 세계 서버 관리 확장을 도와줍니다. 경쟁 플랫폼들과 달리 Solaris 플랫폼은 언어 서비스 관리에 서비스 기반 접근 방식을 사용합니다. 서버 관리자들은 클라이언트 시스템에 관계 없이 전세계 네트워크를 통해 언어 서비스를 활성화시킬 수 있습니다. 클라이언트 독립적 접근 방식은 클라이언트 응용프로그램을 변경하지 않아도 시스템을 업그레이드 가능하게 합니다. 예를 들어, 사용자는 파리의 인터넷 카페에서 아랍어로 전자 우편을 읽기 위해 로컬 클라이언트 응용 프로그램을 변경하지 않아도 됩니다.

새로운 국제화 및 지역화 기능

다음 새로운 기능이 현재 Solaris 릴리스에서 사용 가능합니다.

- 자동 코드화 파인더

자동 코드화 파인더는 국제적 문자 처리용 유틸리티입니다. 일반 용도 인터페이스를 통해 자동 코드화 파인더가 특정한 파일 또는 문자열의 코드화를 쉽게 감지하는 방법을 제공합니다. 코드화 감지가 다양한 언어 문자 인코딩에 대한 액세스를 간소화합니다. 자세한 내용은 `auto_ef(1)` 또는 `libauto_ef(3LIB)` 설명서 페이지를 참조하십시오.

- 로케 관리자

로케일 관리자를 사용하면 명령줄 인터페이스를 통하여 Solaris OS용 로케일을 확인하고 구성할 수 있습니다. `localeadm(1M)` 도구를 사용하여 시스템에 설치되었거나 특정한 장치 또는 디렉토리에 상주하는 로케일 패키지에 대한 정보를 표시할 수 있습니다. 각 지역마다 현재 시스템에 로케일을 추가하거나 제거할 수 있습니다. 예를 들어 현재 시스템에 동유럽 지역의 로케일을 모두 추가할 수 있습니다.

로케일 관리자를 도입하기에 앞서 시스템이 설치되면 시스템의 로케일을 변경하기 위해 개별 패키지를 추가/제거해야 합니다. 개발 패키지 작업은 패키지를 놓치거나 간과하기 쉽기 때문에 오류가 발생하는 경향이 있습니다.

로케일 관리자는 Solaris 설치 프로그램에서 로케일 선택 로직을 보완하는 것입니다. 설치 프로그램은 여전히 Solaris 로케일을 올바르게 설치하기 위한 기본 응용 프로그램입니다.

- mp 향상

Solaris 9 운영 체제와 함께 최초로 출시된 mp 인쇄 필터는 현재 Solaris 릴리스의 `xutops` 인쇄 필터를 대체합니다. mp 인쇄 필터는 이번 릴리스에서 다음 주요한 향상으로 향상되었습니다

- 구성된 글꼴이 `mp.conf` 파일에 없을 경우 mp 프로그램이 인식되지 않는 글꼴을 사용하는 그림 문자를 만날 때까지 계속 실행합니다.
- 트루타입 그림 문자를 인쇄할 때 mp 출력의 크기가 현저하게 줄어듭니다.
- mp이 채용하는 사전 메커니즘은 보다 빠른 인쇄를 위해 미세 조정됩니다.

- 트루타입 엔진이 다양한 폭의 모든 공간 문자를 처리하도록 향상되었습니다.

주 - xutops 인쇄 필터는 Solaris Operating System에서 더 이상 지원되지 않습니다. xutops 인쇄 필터는 이전에 UTF-8 로캘의 국제화된 텍스트를 인쇄하는 데 사용되었습니다. xutops를 대체하는 mp 인쇄 필터는 xutops 인쇄 필터의 지원되는 기능의 슈퍼 세트입니다. 자세한 내용은 mp(1) 설명서 페이지를 참조하십시오.

- 새로운 유럽 키보드 지원

Sun I/O 키보드는 폴란드어 프로그래머 유형 5 키보드와 Sun Ray™ USB 유형 6 러시아어, 에스토니아어 및 캐나다 프랑스어 키보드에 대해 사용 가능합니다.

주 - 현재 이 새로운 유럽 키보드 유형에 사용할 수 있는 하드웨어는 없습니다. 새로운 키보드 소프트웨어를 사용하려면 74 페이지 “새로운 Solaris 키보드 소프트웨어 지원”의 절차를 참조하십시오.

- 유니코드 4.0 지원

다음 UTF-8 로캘은 유니코드 표준의 4.0 버전을 지원하도록 업데이트되었습니다.

- 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

새로운 표준 버전은 추가로 1,226가지 새로운 문자를 도입하고 다양한 규범적이고 정보적인 변경 사항을 포함합니다.

유니코드 3.2는 “UTF-8 Corrigendum”으로서 보다 엄격한 UTF-8 바이트 시퀀스를 정의합니다

표 1-1 법정 UTF-8 바이트 시퀀스

코드 지점	첫번째 바이트	두번째 바이트	세번째 바이트	네번째 바이트
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	ill-formed			
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

이러한 시퀀스는 U+D800과 U+DFFF 사이의 대응 코드 지점을 제외합니다. 시퀀스는 모든 기타 부적합 바이트 값도 금지합니다. 새로운 정의를 준수하기 위해 유니코드 로캘 방법과 UTF-8 iconv 모듈이 새로 정의된 UTF-8 부적합 바이트 시퀀스를 감지하도록 향상되었습니다.

■ 태국어 키보드 레이아웃

다음 키보드 레이아웃이 태국어 입력 방법에 대하여 지원됩니다.

- Kedmanee (TIS820-2531) 키보드 레이아웃. Kedmanee 레이아웃은 컴퓨터 키보드 가 아닌 타이프라이터용으로 디자인되었습니다. 타이프라이터 키보드의 키 수가 제한되어 레이아웃에서 일부 태국어 특수 문자를 사용할 수 없었습니다. TIS820-2531은 컴퓨터 키보드와 함께 사용하도록 Kedmanee 레이아웃을 채택했습니다.
- TIS820-2538 키보드 레이아웃. 이 향상된 Kedmanee 레이아웃은 원래 Kedmanee 레이아웃에서 사용할 수 없었던 일부 태국어 특수 문자를 포함하는 TIS820-2531 레이아웃의 업데이트 버전입니다. 현재, TIS820-2538은 태국 산업 표준 협회에서 발행한 유일한 태국어 키보드 레이아웃 표준입니다.
- Pattajoti 키보드 레이아웃. Pattajoti 레이아웃도 타이프라이터용으로 디자인되었지만 손가락 하중 배분이 보다 우수합니다. Pattajoti는 왕실 관개국 관리가 발명했고 아직까지 해당 부서에서 널리 사용되고 있습니다.
- 태국어 입력 방법에 대한 구성 가능한 키보드 레이아웃이자 사용자 정의 키보드 레이아웃.

■ 인도 언어에 대한 입력 방법 지원

중국어 입력 방법용으로 사용되는 것과 유사한 코드 테이블 입력 방법 인터페이스가 이번 릴리스에서 사용 가능합니다. IIMF SDK 및 SunIM 언어 인터페이스를 바탕으로 하는 인도어 입력 방법은 다음 새로운 기능을 제공합니다.

- 음성, 음역 기반 입력 방법 및 키보드 레이아웃. 지원되는 키보드 레이아웃은 ISCII 표준에서 INSCRIPT 키보드 레이아웃으로 정의됩니다.
- 표준 입력 방법 전환.
- 인도어 스크립트는 힌두어, 타밀어, Kannada, Malayalam, Telugu, Gujarati, 펀잡어 및 벵갈어를 포함합니다. F5 키를 눌러 입력 스크립트를 변경할 수 있습니다.
- 향후에 보다 쉽게 확장할 수 있도록 하는 새로운 키보드 레이아웃용 플러그인 메커니즘. 플러그인 구성 파일이 언어 엔진 모듈에 의해 로드됩니다.

국제화 및 지역화의 개요

국제화와 지역화는 서로 차이가 있습니다. **국제화**는 언어 또는 지역 사이에서 소프트웨어를 이식 가능하게 만드는 프로세스인 반면 **지역화**는 특정 언어 또는 지역용으로 소프트웨어를 적응시키는 프로세스입니다. 국제화된 소프트웨어는, 특정 문화의 요구에 맞도록 프로그램의 동작 방식을 변경시켜주는 인터페이스를 이용해 개발될 수 있습니다. 지역화는 **로케일**이라 부르는 언어나 지역을 지원하는 온라인 정보를 구축하는 것과 관련됩니다.

모국어 및 사용자가 달라지면 완전히 재작성해야 하는 소프트웨어와 달리 국제화된 소프트웨어는 재작성이 필요없습니다. 국제화된 소프트웨어는 하나의 로케일에서 다른 로케일로 아무런 변경없이 이식할 수 있습니다. Solaris 시스템은 국제화되어 있어 국제화된 소프트웨어를 작성하기 위해 필요한 인프라와 인터페이스를 제공합니다.

국제화 기본 절차

국제화된 응용프로그램의 실행 가능 이미지는 언어와 지역에 따라 이식 가능합니다. 소프트웨어를 국제화하려면 다음과 같이 하십시오.

- 이 책에서 설명하는 인터페이스를 사용하여, 동적으로 재컴파일함으로써 수정할 수 있는 환경을 가진 소프트웨어를 만듭니다.
- 소프트웨어를 사용자가 볼 수 있는 실행 코드와 모든 메시지로 분리합니다. 메시지 문자열은 메시지 카탈로그로 보관합니다.

메시지 문자열을 언어나 지역에 맞추어 번역합니다. **로케일**에는 메시지 스트링과 정렬 방식을 지정하는 메소드가 포함됩니다.

제품의 지역화된 버전을 사용하려면 사용자는 특정한 환경 변수를 설정합니다. 그러면 제품이 로케일 언어로 번역 가능한 메시지를 표시합니다. 날짜, 시간, 통화 및 기타 정보가 로케일 특정 규약에 따라 형식화 및 표시됩니다. 메시지 번역과 온라인 도움말 콘텐츠는 다음 도표에서 설명하는 바와 같이 서로 다른 계층에서 제공됩니다.

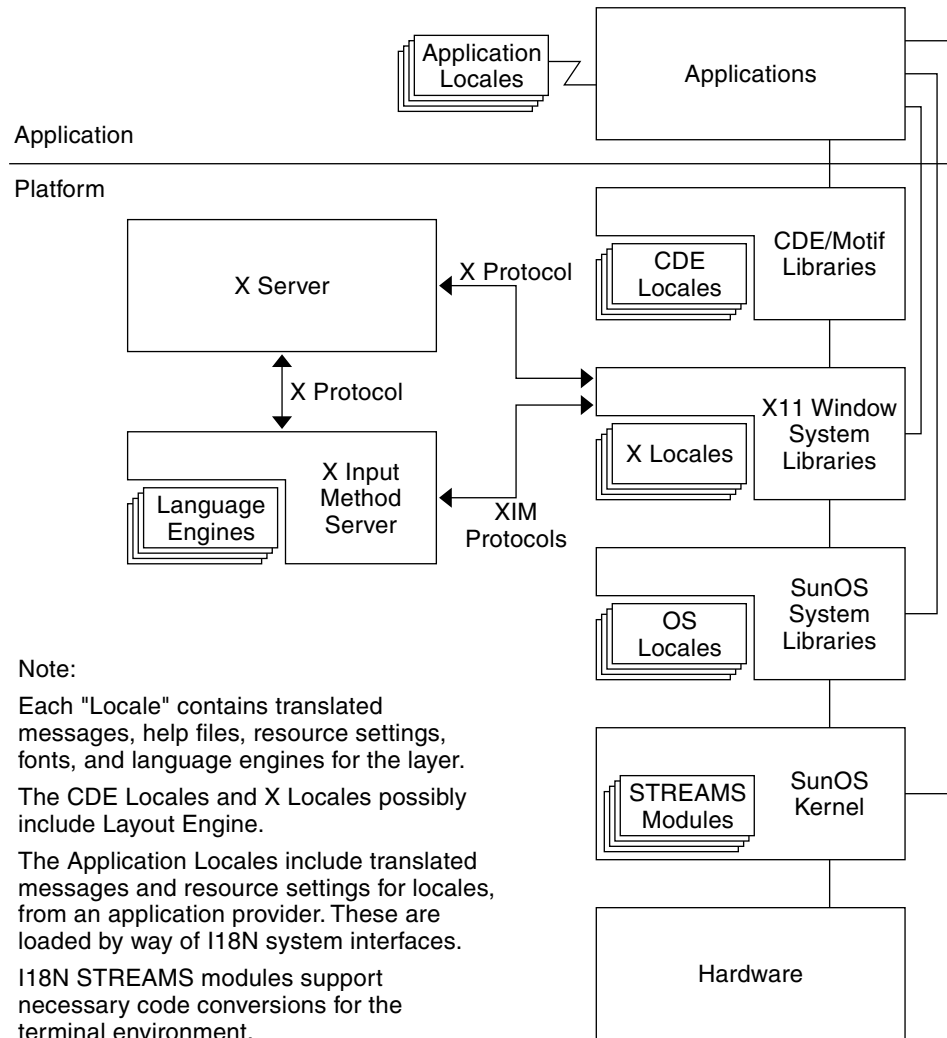


그림 1-1 Solaris Operating System에서 로캘의 기능 및 구조

Solaris 인터페이스의 지역화 기능

OS(운영 체제) 로캘 층은 기초 로캘 데이터베이스와, 응용프로그램의 런타임에 OS 시스템 인터페이스에 연결되는 함수들을 제공합니다. 응용프로그램들은 이 OS 로캘 모듈에 표준 API를 통해 액세스합니다.

X11 로켈 계층은 X 입력 방식 및 X 출력 방법에 인터페이스를 제공해 X11 응용프로그램이 로켈 텍스트를 입력 및 표시할 수 있게 합니다. 응용프로그램이 다양한 언어로 문자를 표시할 수 있게 하는 글꼴이 제공됩니다.

CDE/Motif는 X11 윈도우 시스템을 기반으로 설계되었습니다. 따라서, CDE/Motif는 API를 통해 X11 로켈 기능을 사용할 수 있습니다. Solaris 지역화는 대상 로켈에서 데스크탑이 작동되도록 하기 위해 다양한 로켈별 CDE 응용프로그램 구성을 갖고 있습니다. 메시지 번역과 온라인 도움말 콘텐츠는 서로 다른 계층에서 제공됩니다.

로켈이란?

응용 프로그램의 핵심 개념은 프로그램 **로켈**의 개념과 같습니다. 로켈은 토착 언어 환경의 명시적 모델이자 정의입니다. 로켈의 개념은 ANSI C 언어 표준 라이브러리 정의에 명확히 정의되어 있습니다.

로켈은 국가별로 다른 형식이나 기타 사양을 갖는 여러 개의 범주로 구성됩니다. 프로그램의 로켈은 해당 로켈의 코드 세트, 날짜 및 시간 형식 규약, 통화 규약, 10진수 형식 규약 및 조합(정렬) 순서 등을 정의합니다.

로켈은 기본 언어, 사용하는 국가(영토) 및 선택적 코드 세트로 구성될 수 있습니다. 대개의 경우 코드 세트가 포함됩니다. 예를 들어, 독일어는 de로서 Deutsch의 약어이고 스위스 독일어는 de_CH로서 이 때 CH는 Confederation Helvetica의 약어입니다. 이 규약으로 인해 통화 단위 표기법과 같이 국가별로 차이를 둘 수 있습니다.

둘 이상의 로켈을 특정 언어와 연결하여 지리적 차이를 둘 수 있습니다. 예를 들어, 미국에서 영어를 사용하는 사용자는 en_US 로켈(미국 영어)을 선택할 수 있고 영국의 영어 사용자는 en_GB(영국 영어)를 선택할 수 있습니다.

일반적으로 로켈 이름은 LANG 환경 변수에 의해 지정됩니다. 로켈 범주는 LANG에 종속되지만 별도로 설정할 수 있는데, 이 경우 LANG은 무시됩니다. LC_ALL 연산자가 설정되면 이것이 LANG 및 모든 별개의 로켈 범주를 대체합니다.

로켈 이름 지정 규약은 다음과 같습니다:

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

두 자의 *language* 코드는 ISO 639, 두 자의 *territory* 코드는 ISO 3166의 코드이며, *codeset*은 로켈에서 사용될 코드 세트 이름이고, *modifier*는 그 수정자(modifier)가 없는 로켈에 별도로 존재하는 특성의 이름입니다.

모든 Solaris 제품 로켈은 US-ASCII 코드 값과 함께 PCS(Portable Character Set) 문자를 보존합니다.

이식 가능한 문자 세트에 대한 자세한 정보는 “X/Open CAE Specification: System Interface Definitions, Issue 5” (ISBN 1-85912-186-1)을 참조하십시오.

단일 로케일이 하나 이상의 로케일 이름을 가질 수 있습니다. 예를 들어, POSIX는 C와 동일합니다.

C 로케일 - 기본 로케일

POSIX 로케일이라고도 하는 C 로케일은 모든 POSIX 준수 시스템에 대한 POSIX 시스템 기본 로케일입니다. Solaris 운영 체제는 POSIX 시스템입니다. Single UNIX Specification, Version 3이 C 로케일을 정의합니다. 다음에서 사양을 읽고 다운로드하기 위해 등록하십시오. <http://www.unix.org/version3/online.html>.

사용하는 국제화 프로그램이 다음 중 한 가지 방법으로 C 로케일에서 실행하도록 지정할 수 있습니다.

- 모든 로케일 환경 변수를 설정 해제합니다.

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

모든 로케일 환경 변수를 설정 해제합니다. C 로케일에서 응용 프로그램을 실행합니다.

- 로케일을 C 또는 POSIX로 명시적으로 설정합니다.

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

일부 응용 프로그램에서는 현재 로케일을 참조하기 위해 실제로 setlocale(3C)을 호출하지 않고 LANG 환경 변수를 확인합니다. 이 경우, setenv는 LC_ALL 및 LANG 로케일 환경 변수를 지정하여 C 로케일을 명시적으로 설정합니다. 로케일 환경 변수 사이의 선행 관계에 대해서는 setlocale(3C) 설명서 페이지를 참조하십시오.

단말기 환경에서 현재 로케일 설정을 확인하려면 locale(1) 명령을 실행하십시오.

```
system% locale
```

전체 및 부분 로케일

전체 Solaris 로케일은 관련 언어에서 나열된 모든 기능과 지역화된 시스템 메시지를 갖습니다. **부분 로케일**은 아무런 메시지도 설치되지 않습니다. Solaris 환경의 모든 로케일들은 관련 언어용 지역화된 메시지가 설치만 되어 있다면 지역화된 메시지를 표시할 수 있습니다. 예를 들어, 다음 로케일은 부분 또는 전체 로케일 중 하나입니다.

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

Language CD를 사용해 독일어 메시지 번역을 설치하면 위의 모든 로케일이 완전히 번역된 데스크탑에 액세스할 수 있어 **전체 로케일**이 됩니다. Languages CD에는 다음 언어 및 로케일용 메시지 번역이 포함되어 있습니다:

- 독일어
- 프랑스어
- 스페인어
- 스웨덴어
- 이탈리아어
- 일본어
- 한국어
- 간체 한자 로케일
- 번체 한자 로케일

모든 부분 로케일은 소프트웨어 CD에 포함되어 있습니다. 메시지 번역은 Language CD에 포함되어 있습니다.

모든 영어 로케일은 전체 로케일이며 소프트웨어 CD에 포함되어 있습니다.

로케일의 영향을 받는 동작

다양한 국가에서 숫자를 형식화하고 날짜 및 시간을 작성하고 단어와 구를 구분하거나 서면 또는 구두 자료를 인용하기 위해 대개 여러 가지 규약을 사용합니다. 로케일은 다른 지역에 대하여 다음 작업, 파일, 형식 및 표현식을 처리하는 방법을 결정합니다.

- 텍스트 데이터의 코드화 및 처리
- 자원 파일의 언어 식별 및 코드화
- 텍스트 문자열의 렌더링과 레이아웃
- 클라이언트 간 텍스트의 교환
- 선택한 스크립트의 코드 세트와 텍스트 처리 요구 사항을 충족하는 입력 방법 선택
- 문화적으로 다른 글꼴과 아이콘 파일
- 활동 및 파일 유형
- 사용자 인터페이스 정의(UI) 파일
- 날짜 및 시간 형식
- 숫자 형식
- 통화 형식
- 조합 순서
- 로케일에 따른 정규식 처리
- 정보 제공 메시지 및 진단 메시지와 대화식 응답용 형식

Solaris 환경은 언어와 문화에 따른 정보를 응용프로그램에서 분리하여 응용프로그램 밖에 저장합니다. 이렇게 함으로써 각 시장에 맞추어 응용프로그램을 번역, 재작성 및 재컴파일할 필요성을 제거합니다. 새로운 시장에 진출할 때 필요한 것은 외부 정보를 현지 언어 및 고객에 맞게 지역화하는 것 뿐입니다.

로케일 범주

로케일 범주는 다음과 같습니다.

LC_CTYPE 문자 처리 기능의 동작을 제어합니다.

LC_TIME	달과 요일 이름, 공통된 전체 표현 및 약식 표현을 포함한 날짜와 시간 형식을 지정합니다.
LC_MONETARY	로케용 통화 기호, 1000 구분자, 기호 위치, 분수 자리수 등을 포함한 통화 형식을 지정합니다.
LC_NUMERIC	10진수 분리자(또는 기수 문자), 1000 단위 구분자 및 그룹화를 지정합니다.
LC_COLLATE	조합 순서 및 로케용 정규 표현 정의를 지정합니다.
LC_MESSAGES	지역화된 메시지를 표시하는 언어, 로케의 긍정과 부정 응답(yes 및 no 문자열과 표현)을 지정합니다.
LO_LTYPE	언어 렌더링에 관한 정보를 제공하는 레이아웃 엔진을 지정합니다. 언어 렌더링(또는 텍스트 렌더링)은 스크립트의 모양과 방향 속성에 따라 다릅니다.

지역화용 로케 범주 사용

제품의 지역화는 대상 언어나 지역의 원 사용자의 자문을 얻어 수행해야 합니다. 특정한 정보 스타일 및 형식이 개발자에게 완벽하게 명백하고 보편적으로 보일 수 있습니다. 그러나 사용자에게는 그러한 형식이 어색하고 잘못되었거나 심지어 모욕적으로 보일 수 있습니다. 다음 절은 제품의 지역화 요구 사항을 충족하도록 사용자 정의할 수 있는 Solaris Operating System의 요소에 대하여 설명합니다.

시간 형식

다음 표는 다른로케이 11:59 PM을 작성하는 몇 가지 방법을 보여줍니다.

표 1-2 국제 시간 형식

로케	형식
캐나다어	23:59
핀란드	23.59
독일어	23.59 Uhr
노르웨이	23.59
태국어	23.59
영국 영어	23:59

시간은 12 시간 시계 및 24 시간 시계로 표현됩니다. 시간 및 분 구분자는 콜론 (:) 또는 마침표(. 참조).

시간대 분할이 국가 사이와 국가 내부에서 발생합니다. 시간대는 협정 세계시인 UTC (또는 그리니치 표준시인 GMT)에서 몇 시간 앞인지, 또는 뒤인지에 따라 설명할 수 있지만 이 숫자는 항상 정수인 것은 아닙니다. 예를 들어, 뉴펀들랜드는 인접 시간대와 30분 차이가 있는 시간대입니다.

일조 절약 시간(DST)은 국가마다 다른 날짜에서 시작하고 끝납니다. 많은 국가들은 아예 DST를 설정하지 않습니다. 또한, 일조 절약 시간은 시간대 내부에서 다양할 수 있습니다. 예를 들어, 미국에서는 구현이 국가 결정 사항입니다.

날짜 형식

다음 표는 세계 전역에서 사용되는 몇 가지 날짜 형식을 보여줍니다. 국가 안에서도 달라질 수 있습니다.

표 1-3 국제 날짜 형식

로케일	표시	예
캐나다(영어)	dd/mm/yy	24/08/01
덴마크	yyyy-mm-dd	2001-08-24
핀란드	dd.mm.yyyy	24.08.2001
프랑스어	dd/mm/yyyy	24/08/2001
독일어	yyyy-mm-dd	2001-08-24
이탈리아어	dd/mm/yy	24/08/01
노르웨이	dd-mm-yy	24-08-01
스페인어	dd-mm-yy	24-08-01
스웨덴어	yyyy-mm-dd	2001-08-24
영국	dd/mm/yy	24/08/01
미국	mm-dd-yy	08-24-01
태국어	dd/mm/yyyy	24/08/2001

숫자 형식

영국과 미국은 세계에서 십진수 자리를 표시하기 위해 마침표를 사용하는 몇 안 되는 국가에 속합니다. 그 밖의 많은 국가들은 마침표 대신 콤마를 사용합니다. 소수 구분 기호는 기수 문자라고도 부릅니다. 마찬가지로 영국과 미국은 콤마를 사용해 1000 단위 그룹을 구분하지만 다른 많은 국가들은 대신 마침표를 사용하고 일부 국가는 1000 단위를 좁은 공백으로 구분합니다.

로켈 지향 형식을 포함한 데이터 파일은 다른 로켈의 시스템으로 전달될 때 잘못 해석되는 경우가 종종 발생합니다. 예를 들어, 프랑스어 형식의 숫자를 포함하고 있는 파일은 영국 프로그램에는 적합하지 않습니다.

다음 표는 흔히 사용되는 숫자 형식 일부를 보여줍니다.

표 1-4 국제 숫자 규약

로켈	큰 숫자
캐나다(영어)	4,294,967.00
덴마크	4.294 967.295,00
핀란드	4 294 967 295,00
프랑스어	4 294 967 295,00
독일어	4,294,967.00
이탈리아어	4.294.967,00
노르웨이	4.294.967.295,00
스페인어	4.294.967.295,00
스웨덴어	4 294 967 295,00
영국	4,294,967,295.00
미국	4,294,967,295.00
태국어	4,294,967,295.00

주 - 목록의 숫자를 구분하는 방법을 지정하는 특별한 로켈 규약은 없습니다.

국제적 통화 형식

통화 단위와 표시 순서는 세계 전역에서 대단히 다양합니다. 현지 통화 기호와 국제적으로 통용되는 통화 기호가 서로 다를 수 있습니다. 다음 표는 일부 국가의 통화 형식을 보여줍니다.

표 1-5 국제 통화 규약

로켈	Currency	예
캐나다(영어)	달러 (\$)	\$1,234.56
캐나다(프랑스어)	달러 (\$)	1 234,56\$

표 1-5 국제 통화 규약 (계속)

로케일	Currency	예
덴마크	크로네(kr)	Kr 1.234,56
핀란드	Euro (€)	€ 1 234,56
프랑스어	Euro (€)	€ 1,234
일본어	Yen (¥)	¥ 1,234
노르웨이	크로네 (kr)	kr 1.234,56
스웨덴어	크로너 (Kr)	1 234,56 Kr
영국	Pound (£)	£1,234.56
미국	달러 (\$)	\$1,234.56
태국어	바트	2539 바트
유로	Euro (€)	€ 5,000

현재 릴리스는 유로 통화를 지원합니다. 하지만 역 호환성을 위해 현지 통화 기호도 사용할 수 있습니다.

표 1-6 유로 통화를 지원하는 사용자 로케일

지역	로케일 이름	ISO Code Set
오스트리아	de_AT.ISO8859-15	8859-15
벨기에(프랑스어)	fr_BE.ISO8859-15	8859-15
벨기에(플란더즈어)	nl_BE.ISO8859-15	8859-15
덴마크	da_DK.ISO8859-15	8859-15
에스토니아	et_EE.ISO8859-15	8859-15
핀란드	fi_FI.ISO8859-15	8859-15
프랑스	fr_FR.ISO8859-15	8859-15
독일	de_DE.ISO8859-15	8859-15
영국	en_GB.ISO8859-15	8859-15
아일랜드	en_IE.ISO8859-15	8859-15
이탈리아	it_IT.ISO8859-15	8859-15
네덜란드	nl_NL.ISO8859-15	8859-15
포르투갈	pt_PT.ISO8859-15	8859-15
카탈로니아어 스페인	ca_ES.ISO8859-15	8859-15

표 1-6 유로 통화를 지원하는 사용자 로캘 (계속)

지역	로캘 이름	ISO Code Set
스페인	es_ES.ISO8859-15	8859-15
스웨덴	sv_SE.ISO8859-15	8859-15
미국	en_US.ISO8859-15	8859-15

유로 로캘은 ISO8859-15 코드 세트를 기본으로 합니다.

변환된 통화 금액에 대하여 원래 금액과 다른 공간이 필요할 수 있다는 점을 기억하십시오. 예를 들어, \$1,000이 €1.307.000이 될 수 있습니다.

현재 유로 사용 지역 내의 로캘용 로캘 설정 상태 예는 로캘 유틸리티의 LC_MONETARY 피연산자에 대한 것입니다. 예를 들어, 독일의 상태는 다음 표에 표시되어 있습니다.

표 1-7 독일어 로캘과 관련 LC_MONETARY 피연산자

로캘	LC_MONETARY
de_DE.ISO8859-1	DM
de_DE.ISO8859-15	유로
de_DE.UTF-8	유로
de_DE.ISO8859-15@euro	유로
de_DE.UTF-8@euro	유로

언어 단어 및 글자 차이

이 절에서는 언어간의 중요한 차이점을 설명합니다.

단어 분리자

영어에서 단어는 일반적으로 공간 문자로 구분됩니다. 하지만 중국어, 일본어 및 태국어와 같은 언어에서는 단어 사이에 분리자가 없는 경우가 많습니다.

정렬 순서

특정한 문자에 대한 정렬 순서는 모든 언어에서 동일하지 않습니다. 예를 들어, 문자 “ö”는 독일에서 보통의 “o”와 같이 분류됩니다. 하지만 스웨덴에서는 별개의 문자로 분류되며 알파벳의 끝 문자에 해당합니다. 일부 언어에서 문자에는 문자 시퀀스의 우선 순위를 결정하기 위해 가중치가 주어집니다. 예를 들어, 태국어 사전은 서로 가중치가 다른 문자들의 순서를 통해 정렬을 규정합니다.

문자 세트

문자 세트는 알파벳 문자와 특수 문자 수에 있어서 다를 수 있습니다. 영문자에는 26개 문자가 포함되지만 일부 언어에는 훨씬 많은 문자가 포함됩니다. 예를 들어, 일본어는 20,000자 이상이 포함될 수 있으며 중국어에는 이보다도 더 많은 문자가 포함될 수 있습니다.

서유럽 문자

서유럽 국가의 알파벳은 영어 사용 국가에서 사용되는 표준 26문자 알파벳과 유사합니다. 이러한 알파벳은 몇 가지 추가 기본 문자, 몇 가지 표시 또는 강제 표시되는 문자와 몇 가지 합자도 포함합니다.

일본어 텍스트

일본어 텍스트는 다음 세 가지 다른 스크립트가 혼합되어 구성됩니다

- 간지 는 한자에서 유래한 표의 문자입니다.
- 히라가나 및 가타카나, 두 가지 음성 스크립트(또는 음절표)

히라가나의 각 문자는 가타카나에도 상응하는 문자가 있지만 히라가나는 가장 공통된 문자이고 블록체보다는 초서체에 가깝습니다. 간지 문자는 핵심 단어를 쓰는데 사용됩니다. 가타카나는 대개 “외래어”, 다시 말해 일본어 이외의 언어에서 수입된 단어를 표현하는데 사용됩니다.

간지는 수 만개의 문자를 갖고 있지만 시간이 지날수록 자주 사용되는 문자 수는 줄어들고 있습니다. 평균적인 일본어 작가들은 대략 2000자의 간지 문자를 사용할 수 있는 어휘력을 갖고 있지만 지금은 약 3500자만이 자주 사용되고 있습니다. 하지만 컴퓨터 시스템은 일본 공업 표준(JIS)이 요구하는 7000자 이상을 지원해야 합니다. 아울러 약 170개의 히라가나와 가타카나 문자가 있습니다. 평균적으로 일본어 텍스트의 55%가 히라가나이며 35%는 간지 그리고 10%가 가타카나입니다. 아라비아 숫자와 로마 문자 역시 일본어 텍스트에 사용됩니다.

간지를 전혀 사용하지 않는 것도 가능하지만 대부분의 일본 독자들은 간지를 전혀 사용하지 않고 작성한 텍스트는 이해하기 힘들어 합니다.

한국어 텍스트

한국어 텍스트는 한글이라는 음성 쓰기 체계를 사용하여 작성할 수 있습니다. 한글은 자모로 알려져 있는 자음과 모음으로 구성된 11,000자 이상의 문자를 갖고 있습니다. 전체 한글 문자 사전 중에서 대략 3000자가 한국어 컴퓨터 시스템에서 일상적으로 사용됩니다. 한국인들은 중국에서 발명된 문자를 바탕으로 한 한자라는 표의 문자도 사용합니다. 한국어 텍스트에는 6,000자 이상의 한자 문자가 필요합니다. 한자는 대개 한글의 의미가 모호할 때 혼동을 방지하기 위해 사용됩니다. 한글 문자는 자음과 모음을 결합하여 형성됩니다. 이들 문자가 결합되어 하나의 한글 문자에 해당하는 하나의 음절을 합성할 수 있습니다. 한글 문자는 보통 사각형으로 배열되며 이 그룹은 한자 문자와 동일한 공간을 점유합니다. 아라비아 숫자, 로마 문자 및 특수 기호 문자도 한국어 텍스트에 사용됩니다.

태국어 텍스트

태국어 문자는 디스플레이 화면에 네 개의 디스플레이 셀로 구성되는 열 위치로 정의할 수 있습니다. 각 열 위치에 최대 세 개의 문자를 표현할 수 있습니다. 디스플레이 셀의 구성은 태국어 문자의 분류법에 따릅니다. 일부 태국어 문자는 다른 문자의 분류법과 함께 구성될 수 있습니다. 함께 구성할 수 있는 경우에는 두 문자가 동일한 셀에 표시됩니다. 그렇지 않으면 별개의 셀들에 표시됩니다.

중국어 텍스트

중국어는 보통 hanzi라는 표의 문자로만 구성됩니다.

- 중화인민공화국(PRC)에는 유니코드 3.0에 정의된 모든 CJK 확장 A 문자를 포함하여 GB2312(zh 로캘)의 공통적으로 사용되는 Hanzi 문자 7000개와 GBK 문자 세트(zh.GBK 로캘)의 20,000개 이상의 문자와 GB18030-2000 문자 세트(zh_CN.GB18030 로캘)의 약 30,000개 문자가 있습니다.
- 대만에서 가장 많이 사용되는 문자 세트는 CNS11643-1992(zh_TW 로캘) 및 Big5(zh_TW.BIG5 로캘)입니다. 이들은 약 13,000자의 Hanzi 문자를 공유합니다.
- 홍콩에서는 4702개 문자가 Big5 문자 세트에 추가되어 Big5-HKSCS 문자 세트(zh_HK.BIG5HK)가 되었습니다.

특정 문자가 기본 문자가 아닌 경우에는 대개 두 개 이상의 부분으로 구성되며 이 중 두 자는 매우 자주 쓰이는 문자입니다. 두 부분으로 구성된 문자에서 한 부분은 보통 의미를, 다른 한 부분은 발음을 나타냅니다. 두 부분이 모두 의미를 나타내는 경우도 있습니다. 부수는 가장 중요한 요소이며 문자는 전통적으로 부수별로 정렬되며 수 백개의 부수 문자가 있습니다. 하나의 음은 서로 상호 교환해 사용할 수 없는 여러 개의 서로 다른 문자들로 표현될 수 있습니다. 하나의 문자는 여러 가지 음을 가질 수 있습니다.

어떤 문자가 주어진 컨텍스트에서 다른 것보다 더 적합합니다. 적합한 문자는 음조 사용으로 음성적으로 구분됩니다. 이와 대조적으로 일본어와 한국어는 성조가 많지 않습니다.

몇 가지 음성 체계가 중국어를 나타냅니다. 중국에서 가장 흔한 것은 *pinyin*으로서 로마 문자를 사용하며 Beijing과 같은 장소 이름을 표기하기 위해 서구에서 널리 사용되고 있습니다. Wade-Giles 체계는 이보다 오래된 음성학적 체계로서 과거에 Peking과 같이 장소 이름에 사용되었습니다. 대만에서는 대신 고유한 문자 형식의 음성 알파벳 *zhuyin* (또는 *bopomofo*)이 사용됩니다.

히브리어 텍스트

히브리어 텍스트는 히브리어와 Yiddish 언어로 스크립트를 작성하는 데 사용됩니다. 히브리어는 양방향 스크립트를 사용합니다. 히브리어 문자는 오른쪽에서 왼쪽으로 쓰고 읽는 반면 숫자는 왼쪽에서 오른쪽으로 읽습니다. 히브리어 텍스트에 포함된 모든 영어 텍스트 역시 왼쪽에서 오른쪽으로 읽습니다.

히브리어는 27개 문자를 사용하며 표준 라틴(또는 영어) 문자의 구두점과 숫자를 사용합니다. 히브리어 텍스트에는 또한 모음과 발음 기호가 포함되어 있습니다. 이 기호들은 기본 문자 안의 점(*dagesh*), 문자 아래의 모음 기호 또는 문자 상단 왼쪽의 악센트 기호로 표시됩니다. 이 기호들은 보통 예배용 텍스트에서만 사용되며 일상적으로는 거의 사용되지 않습니다. 히브리어에는 대문자가 없습니다.

힌디어 텍스트

힌디어 텍스트는 "신의 기록"이라는 의미를 갖고 있는 Devanagari라 부르는 문자로 작성됩니다. 힌디어는 표음 언어로서 연속적인 음절로 쓰여집니다. 각각의 음절은 다음 세 가지 문자 조각(Devanagari 문자)으로 구성됩니다: 자음 문자, 독립 모음 및 종속 모음 기호. 음절 자체는 자음과 기본 모음으로 구성되며 별도의 전치 모음이 있을 수 있습니다. 기준선에서 시작되는 영어와 달리 Devanagari 문자는 문자의 상단에 쓰여 있는 가로선(head stroke)에 매달려 있습니다. 이 문자들은 문맥에 따라 결합하거나 모양을 변경할 수 있습니다. 히브리어와 마찬가지로 힌디어 텍스트는 대소문자 구분을 하지 않습니다.

키보드 차이점

미국 키보드의 모든 문자가 다른 키보드에 나타나는 것은 아닙니다. 마찬가지로 다른 키보드에는 미국 키보드에서 볼 수 없는 많은 문자들이 포함되어 있습니다.

입력은 Solaris Operating System에 의해 처리되기 때문에 어떤 로케일의 입력 문자든 임의의 키보드를 사용해 입력할 수 있습니다.

주 - SPARC® 시스템에서는 Compose 키를 사용해 지원되는 ISO8859 문자 세트의 분음 부호를 갖고 있는 라틴 문자를 만들 수 있습니다. Compose 키는 라틴 기반 로케일과 사용할 수 있지만 UTF-8 로케일을 제외한 한국어, 중국어 또는 일본어 로케일은 사용할 수 없습니다.

용지 크기의 차이

각 국가 내에서 적은 수의 용지 크기가 일반적으로 사용됩니다. 일반적으로 이 크기 중 한 가지가 다른 것들보다 훨씬 일반적입니다. 대부분의 국가들은 ISO 표준 216을 준수합니다. “Writing paper and certain classes of printed matter-Trimmed sizes-A and B series.”

국제화된 응용프로그램들은 이용 가능한 페이지 크기에 대해 가정해선 안 됩니다. Solaris 시스템은 출력 페이지 크기 추적을 지원하지 않습니다. 페이지 크기 추적은 응용프로그램의 책임입니다. 다음 표는 공통적인 국제 페이지 크기를 보여줍니다.

표 1-8 공통 국제 페이지 크기

용지 유형	규격	국가
ISO A4	21.0 cm X 29.7 cm	미국을 제외한 모든 국가
ISO A5	14.8 cm X 21.0 cm	미국을 제외한 모든 국가
JIS B4	25.9 cm X 36.65 cm	일본
JIS B5	18.36 cm X 25.9 cm	일본
U.S. Letter	8.5 인치 X 11 인치	미국 및 캐나다
U.S. Legal	8.5 인치 X 14 인치	미국 및 캐나다

일반 국제화 기능

이 절은 Solaris Operating System 환경에 포함된 여러 국제화 기능에 대해 다룹니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 35 페이지 “코드 세트 독립 지원”
- 37 페이지 “로캘 데이터베이스”
- 38 페이지 “프로세스 코드 형식”
- 38 페이지 “멀티바이트 지원 환경”
- 39 페이지 “동적으로 링크된 응용프로그램”
- 39 페이지 “변경된 인터페이스”
- 40 페이지 “ctype 매크로”
- 41 페이지 “libc의 국제화 API”
- 48 페이지 “genmsg 유틸리티”
- 49 페이지 “사용자 정의 및 사용자 확장 가능 코드 변환”
- 49 페이지 “국제화된 도메인 이름(IDN) 지원”

코드 세트 독립 지원

EUC는 Extended UNIX® Code의 축약입니다. Solaris Operating System은 일본의 PC-Kanji(Shift_JIS로 더 잘 알려짐), 대만의 Big5, 중화인민공화국의 GBK와 같은 비EUC 코드화를 지원합니다. 대다수의 컴퓨터 시장에서 비 EUC 코드 세트 지원을 요구하기 때문에 현재 Solaris 환경은 EUC와 비 EUC 코드 세트 지원을 활성화하는 견고한 프레임워크를 제공합니다. 이 지원을 **코드 세트 독립** 또는 **CSI**라고 합니다.

CSI의 목적은 Solaris Operating System 라이브러리와 명령으로부터, 특정한 코드 세트나 코드화 방법에 대한 의존성을 제거하는데 있습니다. CSI 아키텍처는 Solaris Operating System에서 모든 UNIX 파일 시스템 안전 코드화를 지원할 수 있도록 합니다. CSI는 UTF-8, PC-Kanji 및 Big5와 같은 많은 새 코드 세트를 지원합니다.

CSI 접근 방법

CSI는 응용프로그램 및 플랫폼 소프트웨어 개발자가 UTF-8 같은 모든 코드화와 무관하게 코드를 작성할 수 있도록 해 주며, 또한 소스 코드를 수정하지 않고서도 새로운 코드화 방식을 적용할 수 있도록 해 줍니다. 이 아키텍처 접근 방법은 응용 프로그램이 UTF 종속적일 필요가 없기 때문에 Java™ 국제화와 다릅니다.

많은 기존 국제화된 응용프로그램(예를 들어, Motif)이 기본 시스템에서 자동으로 CSI 지원을 상속합니다. 이러한 응용프로그램은 수정 없이 새 로케에서 작동합니다.

CSI는 본질적으로 모든 코드 세트로부터 독립적입니다. 그러나 파일 코드 코드화(코드 세트)에 대한 다음 가정이 현재 Solaris 환경에 적용됩니다.

- 파일 코드는 ASCII의 슈퍼 세트입니다.
- 널 바이트 값(0x00)은 널로 종결되는 문자열의 지원을 위해 멀티바이트 문자 바이트의 일부로 나타나지 않습니다.
- ASCII 슬래시 문자 바이트 값(0x2f)은 UNIX 경로 이름 지원을 위해 멀티바이트 문자 바이트로 나타나지 않습니다.

CSI화 된 명령

이 절은 현재 Solaris 환경의 CSI화 된 명령을 나열합니다. 각 명령어의 설명서 페이지에는 명령이 CSI화 되었는지를 나타내는 속성 부분이 있습니다.

모든 명령은 특별히 표시되지 않는 한, /usr/bin 디렉토리에 있습니다.

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

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

CSI 활성화 라이브러리

libc (/usr/lib/libc.so)의 거의 모든 기능이 CSI 활성화되었습니다. 그러나 libc의 다음 함수는 EUC 종속 함수이기 때문에 CSI화 되지 않았습니다.

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

현재 Solaris 제품에서 libgen /usr/ccs/lib/libgen.a 및 libcurses /usr/ccs/lib/libcurses.a 는 국제화되었지만 CSI화 되지 않았습니다.

로켈 데이터베이스

로켈 데이터베이스 형식과 구조는 운영 환경 내부적이며 향후 릴리스에서 변경될 것입니다. 국제화 응용 프로그램을 개발할 경우 libc의 국제화 API를 사용합니다. 이 API는 로켈 데이터베이스에 연결되지 않고 41 페이지 “libc의 국제화 API”에서 설명됩니다.

주 - Solaris 환경으로 작업할 때 현재 Solaris 제품에 포함된 로켈 데이터베이스를 사용하십시오. 이전 Solaris 버전의 로켈을 사용하지 마십시오.

프로세스 코드 형식

Solaris 9 제품에서 '넓은 문자'로 알려져 있는 프로세스 코드 형식은 Solaris Operating System 운영 환경 내부적인 것이며 향후 릴리스에서 변경될 것입니다. 따라서, 국제화된 응용프로그램을 개발할 때 프로세스 코드 형식이 동일하다고 가정하지 마십시오. 대신 41 페이지 "libc의 국제화 API"에서 설명된 libc의 국제화 API를 사용하십시오.

주 - 모든 유니코드의 프로세스 코드는 UTF 32 표현에 있습니다. UTF-32에 대한 자세한 내용은 유니코드 컨소시엄의 "Unicode Standard Annex #19: UTF 3 및 Unicode Standard Annex #27: 유니코드 컨소시엄의 Unicode 3.1 또는 <http://www.unicode.org/>를 참조하십시오.

멀티바이트 지원 환경

멀티바이트 문자는 중국어, 일본어 또는 한국어 문자와 같이 단일 바이트로는 저장할 수 없는 문자입니다. 이러한 문자는 기억 장치의 2, 3 또는 4바이트가 필요합니다. 더 정확한 정의는 ISO/IEC 9899:1990 부속 절 3.13에서 찾을 수 있습니다.

추가된 새 국제화 기능, ISO/IEC 9899:1990라고도 하는 ANSIC에 대한 개정 1은 일괄하여 멀티바이트 지원 환경(MSE)이라고도 합니다. 개정 1은 상태가 있는 멀티바이트 코드 세트와 더 나은 와이드캐릭터 처리 지원을 위한 추가 국제화 API를 정의합니다.

프로그래밍 모델은 멀티바이트 문자가 논리 장치에서 읽히고 내부적으로 넓은 문자로 저장 가능하게 합니다. 넓은 문자는 프로그램에 의해 논리적 엔티티로 처리될 수 있습니다. 최종적으로 이 넓은 문자는 적절한 변환을 거쳐 논리적 단위로 외부장치로 쓰여질 수 있습니다.

절차는 단일바이트 문자가 읽히고 조작되며 재작성되는 방식과 유사합니다. MSE는 프로그램이 단일바이트 문자에 사용되는 동일한 프로그래밍 모델을 사용하여 멀티바이트 문자를 처리하도록 합니다.

동적으로 링크된 응용프로그램

동적 링크 만들기 또는 정적 링크 만들기를 사용하여 libc 응용 프로그램을 시스템 라이브러리와 함께 링크로 만들 수 있습니다. 시스템 라이브러리에 국제화 기능이 필요한 모든 응용 프로그램은 동적으로 링크되어야 합니다. 만약 응용 프로그램이 정적으로 링크가 되어 있을 경우, `setlocale`을 호출해서 로케일을 C나 POSIX 이외의 다른 로케일로 설정할 수 없습니다. 정적으로 링크된 응용 프로그램은 C 및 POSIX 로케일에서만 작동할 수 있습니다.

기본적으로, 링커 프로그램은 응용 프로그램을 동적으로 연결하려고 시도합니다. 링커 및 컴파일러에 대한 명령줄 옵션이 `-Bstatic` 또는 `-dn` 사양을 포함하면 응용 프로그램이 정적으로 링크될 수 있습니다. 기존 응용 프로그램이 `/usr/bin/ldd` 명령을 사용하여 동적으로 링크되었는지 확인할 수 있습니다.

예를 들어, 다음 명령에 대한 응답은 `/sbin/sh` 명령이 동적으로 링크된 프로그램이 아님을 나타냅니다.

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

이 응답은 `/usr/bin/ls` 명령이 두 라이브러리, `libc.so.1` 및 `libdl.so.1`과 동적으로 링크되었음을 나타냅니다.

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

변경된 인터페이스

`libw` 및 `libintl`은 `libc`로 이동하여 더 이상 `libw` 및 `libintl`에 없습니다.

공유 객체(shared object)는 기존 응용 프로그램에 대한 런타임 호환성을 보장하고, 아카이브와 함께 응용 프로그램을 빌드하기 위한 컴파일 환경의 호환성도 제공합니다. 그러나 더 이상 `libw`나 `libintl`에 프로그램을 링크시키지 말아야 합니다.

다음 목록은 `libw`의 스티브 시작점을 보여줍니다.

<code>fgetwc</code>	<code>getws</code>	<code>iswalnum</code>	<code>iswlower</code>
<code>fgetws</code>	<code>isenglish</code>	<code>iswalpha</code>	<code>iswprint</code>
<code>fputwc</code>	<code>isideogram</code>	<code>iswcntrl</code>	<code>iswpunct</code>
<code>fputws</code>	<code>isnumber</code>	<code>iswctype</code>	<code>iswspace</code>
<code>getwc</code>	<code>isphonogram</code>	<code>iswdigit</code>	<code>iswupper</code>
<code>getwchar</code>	<code>isspecial</code>	<code>iswgraph</code>	<code>iswxdigit</code>

putwc	wscscpn	wcsxfrm	wsncat
putwchar	wcsftime	wctype	wsncmp
putws	wscncat	wcwidth	wsncpy
strtows	wcsncmp	wscasecmp	wspbrk
towlower	wcsncpy	wscat	wsprintf
toupper	wcspbrk	wchr	wsrchr
ungetwc	wcsrchr	wscmp	wsscanf
watoll	wcsspn	wscoll	wsspn
wscat	wcstod	wscoll	wstod
wchr	wcstok	wscpy	wstok
wcsclen	wcstol	wscspn	wstol
wscmp	wcstoul	wsdup	wstoll
wscoll	wcswcs	wslen	wstostr
wscopy	wcswidth	wsncasecmp	wsxfrm

다음 목록은 libintl의 스테브 시작점을 보여줍니다.

```
bindtextdomain
dcgettext
dgettext
gettext
textdomain
```

ctype 매크로

문자 분류 및 문자 변형 매크로는 /usr/include/ctype.h에서 정의됩니다. 현재 Solaris 환경은 XPG4로 정의되는 문자 분류 및 변형 의미를 지원하는 ctype 매크로 세트를 제공합니다. 모든 XPG4 및 XPG4.2 응용 프로그램의 경우 다음 중 하나의 조건이 충족되어야 합니다.

- `_XPG4_CHAR_CLASS`가 정의됩니다.
- `_XOPEN_SOURCE` 및 `_XOPEN_VERSION=4`가 정의됩니다.
- `_XOPEN_SOURCE` 및 `_XOPEN_SOURCE_EXTENDED=1`가 정의됩니다.

`_XOPEN_SOURCE`, `_XOPEN_VERSION` 및 `_XOPEN_SOURCE_EXTENDED`가 새 ctype 매크로 외에 추가 XPG4 관련 기능을 가져오기 때문에 비XPG4 또는 XPG4.2 응용 프로그램은 `_XPG4_CHAR_CLASS`를 사용해야 합니다.

해당 ctype 함수도 존재합니다. 현재 Solaris 환경 함수는 XPG4 의미도 지원합니다.

libc의 국제화 API

현재 Solaris 환경은 두 개의 API 세트를 제공합니다.

- 멀티바이트(파일 코드)
- 넓은 문자(프로세스 코드)

와이드캐릭터는 논리적 엔티티로서 고정폭의 단위입니다. 따라서, 멀티바이트 문자를 사용할 때처럼 계속해서 문자의 경계를 추적해야 할 필요가 없습니다.

프로그램이 파일에서 입력을 받으면 사용자는 `fscanf` 및 `fwscanf`와 같은 입력 기능으로 직접 또는 입력 후 `mbtowc` 및 `mbsrtowcs`와 같은 변환 기능을 사용하여 자기 파일의 멀티바이트 데이터를 넓은 문자 프로세스 코드로 직접 변환할 수 있습니다. 넓은 문자 형식에서 멀티바이트 문자 형식으로 출력 데이터를 변환하려면 `fwprintf(3S)` 및 `fprintf(3S)`와 같은 출력 함수를 사용하거나 출력 후 `wctomb(3C)` 및 `wcsrtombs(3C)`와 같은 변환 함수를 적용하십시오.

이 장의 나머지 표는 현재 Solaris 제품에 포함된 국제화 API에 대해 설명합니다.

다음 표는 libc의 메시지 처리 함수 API를 설명합니다.

표 2-1 메시징 libc의 기능

라이브러리 루틴	설명
<code>bindtextdomain()</code>	메시지 도메인에 대한 경로 바인드
<code>catclose()</code>	메시지 카탈로그 닫기
<code>catgets()</code>	프로그램 메시지 읽기
<code>catopen()</code>	메시지 카탈로그 열기
<code>dcgettext()</code>	도메인과 범주를 지정하여 메시지 카탈로그에서 메시지 얻기
<code>dgettext()</code>	도메인을 지정하여 메시지 카탈로그에서 메시지 얻기
<code>gettext()</code>	메시지 데이터베이스에서 텍스트 문자열 검색
<code>textdomain()</code>	현재 도메인을 설정 및 질의

다음 표는 libc의 코드 변환 기능 API를 설명합니다.

표 2-2 코드 libc의 변환

라이브러리 루틴	설명
<code>iconv()</code>	코드 변환

표 2-2 코드 libc의 변환 (계속)

라이브러리 루틴	설명
iconv_close()	변환 설명자 할당 해제
iconv_open()	변환 설명자 할당

다음 표는 libc의 정규 표현식 API를 설명합니다.

표 2-3 정규 libc의 표현식

라이브러리 루틴	설명
fnmatch()	파일 이름 또는 경로 이름 일치
regcomp()	정규식 컴파일
regerror()	오류 코드에서 오류 메시지로의 매핑을 제공
regexexec()	정규식 매칭 실행
regfree()	regcomp()가 할당한 사용 가능 메모리

다음 표는 libc의 넓은 문자 함수 API를 설명합니다.

표 2-4 넓은 libc의 문자 클래스

라이브러리 루틴	설명
wctrans()	문자 매핑 정의
wctype()	문자 클래스 정의

다음은 libc의 로캘 수정 및 질의를 나열합니다.

표 2-5 libc의 로캘 수정 및 질의

라이브러리 루틴	설명
setlocale()	프로그램의 로캘 수정 및 질의

다음 표는 libc의 로캘 데이터 질의를 나열합니다.

표 2-6 libc의 로캘 데이터 질의

라이브러리 루틴	설명
localeconv()	현재 로캘의 통화 및 숫자 형식 정보 얻기
nl_langinfo()	현재 로캘의 언어 및 문화 정보 얻기

다음 표는 libc의 문자 분류 함수 API를 설명합니다.

표 2-7 libc의 문자 분류 및 음역

라이브러리 루틴	설명
isalnum()	문자가 알파벳인가 숫자인가?
isalpha()	문자가 알파벳인가?
isascii()	문자가 ASCII 문자인가?
iscntrl()	문자가 제어 문자인가?
isdigit()	문자가 숫자인가?
isenglish()	영어 알파벳의 넓은 문자가 추가 코드 세트에 왔는가?
isgraph()	문자가 가시 문자인가?
isideogram()	넓은 문자가 표의 문자인가?
islower()	문자가 소문자인가?
isnumber()	넓은 문자가 추가 코드 세트의 숫자인가?
isphonogram()	넓은 문자가 표음 문자인가?
isprint()	문자가 인쇄 가능한가?
ispunct()	문자가 구두점인가?
isspace()	문자가 공백인가?
isspecial()	특수 넓은 문자가 추가 코드 세트에 있는가?
isupper()	문자가 대문자인가?
iswalnum()	넓은 문자가 알파벳 문자인가 숫자인가?
iswalpha()	넓은 문자가 알파벳인가?
iswascii()	넓은 문자가 ASCII 문자인가?
iswcntrl()	넓은 문자가 제어 문자인가?
iswdigit()	문자가 숫자인가?
iswgraph()	넓은 문자가 가시 문자인가?
iswlower()	넓은 문자가 소문자인가?
iswprint()	넓은 문자가 인쇄 가능 문자인가?
iswpunct()	넓은 문자가 구두점인가?
iswspace()	넓은 문자가 빈 공백인가?
iswupper()	넓은 문자가 대문자인가?
iswxdigit()	넓은 문자가 6진수인가?

표 2-7 libc의 문자 분류 및 음역 (계속)

라이브러리 루틴	설명
isxdigit()	문자가 16진수인가?
tolower()	대문자를 소문자로 변환합니다.
toupper()	소문자를 대문자로 변환합니다.
towctrans()	넓은 문자 매핑
tolower()	대문자 넓은 문자를 소문자로 변환합니다.
toupper()	소문자 넓은 문자를 대문자로 변환합니다.

다음 표는 libc의 문자 조합 함수 API를 설명합니다.

표 2-8 libc의 문자 조합

라이브러리 루틴	설명
strcoll()	문자열 조합
strxfrm()	비교를 위해 문자열 변환
wscoll()	넓은 문자열 조합
wcsxfrm()	비교를 위해 넓은 문자열 변환

다음 표는 libc의 통화 처리 함수 API를 설명합니다.

표 2-9 libc의 통화 형식

라이브러리 루틴	설명
localeconv()	현재 로캘의 통화 형식 정보 얻기
strfmon()	통화 값을 문자열 표현으로 변환

다음 표는 libc의 날짜 및 시간 형식을 설명합니다.

표 2-10 libc의 날짜 및 시간 형식

라이브러리 루틴	설명
getdate()	사용자 형식 날짜 및 시간을 변환
strftime()	날짜 및 시간을 문자열 표현으로 변환%u 변환 기능은 X/Open CAE 사양, 시스템 인터페이스 및 헤더, 4판, 버전 2를 준수합니다. 이 기능은 평일을 십진수로 표현하여 [1,7]에서 1은 월요일을 나타냅니다..

표 2-10 libc의 날짜 및 시간 형식 (계속)

라이브러리 루틴	설명
strptime()	날짜 및 시간 변환

다음 표는 libc의 멀티바이트 처리 함수 API를 설명합니다.

표 2-11 멀티바이트 libc의 처리

라이브러리 루틴	설명
btowc()	단일바이트를 넓은 문자로 변환
mbllen()	문자의 바이트 수 얻기
mbrlen()	문자의 바이트 수 얻기(다시 시작 가능)
mbrtowc()	문자를 넓은 문자 코드로 변환(다시 시작 가능)
mbsinit()	변환 객체 상태 결정
mbsrtowcs()	문자열을 넓은 문자열로 변환(다시 시작 가능)
mbstowcs()	문자열을 넓은 문자열로 변환
mbtowc()	문자를 넓은 문자 코드로 변환

다음 표는 libc의 넓은 문자 및 문자열 처리를 설명합니다.

표 2-12 libc의 넓은 문자 및 문자열 처리

라이브러리 루틴	설명
wcrtomb()	넓은 문자를 문자로 변환(다시 시작 가능)
wcscat()	넓은 문자열 연결
wcschr()	넓은 문자열에서 문자 찾기
wcscmp()	넓은 문자열 비교
wcscpy()	넓은 문자열 복사
wcscspn()	다른 넓은 문자열에 없는 하나의 넓은 문자열 범위 반환
wcslen()	넓은 문자열 길이 얻기
wcsncat()	넓은 문자를 길이 n 으로 연결
wcsncmp()	넓은 문자를 길이 n 과 비교
wcsncpy()	넓은 문자를 길이 n 으로 복사
wcspbrk()	다른 넓은 문자열에 없는 하나의 넓은 문자열 포인터 반환

표 2-12 libc의 넓은 문자 및 문자열 처리 (계속)

라이브러리 루틴	설명
wcsrchr()	오른쪽부터 넓은 문자열에서 문자 찾기
wcsrtombs()	넓은 문자열을 문자열로 변환(다시 시작 가능)
wcsspn()	다른 넓은 문자열에 있는 하나의 넓은 문자열 범위 반환
wcstod()	넓은 문자열을 배정도로 변환
wcstok()	넓은 문자열을 통해 토큰 이동
wcstol()	넓은 문자열을 긴 정수로 변환
wcstombs()	넓은 문자열을 멀티바이트 문자열로 변환
wcstoul()	넓은 문자열을 부호없는 긴 정수로 변환
wscwcs()	넓은 문자열에서 문자열 찾기
wcswidth()	넓은 문자열의 열 위치 수 결정
wctob()	넓은 문자를 단일바이트로 변환
wctomb()	넓은 문자를 멀티바이트 문자로 변환
wcwidth()	넓은 문자의 열 위치 수 결정
wscol()	넓은 문자열의 표시 너비 반환
wsdup()	넓은 문자열 중복

다음 표는 libc의 형식화된 넓은 문자 입출력을 설명합니다.

표 2-13 libc의 형식화된 넓은 문자 입력 및 출력

라이브러리 루틴	설명
fwprintf()	형식화된 넓은 문자 출력 인쇄
fwscanf()	형식화된 넓은 문자 입력 변환
swprintf()	형식화된 넓은 문자 출력 인쇄
swscanf()	형식화된 넓은 문자 입력 변환
vwprintf()	stdarg 인자 목록의 넓은 문자 형식화된 출력
vswprintf()	stdarg 인자 목록의 넓은 문자 형식화된 출력
wprintf()	형식화된 넓은 문자 출력 인쇄
wscanf()	형식화된 넓은 문자 입력 변환
wswprintf()	형식에 따라 넓은 문자열 생성

표 2-13 libc의 형식화된 넓은 문자 입력 및 출력 (계속)

라이브러리 루틴	설명
wscanf()	형식화된 입력 변환

이 표는 libc의 넓은 문자열 함수 API를 설명합니다.

표 2-14 libc의 넓은 문자열

라이브러리 루틴	설명
wcsstr()	넓은 문자열 부속 문자열 찾기
wmemchr()	메모리의 넓은 문자 찾기
wmemcmp()	메모리의 넓은 문자 비교
wmemcpy()	메모리의 넓은 문자 복사
wmemmove()	겹친 영역이 있는 메모리의 넓은 문자 복사
wmemset()	메모리의 넓은 문자 설정
wscasecmp()	넓은 문자열을 비교하고 대소문자 차이를 무시
wscasecmp()	코드 문자열 작업 프로세스

다음 표는 libc의 넓은 문자 입출력을 설명합니다.

표 2-15 libc의 넓은 문자 입력 및 출력

라이브러리 루틴	설명
fgetc()	스트림에서 멀티바이트 문자를 얻어 넓은 문자로 변환
fgetws()	스트림에서 멀티바이트 문자열을 얻어 넓은 문자로 변환
fputc()	넓은 문자열을 멀티바이트 문자로 변환하여 스트림에 넣기
fputws()	넓은 문자를 멀티바이트 문자열로 변환하여 스트림에 넣기
fwide()	스트림 방향 설정
getwchar()	stdin에서 멀티바이트 문자를 얻어 넓은 문자로 변환
getws()	stdin에서 멀티바이트 문자열을 얻어 넓은 문자로 변환
putwchar()	넓은 문자열을 멀티바이트 문자로 변환하여 stdin에 넣기
putws()	넓은 문자를 멀티바이트 문자열로 변환하여 stdin에 넣기
ungetc()	넓은 문자를 입력 스트림으로 푸시백

genmsg 유틸리티

새 genmsg 유틸리티는 catgets() 계열의 함수들이 사용할 수 있는, 국제화된 소스 메시지 카탈로그를 만드는데 사용됩니다. 유틸리티는 catgets의 함수에 대한 호출을 위해 소스 프로그램 파일을 조사하고 찾은 정보에서 소스 메시지 카탈로그를 구축합니다. 예를 들면 다음과 같습니다.

```
% 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 */
```

위 예제에서, genmsg를 소스 파일 example.c에 실행하여, example.c.msg라는 소스 메시지 카탈로그를 생성해 냈습니다. 인자 NOTE를 옵션 -c와 함께 사용하여 genmsg가 카탈로그 안에 주석을 포함하도록 합니다. 소스 프로그램의 주석이 지정된 문자열을 포함하면 주석은 메시지 카탈로그에서 catgets에 대한 호출에서 추출된 다음 문자열 후에 나타납니다.

genmsg를 사용하여 메시지 세트에서 메시지를 자동으로 번호 매길 수 있습니다.

자세한 내용은 genmsg(1) 설명서 페이지를 참조하십시오.

형식화된 메시지 카탈로그 파일을 생성하려면 gencat (1) 유틸리티를 사용하십시오.

이식 가능 메시지 파일(.po files)에 대한 메시지 추출 유틸리티와 .po파일에서 메시지 객체 파일(.mo 파일)을 생성하는 방법에 대한 자세한 내용을 참조하십시오.

사용자 정의 및 사용자 확장 가능 코드 변환

사용자는 `geniconvtbl` 유틸리티를 사용하여 사용자 정의 코드 세트 변환기를 작성할 수 있습니다.

이 유틸리티를 사용하여 `iconv(1)` 및 `iconv(3C)`와 같은 표준 시스템 유틸리티 및 인터페이스를 사용하는 사용자 정의 및 사용자 정의 가능 코드 세트 변환을 할 수 있습니다. 이 기능은 응용프로그램이 호환되지 않는 데이터 유형, 특히 독점 소유 또는 구형 응용프로그램의 데이터를 처리하는 능력을 향상시킵니다. 기존 Solaris 코드 세트 변환에 대한 수정도 지원됩니다.

유틸리티에 대한 예제 입력 소스 파일은 `/usr/lib/iconv/geniconvtbl/srcs/` 디렉토리로 사용 가능합니다.

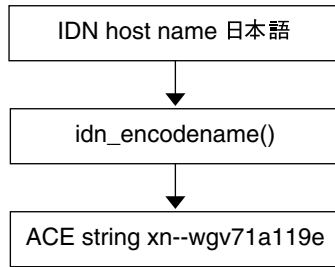
사용자 정의 코드 변환이 제대로 준비 및 배치되면 사용자는 32비트 및 64비트 Solaris Operating System의 `iconv(1)` 유틸리티 및 `iconv(3C)` 기능의 코드 변환을 사용할 수 있습니다.

국제화된 도메인 이름(IDN) 지원

국제화된 도메인 이름(IDN)을 사용하여 비영어 토착 언어 이름을 호스트 및 도메인 이름으로 사용할 수 있습니다. 비영어 호스트 및 도메인 이름을 사용하려면 이름을 RFC 3490에 지정된 해결자 루틴으로 보내기 전에 ASCII 호환 코드화(ACE) 코드화된 이름으로 이름을 변환합니다. 시스템 관리자는 또한 시스템 관리자 응용 프로그램이 IDN을 지원하지 않는 시스템 파일과 응용 프로그램에서 ACE 이름을 사용해야 합니다.

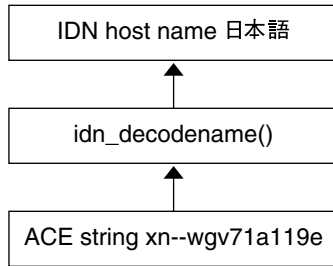
RFC 3490 응용 프로그램의 국제화 도메인 이름(DNA)을 참조하십시오.

`libidnkit(3EXT)`의 국제화 도메인 이름용 API는 UTF-8 또는 응용 프로그램 로케일의 코드 세트 및 ACE 사이의 편리한 변환을 제공합니다. `idn_decodename2(3EXT)`이 사용되면 자의적 코드 세트 이름을 입력 인자의 코드 세트로 지정할 수 있습니다.



Use ACE string as input to resolver routines such as `getaddrinfo(3SOCKET)`

그림 2-1 IDN 대 ACE 변환



ACE string returned from resolver routines such as `getnameinfo(3SOCKET)`

그림 2-2 ACE 대 IDN 변환

다음 표는 사용할 수 있는 `iconv` 코드 변환을 표시합니다.

표 2-16 `iconv` 코드 변환

출처 코드	대상 코드
ACE	UTF-8
ACE-ALLOW-UNASSIGNED	UTF-8
UTF-8	ACE
UTF-8	ACE-ALLOW-UNASSIGNED

ACE 및 ACE-ALLOW-UNASSIGNED `iconv` 코드 변환 이름은 다음 의미를 갖습니다.

- ACE.

ACE는 RFC 3490에 정의된 ASCII 호환 코드화를 참조하는 iconv 코드 변환에서 사용할 수 있는 fromcode 또는 tocode 이름입니다. 이 변환은 STD3 ASCII 규칙을 사용합니다. 할당되지 않은 문자는 허용되지 않습니다. ACE는 일반적으로 시스템에 호스트 또는 도메인 이름을 저장 또는 부여하는 데 사용됩니다.

■ ACE-ALLOW-UNASSIGNED.

ACE-ALLOW-UNASSIGNED는 ACE-ALLOW-UNASSIGNED가 비할당 문자를 허용한다는 점을 제외하고 ACE와 동일한 작업을 수행합니다.

ACE-ALLOW-UNASSIGNED는 일반적으로 질의 목적에 사용됩니다.

다음 예는 hostnames.txt 파일의 입력을 가진 ACE에서 UTF-8로 변환을 보여줍니다. 출력은 표준 출력입니다.

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

전용 IDN 변환 유틸리티 idnconv(1)는 여러 옵션을 가진 IDN 변환을 제공합니다. 옵션은 변환 세부사항을 제어합니다.

IDN, 변환 루틴 및 iconv 코드 변환에 대한 자세한 내용은 libidnkit(3LIB), idn_decodename(3EXT), idn_decodename2(3EXT), idn_encodename(3EXT) 및 iconv_en_US.UTF-8(5) 설명서 페이지를 참조하십시오.

Solaris 환경에서 지역화

이 장은 현재 Solaris 환경의 지역화 기능에 대해 설명합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 53 페이지 “지역화를 위한 소프트웨어 지원”
- 54 페이지 “지원되는 로켈”
- 60 페이지 “로켈용 복수 키 구성 시퀀스”
- 61 페이지 “Solaris 환경에서의 키보드 지원”
- 74 페이지 “새로운 Solaris 키보드 소프트웨어 지원”

지역화를 위한 소프트웨어 지원

이 절은 Solaris 로켈 패키지, CD-ROM 디스크, 지역화 기능 및 스크립트 환설화에 대한 정보를 포함합니다.

Solaris 로켈 패키지 요약

모든 현재 Solaris 로켈 패키지는 전체 로켈 또는 부분 로켈로 분류됩니다.

부분 로켈은 로켈의 활성화자입니다. 시스템에 설치되어 있는 부분 로켈을 통해 사용자는 대상 로켈로 텍스트를 입력, 표시, 인쇄하고 응용프로그램을 실행할 수 있습니다. 하지만 Solaris Operating System의 OS/GUI 메시지는 영어로 표현됩니다. 모든 부분 로켈 패키지는 Solaris Software CD에 들어 있습니다. 일본어와 아시아어 부분 로켈은 언어에 따라 패키지화됩니다. 부분 로켈은 지리적 지역에 따라 패키지화됩니다.

전체 로켈 패키지는 소프트웨어 메시지용 번역기, 온라인 도움말 파일, 선택 글꼴, 언어 특정 기능을 포함합니다. 전체 로켈 패키지는 여러 언어에 전체 언어 기능 모음을 제공합니다. 다음 언어를 바탕으로 한 모든 로켈은 전체 로켈입니다:

- 독일어

- 프랑스어
- 스페인어
- 스웨덴어
- 이탈리아어
- 일본어
- 한국어
- 간체 한자
- 번체 한자

전체 로케일 패키지는 언어별로 패키징되며 Language CD에서 사용할 수 있습니다.

주 - 전체 로케일이 작동하려면 부분 로케일 패키지(로케일 허가자)를 설치해야 합니다.

Solaris 설치 프로세스에서 지원이 필요한 지역을 선택하라는 메시지가 나타납니다. 설치가 완료된 뒤에 사용 가능한 로케일 지원은 이 단계에서의 선택에 따라 달라집니다. 부분 로케일은 Solaris Operating System 운영 환경이 포함된 Solaris 소프트웨어 CD-ROM에서 설치하며 전체 로케일은 Language CD에서 설치합니다. 전체 로케일 지원이 필요치 않으면 설치 프로세스 중에 Language CD에서 설치를 건너뛸 수 있습니다. 영어 로케일은 기본 설치된다는 점에 주의하십시오.

지원되는 로케일

다음 표는 Solaris 환경에서 지원되는 모든 로케일을 나열합니다. 로케일 이름은 국제 언어 지정 표준을 준수합니다.

표 3-1 아시아 로케일

로케일	사용자 인터페이스	지역	코드 세트	언어 지원
hi_IN.UTF-8	영어	인도	UTF-8	힌두어(UTF-8) 유니코드 4.0
ja	일본어	일본	eucJP ¹	일본어(EUC) JIS X 0201-1976 JIS X 0208-1990 JIS X 0212-1990
ja_JP.eucJP	일본어	일본	eucJP	일본어(EUC) JIS X 0201-1976

¹ eucJP는 일본어 EUC 코드 세트를 의미합니다. ja_JP.eucJP 로케일 사양은 UI_OSF Japanese Environment Implementation Agreement 버전 1.1을 따르며, ja 로케일은 이전 Solaris 릴리스의 전통적인 사양을 준수합니다.

표 3-1 아시아 로캘 (계속)

로캘	사용자 인터페이스	지역	코드 세트	언어 지원
				JIS X 0208-1990
				JIS X 0212-1990
ja_JP.PCK	일본어	일본	PCK ²	일본어(PC 간지)
				JIS X 0201-1976
				JIS X 0208-1990
ja_JP.UTF-8	일본어	일본	UTF-8	일본어(UTF-8) 유니코드 4.0
ko_KR.EUC	한국어	한국	1001	한국어(EUC) KS X 1001
ko_KR.UTF-8	한국어	한국	UTF-8	한국어(UTF-8) 유니코드 4.0
th_TH.UTF-8	영어	태국	UTF-8	태국어(UTF-8) 유니코드 4.0
th_TH.TIS620	영어	태국	TIS620.2533	태국어 TIS620.2533
zh_CN.EUC	간체 한자	PRC	gb2312 ³	간체 한자(EUC) GB2312-1980
zh_CN.GBK	간체 한자	PRC	GBK ⁴	간체 한자(GBK)
zh_CN.GB18030	간체 한자	PRC	GB18030-2000	간체 한자(GB18030-2000) GB18030-2000
zh_CN.UTF-8	간체 한자	PRC	UTF-8	간체 한자(UTF-8) 유니코드 4.0
zh_HK.BIG5HK	번체 한자	홍콩	Big5+HKSCS	번체 한자(BIG5+HKSCS)
zh_HK.UTF-8	번체 한자	홍콩	UTF-8	번체 한자(UTF-8) 유니코드 4.0
zh_TW.EUC	번체 한자	대만	cns11643	번체 한자(EUC) CNS 11643-1992
zh_TW.BIG5	번체 한자	대만	BIG5	번체 한자(BIG5)
zh_TW.UTF-8	번체 한자	대만	UTF-8	번체 한자(UTF-8) 유니코드 4.0

² PCK는 Shift_JIS (SJIS)로도 알려져 있습니다.

³ gb2312는 간체 한자 EUC 코드 세트를 의미하며 GB 1988-80 및 GB 2312-80을 포함합니다.

⁴ GBK는 GB 확장을 의미합니다. 여기에는 모든 GB 2312-80 문자 및 ISO/IEC 10646-1의 모든 Unified Han 문자, 그리고 일본어 히라가나와 가타카나가 포함됩니다. 또한 한자, 일본어 및 한국어 문자 집합과 ISO/IEC 10646-1의 많은 문자들이 포함됩니다.

표 3-2 오스트랄라시아 로캘

로캘	사용자 인터페이스	지역	코드 세트	언어 지원
en_AU.ISO8859-1	영어	호주	ISO8859-1	영어(호주)
en_NZ.ISO8859-1	영어	뉴질랜드	ISO8859-1	영어(뉴질랜드)

표 3-3 중앙 아메리카 로캘

로캘	사용자 인터페이스	지역	코드 세트	언어 지원
es_CR.ISO8859-1	스페인어	코스타리카	ISO8859-1	스페인어(코스타리카)
es_GT.ISO8859-1	스페인어	과테말라	ISO8859-1	스페인어(과테말라)
es_NI.ISO8859-1	스페인어	니카라과	ISO8859-1	스페인어(니카라과)
es_PA.ISO8859-1	스페인어	파나마	ISO8859-1	스페인어(파나마)
es_SV.ISO8859-1	스페인어	엘살바도르	ISO8859-1	스페인어(엘살바도르)

표 3-4 중앙 유럽 로캘

로캘	사용자 인터페이스	지역	코드 세트	언어 지원
cs_CZ.ISO8859-2	영어	체코 공화국	ISO8859-2	체코어(체코 공화국)
de_AT.ISO8859-1	독일어	오스트리아	ISO8859-1	독일어(오스트리아)
de_AT.ISO8859-15	독일어	오스트리아	ISO8859-15	독일어(오스트리아, ISO8859-15 - Euro)
de_CH.ISO8859-1	독일어	스위스	ISO8859-1	독일어(스위스)
de_DE.UTF-8	독일어	독일	UTF-8	독일어(독일, 유니코드 4.0)
de_DE.ISO8859-1	독일어	독일	ISO8859-1	독일어(독일)
de_DE.ISO8859-15	독일어	독일	ISO8859-15	독일어(독일, ISO8859-15 - Euro)
fr_CH.ISO8859-1	프랑스어	스위스	ISO8859-1	프랑스어(스위스)
hu_HU.ISO8859-2	영어	헝가리	ISO8859-2	헝가리어(헝가리)
pl_PL.ISO8859-2	영어	폴란드	ISO8859-2	폴란드어(폴란드)
pl_PL.UTF-8	영어	폴란드	UTF-8	폴란드어(폴란드, 유니코드 4.0)
sk_SK.ISO8859-2	영어	슬로바키아	ISO8859-2	슬로바키아어(슬로바키아)

표 3-5 동유럽 로캘

로캘	사용자 인터페이스	지역	코드 세트	언어 지원
bg_BG.ISO8859-5	영어	불가리아	ISO8859-5	불가리아어(불가리아)
et_EE.ISO8859-15	영어	에스토니아	ISO8859-15	에스토니아어(에스토니아)
hr_HR.ISO8859-2	영어	크로아티아	ISO8859-2	크로아티아어(크로아티아)

표 3-5 동유럽 로캘 (계속)

로캘	사용자 인터페이스	지역	코드 세트	언어 지원
lt_LT.ISO8859-13	영어	리투아니아	ISO8859-13	리투아니아어(리투아니아)
lv_LV.ISO8859-13	영어	라트비아	ISO8859-13	라트비아어(라트비아)
mk_MK.ISO8859-5	영어	마케도니아	ISO8859-5	마케도니아어(마케도니아)
ro_RO.ISO8859-2	영어	루마니아	ISO8859-2	루마니아어(루마니아)
ru_RU.KOI8-R	영어	러시아	KOI8-R	러시아어(러시아, KOI8-R)
ru_RU.ANSI1251	영어	러시아	ansi-1251	러시아어(러시아, ANSI 1251)
ru_RU.ISO8859-5	영어	러시아	ISO8859-5	러시아어(러시아)
ru_RU.UTF-8	영어	러시아	UTF-8	러시아어(러시아, 유니코드 4.0)
sh_BA.ISO8859-2@bosnia	영어	보스니아	ISO8859-2	보스니아어(보스니아)
sl_SI.ISO8859-2	영어	슬로베니아	ISO8859-2	슬로베니아어(슬로베니아)
sq_AL.ISO8859-2	영어	알바니아	ISO8859-2	알바니아어(알바니아)
sr_YU.ISO8859-5	영어	세르비아	ISO8859-5	세르비아어(세르비아)
tr_TR.ISO8859-9	영어	터키	ISO8859-9	터키어(터키)
tr_TR.UTF-8	영어	터키	UTF-8	터키어(터키, 유니코드 4.0)

표 3-6 중동 로캘

로캘	사용자 인터페이스	지역	코드 세트	언어 지원
He	영어	이스라엘	ISO8859-8	히브리어(이스라엘)

표 3-7 북 아프리카 로캘

로캘	사용자 인터페이스	지역	코드 세트	언어 지원
ar_EG.UTF-8	영어	이집트	UTF-8	아랍어(이집트)

표 3-7 북 아프리카 로캘 (계속)

로캘	사용자 인터페이스	지역	코드 세트	언어 지원
Ar	영어	이집트	ISO8859-6	아랍어(이집트)

표 3-8 북 미 로캘

로캘	사용자 인터페이스	지역	코드 세트	언어 지원
en_CA.ISO8859-1	영어	캐나다	ISO8859-1	영어(캐나다)
en_US.ISO8859-1	영어	미국	ISO8859-1	영어(미국)
en_US.ISO8859-15	영어	미국	ISO8859-15	영어(미국, ISO8859-15 - Euro)
en_US.UTF-8	영어	미국	UTF-8	영어(미국, 유니코드 4.0)
fr_CA.ISO8859-1	프랑스어	캐나다	ISO8859-1	프랑스어(캐나다)
es_MX.ISO8859-1	스페인어	멕시코	ISO8859-1	스페인어(멕시코)

표 3-9 북부 유럽 로캘

로캘	사용자 인터페이스	지역	코드 세트	언어 지원
da_DK.ISO8859-1	영어	덴마크	ISO8859-1	덴마크어(덴마크)
da_DK.ISO8859-15	영어	덴마크	ISO8859-15	덴마크어(덴마크, ISO8859-15-Euro)
fi_FI.ISO8859-1	영어	핀란드	ISO8859-1	핀란드어, 유니코드 4.0
fi_FI.ISO8859-15	영어	핀란드	ISO8859-15	핀란드어(핀란드, ISO8859-15-Euro)
fi_FI.UTF-8	영어	핀란드	UTF-8	핀란드어(핀란드)
is_IS.ISO8859-1	영어	아이슬란드	ISO8859-1	아이슬란드어(아이슬란드)
no_NO.ISO8859-1@bokmal	영어	노르웨이	ISO8859-1	노르웨이어(노르웨이-북말)
no_NO.ISO8859-1@nyorsk	영어	노르웨이	ISO8859-1	노르웨이어(노르웨이-니노르스크)
sv_SE.ISO8859-1	스웨덴어	스웨덴	ISO8859-1	스웨덴어(스웨덴)
sv_SE.ISO8859-15	스웨덴어	스웨덴	ISO8859-15	스웨덴어(스웨덴, ISO8859-15-Euro)
sv_SE.UTF-8	스웨덴어	스웨덴	UTF-8	스웨덴어(스웨덴, 유니코드 4.0)

표 3-10 남미 로캘

로캘	사용자 인터페이스	지역	코드 세트	언어 지원
es_AR.ISO8859-1	스페인어	아르헨티나	ISO8859-1	스페인어(아르헨티나)
es_BO.ISO8859-1	스페인어	볼리비아	ISO8859-1	스페인어(볼리비아)
es_CL.ISO8859-1	스페인어	칠레	ISO8859-1	스페인어(칠레)
es_CO.ISO8859-1	스페인어	콜롬비아	ISO8859-1	스페인어(콜롬비아)
es_EC.ISO8859-1	스페인어	에콰도르	ISO8859-1	스페인어(에콰도르)
es_PE.ISO8859-1	스페인어	페루	ISO8859-1	스페인어(페루)
es_PY.ISO8859-1	스페인어	파라과이	ISO8859-1	스페인어(파라과이)
es_UY.ISO8859-1	스페인어	우루과이	ISO8859-1	스페인어(우루과이)
es_VE.ISO8859-1	스페인어	베네수엘라	ISO8859-1	스페인어(베네수엘라)
pt_BR.ISO8859-1	영어	브라질	ISO8859-1	포르투갈어(브라질)
pt_BR.UTF-8	영어	브라질	UTF-8	포르투갈어(브라질, 유니코드 4.0)

표 3-11 남유럽 로캘

로캘	사용자 인터페이스	지역	코드 세트	언어 지원
ca_ES.ISO8859-1	영어	스페인	ISO8859-1	카탈로니아어(스페인)
ca_ES.ISO8859-15	영어	스페인	ISO8859-15	카탈로니아어(스페인, ISO8859-15 - Euro)
el_GR.ISO8859-7	영어	그리스	ISO8859-7	그리스어(그리스)
es_ES.ISO8859-1	스페인어	스페인	ISO8859-1	스페인어(스페인)
es_ES.ISO8859-15	스페인어	스페인	ISO8859-15	스페인어(스페인, ISO8859-15 - Euro)
es_ES.UTF-8	스페인어	스페인	UTF-8	스페인어(스페인, 유니코드 4.0)
it_IT.ISO8859-1	이탈리아어	이탈리아	ISO8859-1	이탈리아어(이탈리아)
it_IT.ISO8859-15	이탈리아어	이탈리아	ISO8859-15	이탈리아어(이탈리아, ISO8859-15 - Euro)
it_IT.UTF-8	이탈리아어	이탈리아	UTF-8	이탈리아어(이탈리아, 유니코드 4.0)
pt_PT.ISO8859-1	영어	포르투갈	ISO8859-1	포르투갈어(포르투갈)
pt_PT.ISO8859-15	영어	포르투갈	ISO8859-15	포르투갈어(포르투갈, ISO8859-15 - Euro)

표 3-12 서유럽 로케일

로케일	사용자 인터페이스	지역	코드 세트	언어 지원
en_GB.ISO8859-1	영어	영국	ISO8859-1	영어(영국)
en_IE.ISO8859-1	영어	아일랜드	ISO8859-1	영어(아일랜드)
fr_BE.ISO8859-1	프랑스어	벨기에-왈론	ISO8859-1	프랑스어(벨기에-왈론, 유니코드 4.0)
fr_BE.UTF-8	프랑스어	벨기에-왈론	UTF-8	프랑스어(벨기에-왈론, 유니코드 4.0)
fr_FR.ISO8859-1	프랑스어	프랑스	ISO8859-1	프랑스어(프랑스)
fr_FR.UTF-8	프랑스어	프랑스	UTF-8	프랑스어(프랑스, 유니코드 4.0)
nl_BE.ISO8859-1	영어	벨기에-플란더즈	ISO8859-1	네덜란드어(벨기에-플란더즈)
nl_NL.ISO8859-1	영어	네덜란드	ISO8859-1	네덜란드어(네덜란드)

로케일용 복수 키 구성 시퀀스

많은 Solaris 로케일들, 특히 유럽 및 유니 코드 로케일들은 Compose 키 시퀀스로 알려져 있는 “죽은 키 시퀀스”를 사용한 다양한 문자 입력을 허용합니다.

Compose 키 시퀀스는 키보드 키 캡에 표시되지 않는 구분 부호와 기타 문자를 가진 문자를 입력하는 데 사용됩니다.

다음 표는 Compose 키 시퀀스의 몇 가지 예를 보여줍니다. Compose 키 시퀀스에 관한 자세한 정보는 117 페이지 “영어/유럽어 입력 모드”를 참조하십시오.

표 3-13 Compose 키로 만든 구분 기호

기호	Compose 키 조합	예
분음 기호	”	Compose A ” → 분음 기호가 있는 A
카론	v	Compose Z v → 카론이 있는 Z
단음 기호	u	Compose G u → 단음 기호가 있는 G
오고넥	a	Compose A a → 오고넥이 있는 A
Cedilla	,	Compose K , → cedilla가 있는 K

표 3-13 Compose 키로 만든 구분 기호 (계속)

기호	Compose 키 조합	예
등록 기호	R O	Compose R O → 등록 기호
전치 느낌표	!!	Compose !! → 반전 감탄 기호

주 - compose 키 시퀀스는 문자가 현재 로케일의 코드 세트의 일부가 아닌 경우에는 문자를 생성할 수 없습니다. 예를 들어, 카논이 있는 Z는 ISO8859-1 코드 세트에 없기 때문에 en_US.ISO8859-1 로케일에서 카논이 있는 Z를 입력할 수 없습니다.

Solaris 환경에서의 키보드 지원

특정 지역에 대한 여러 레이아웃을 가진 키보드는 SPARC 및 Intel Architecture(IA) 플랫폼의 경우 지원됩니다. Solaris Operating System는 다음 표에 나열된 지역 키보드를 지원합니다.

표 3-14 지역 키보드 지원

지역	국가	Sun 키보드 (유형 4/5/5c)	Sun 키보드 (유형 6)	PC 키보드
아시아	일본	X	X	X
	한국	X	X	X
	대만	X	X	X
유럽	벨기에	X	X	X
	체코 공화국	X		X
	덴마크	X	X	X
	핀란드		X	
	프랑스	X	X	X
	독일	X	X	X
	영국	X	X	X
	그리스	X		X
	헝가리	X		X
	이탈리아	X	X	X

표 3-14 지역 키보드 지원 (계속)

지역	국가	Sun 키보드 (유형 4/5/5c)	Sun 키보드 (유형 6)	PC 키보드
	라트비아	X		X
	리투아니아	X		X
	네덜란드	X	X	X
	노르웨이	X	X	X
	폴란드	X		X
	포르투갈	X	X	X
	러시아	X	X	X
	스페인	X	X	X
	스웨덴	X	X	X
	스위스(프랑스어)	X	X	X
	스위스(독일어)	X	X	X
	터키	X	X	X
아메리카	캐나다(프랑스어)	X	X	X
	남미 (스페인어)	X		
	미국	X	X	X
중동	아랍어	X	X	

중국과 같이 국제 표준을 따르는 키보드 레이아웃을 갖고 있는 지역에서는 미국용으로 제공되는 키보드 레이아웃 지원을 이용해 로캘 문자를 입력합니다. 기반 키보드 매핑은 동일합니다. 일본, 터키 및 스위스와 같은 일부 국가들은 여러 언어를 사용하거나 복수 키보드 레이아웃이 존재하므로 여러 개의 키보드를 갖고 있습니다.

Sun 유형 4, 5 및 5c 키보드는 Mini DIN 8-핀 연결을 통해 Sun I/O 인터페이스를 사용합니다. Sun 유형 6 키보드에는 두 가지 인터페이스 버전이 있습니다.

- Mini DIN 8-핀 연결을 통한 Sun I/O
- USB

Sun 키보드 유형은 각각의 Sun 키보드 뒷면에 인쇄되어 있습니다.

PC 키보드는 PS/2나 USB와 같은 다양한 인터페이스를 사용합니다.

SPARC 시스템에서 키보드 변경하기

대부분의 Sun 유형 4, 5 및 5c 키보드에서 DIP 스위치 설정을 사용하여 Solaris 시스템의 키보드 레이아웃을 변경할 수 있습니다. DIP 스위치 설정을 위해 사용할 수 있는 키보드 유형, 이름 및 관련 레이아웃 ID는 `/usr/openwin/share/etc/keytables/keytable.map` 파일에 있습니다.

주 - 키보드 후면에 DIP 스위치가 없기 때문에 유형 6 키보드의 레이아웃을 변경할 수 없습니다. 일부 유형 5 및 5c 키보드, 예를 들어 미국, 미국/UNIX 및 일본어 키보드에는 DIP 스위치 대신 접퍼가 있습니다. `xmodmap(1)`과 같은 유틸리티 외에 SPARC 플랫폼이나 IA 플랫폼은 키보드를 전환하는 데 사용할 수 있는 유틸리티나 도구를 제공하지 않습니다.

다음은 유형 4, 5 및 5c 키보드에 대한 레이아웃 ID의 표입니다(1 = 스위치 켜기, 0 = 스위치 끄기).

표 3-15 유형 4, 5 및 5c 키보드용 레이아웃

DIP 스위치	키보드 (키 테이블 파일)	바이너리로 설정
0	미국 (US4.kt)	000000
1	미국 (US4.kt)	000001
2	벨기에(FranceBelg4.kt)	000010
3	캐나다 (Canada4.kt)	000011
4	덴마크 (Denmark4.kt)	000010
5	독일 (Germany4.kt)	000101
6	이탈리아 (Italy4.kt)	000110
7	네덜란드 (Netherland4.kt)	000111
8	노르웨이 (Norway4.kt)	000010
9	포르투갈 (Portugal4.kt)	001001
10 (0x0a)	남미/스페인어 (SpainLatAm4.kt)	001010
11 (0x0b)	스웨덴 (SwedenFin4.kt)	001011
12 (0x0c)	스위스/프랑스어 (Switzer_Fr4.kt)	000110
13 (0x0d)	스위스/독일어 (Switzer_Ge4.kt)	001011
14 (0x0e)	영국 (UK4.kt)	001110
16 (0x10)	한국 (Korea4.kt)	000010

표 3-15 유형 4, 5 및 5c 키보드용 레이아웃 (계속)

DIP 스위치	키보드(키 테이블 파일)	바이너리로 설정
17 (0x11)	대만 (Taiwan4.kt)	001001
23	러시아어	100001
33 (0x21)	미국 (US5.kt)	100111
34 (0x22)	미국/UNIX (US_UNIX5.kt)	100010
35 (0x23)	프랑스 (France5.kt)	100011
36 (0x24)	덴마크 (Denmark5.kt)	100010
37 (0x25)	독일 (Germany5.kt)	100101
38 (0x26)	이탈리아 (Italy5.kt)	100110
39 (0x27)	네덜란드 (Netherland5.kt)	100111
40 (0x28)	노르웨이 (Norway5.kt)	101000
41 (0x29)	포르투갈 (Portugal5.kt)	100101
42 (0x2a)	스페인 (Spain5.kt)	101010
43 (0x2b)	스웨덴 (Sweden5.kt)	101011
44 (0x2c)	스위스/프랑스어 (Switzer_Fr5.kt)	101101
45 (0x2d)	스위스/독일어 (Switzer_Ge5.kt)	101110
46 (0x2e)	영국 (UK5.kt)	101111
47 (0x2f)	한국 (Korea5.kt)	101111
48 (0x30)	대만 (Taiwan5.kt)	110000
49 (0x31)	일본 (Japan5.kt)	110001
50 (0x32), 참조: 63 (0x3f)	캐나다/프랑스어 (Canada_Fr5.kt)	110010
51 (0x33)	헝가리 (Hungary5.kt)	110011
52 (0x34)	폴란드 (Poland5.kt)	110100
53 (0x35)	체코 (Czech5.kt)	110101
54 (0x36)	러시아 (Russia5.kt)	110110
55 (0x37)	라트비아 (Latvia5.kt)	110111
56 (0x38) 참조: 62 (0x3e)	터키-Q5 (TurkeyQ5.kt)	000111
57 (0x39)	그리스 (Greece5.kt)	111001

표 3-15 유형 4, 5 및 5c 키보드용 레이아웃 (계속)

DIP 스위치	키보드(키 테이블 파일)	바이너리로 설정
58 (0x3a)	아랍어 (Arabic5.kt)	110111
59 (0x3b)	리투아니아 (Lithuania5.kt)	111010
60 (0x3c)	벨기에 (Belgian5.kt)	111100
62 (0x3e)	캐나다/프랑스어 (Canada_Fr5_TBITS5.kt)	111111
	프랑스 캐나다어	
	폴란드 프로그래머	
	에스토니아어	

4가 포함되어 있는 키테이블 파일명은 유형 4 키보드용입니다. 5가 포함되어 있는 키테이블 파일명은 유형 5 키보드용입니다.

▼ 체코 레이아웃으로 키보드 레이아웃을 변경하는 방법

- 테이블이나 `/usr/openwin/share/etc/keytables/keytable.mp` 파일에서 올바른 DIP 스위치 ID(또는 레이아웃 ID)를 찾습니다. `keytable.mp` 파일의 레이아웃 ID 값은 십진수 값입니다.
체코의 경우 레이아웃 ID는 십진수로 53입니다(16진수로 0x35).
- 레이아웃 ID를 이진수로 변환하거나 위 테이블의 열에 있는 적당한 이진수 설정 값을 사용하십시오. 기수 변환을 위해서는 `dtcalc(1)`과 같은 계산 유틸리티를 사용할 수 있습니다.
예를 들어, 체코 키보드용 올바른 이진 값은 110101입니다.
- 시스템을 종료한 뒤 전원을 끄십시오.
- 단계 2의 이진값을 사용해 키보드 뒷면의 DIP 스위치 설정을 변경하십시오.
첫 번째 DIP 스위치는 왼쪽에 있습니다. 1의 경우에는 위쪽으로, 0의 경우에는 아래쪽으로 옮기십시오.
체코 키보드 이진 값 110101은 다음에 해당됩니다: 위쪽 위쪽 아래쪽 위쪽 아래쪽 위쪽
- 시스템 전원을 켜 뒤 부팅시켜 사용하십시오.

주 - 유형 4 키보드와 달리 유형 5 및 5c 키보드에는 DIP 스위치가 다섯 개 뿐입니다. 유형 5 및 5c 키보드에서는 첫 번째 이진수는 무시하십시오. 체코 유형 5c 키보드의 경우 올바른 DIP 스위치 설정은 위쪽, 아래쪽, 위쪽, 아래쪽, 위쪽이며 10101에서 마지막 다섯 자리만 사용합니다.

Intel 시스템에서 키보드 변경하기

Intel 구조 시스템에서 키보드는 설치의 kdmconfig(1M) 부분 동안 선택됩니다. 설치 뒤에 변경하려면 GUI 데스크탑 환경에서 명령줄 모드로 빠져나오십시오. 슈퍼 유저로서 kdmconfig를 입력해 프로그램을 실행하십시오. 원하는 키보드 레이아웃을 얻으려면 지침에 따르십시오.

키보드 레이아웃 도표

다음 그림은 아랍어 키보드를 보여줍니다.

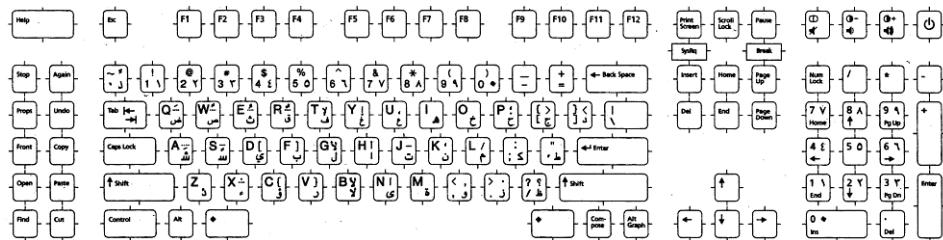


그림 3-1 아랍어 키보드

다음 그림은 벨기에 키보드를 보여줍니다.

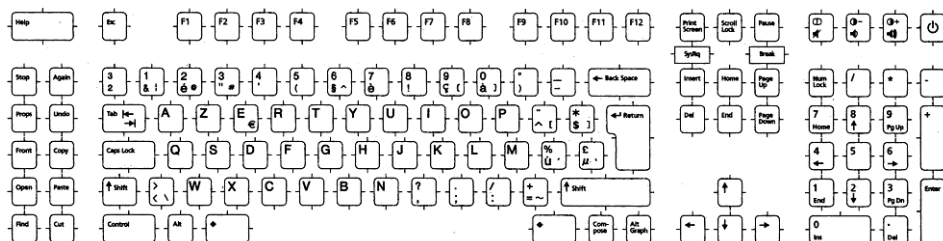


그림 3-2 벨기에 키보드

다음 그림은 키릴어 키보드를 보여줍니다.

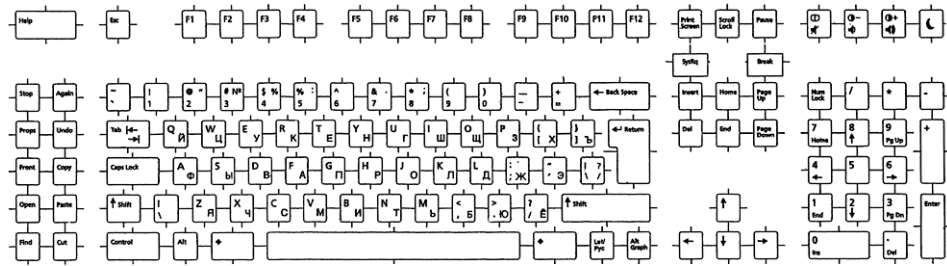


그림 3-3 키릴어(러시아어) 키보드

다음 그림은 덴마크어 키보드를 보여줍니다.

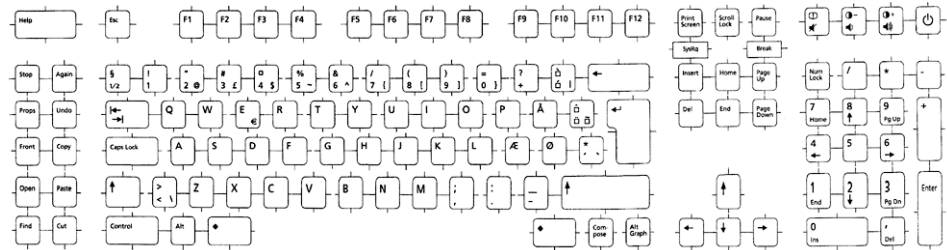


그림 3-4 덴마크어 키보드

다음 그림은 핀란드어 키보드를 보여줍니다.

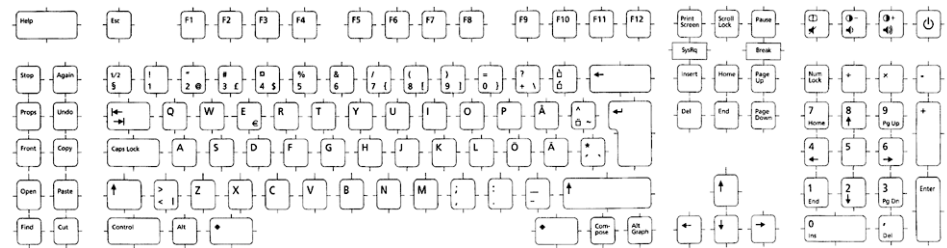


그림 3-5 핀란드어 키보드

다음 그림은 프랑스어 키보드를 보여줍니다.

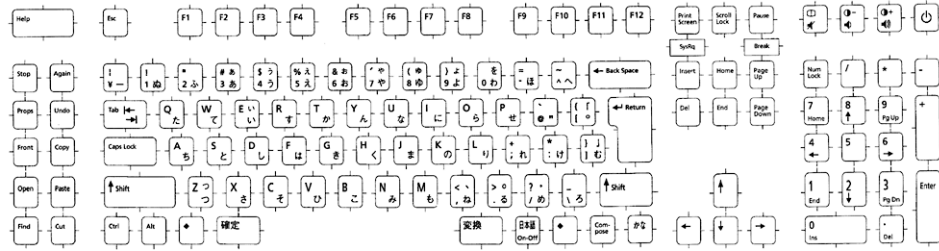


그림 3-9 일본어 키보드

다음 그림은 한국어 키보드를 보여줍니다.

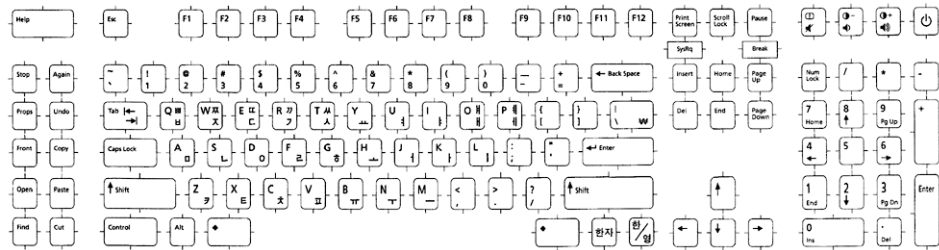


그림 3-10 한국어 키보드

다음은 네덜란드(네덜란드어) 키보드를 보여줍니다.

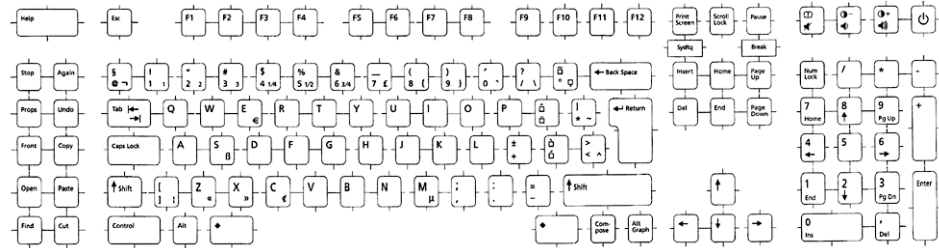


그림 3-11 네덜란드 (네덜란드어) 키보드

다음 그림은 노르웨이어 키보드를 보여줍니다.

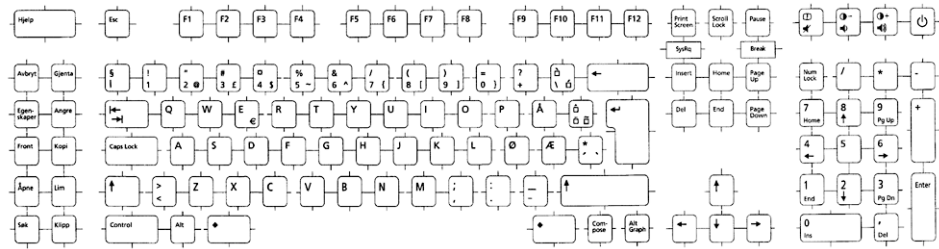


그림 3-12 노르웨이어 키보드

다음 그림은 포르투갈어 키보드를 보여줍니다.

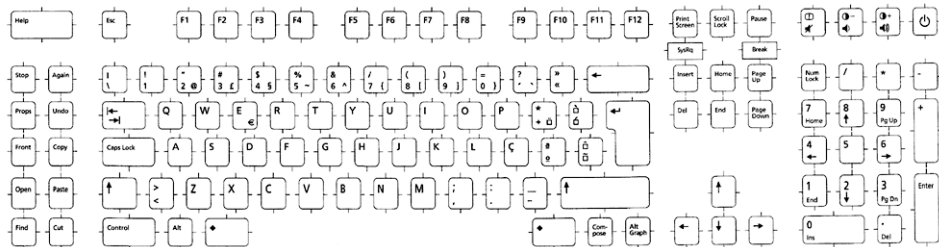


그림 3-13 포르투갈어 키보드

다음 그림은 스페인어 키보드를 보여줍니다.

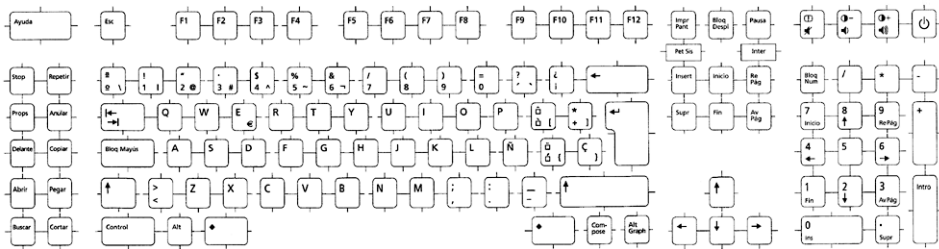


그림 3-14 스페인어 키보드

다음 그림은 스웨덴어 키보드를 보여줍니다.

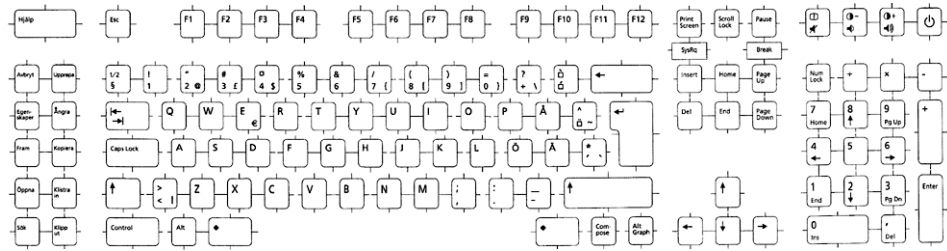


그림 3-15 스웨덴어 키보드

다음 그림은 스위스(프랑스어) 키보드를 보여줍니다.

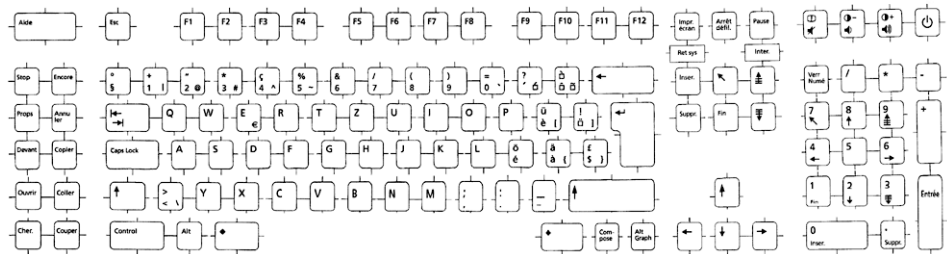


그림 3-16 스위스(프랑스어) 키보드

다음 그림은 스위스(독일어) 키보드를 보여줍니다.

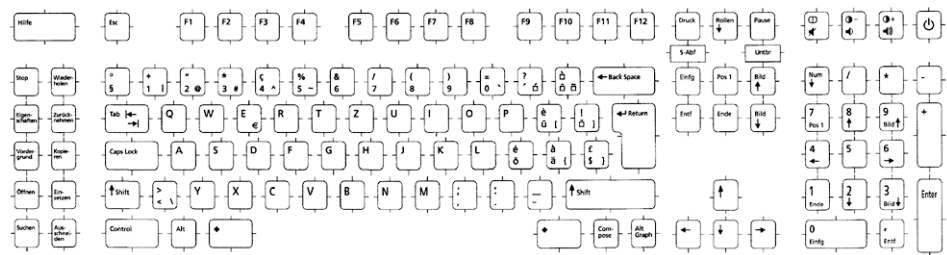


그림 3-17 스위스(독일어) 키보드

다음 그림은 정체 한자키보드를 보여줍니다.

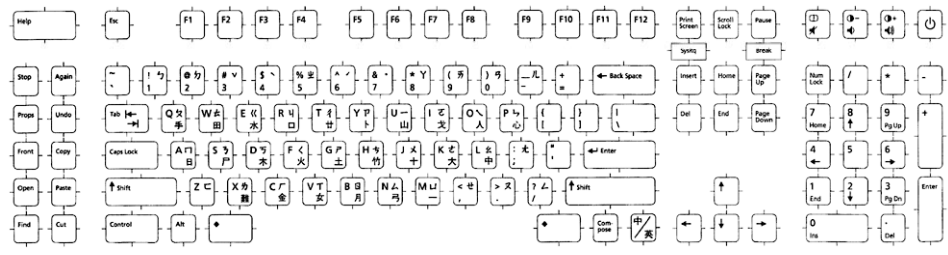


그림 3-18 정체 한자 키보드

다음 그림은 터키어 F 키보드를 보여줍니다.

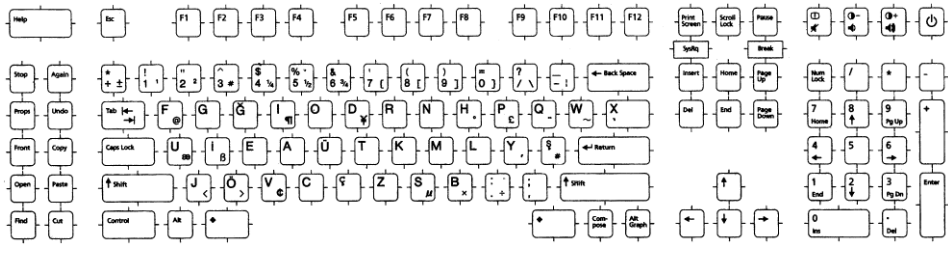


그림 3-19 터키어 F 키보드

다음 그림은 터키어 Q 키보드를 보여줍니다.

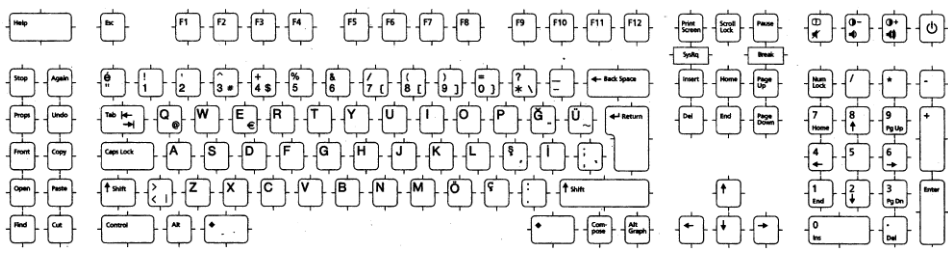


그림 3-20 터키어 Q 키보드

다음 그림은 영국 키보드를 보여줍니다.

새로운 Solaris 키보드 소프트웨어 지원

다음 추가 키보드에 대한 소프트웨어 지원이 이 릴리스에서 사용 가능합니다.

- 러시아어 유형 6 USB 키보드
- 에스토니아어 유형 6 USB 키보드
- 캐나다 프랑스어 유형 6 USB 키보드
- 폴란드 프로그래머 유형 5 키보드

소프트웨어를 사용하여 러시아, 캐나다, 에스토니아, 폴란드 사용자가 표준 미국 키보드 레이아웃을 개별 언어 요구를 충족하도록 수정할 수 있습니다. 현재 이 세 가지 추가 키보드 유형에 사용할 수 있는 하드웨어는 없습니다. 이 새로운 키보드 소프트웨어를 이용하려면 이 절의 절차 단계를 따르십시오.

▼ 에스토니아어 유형 6 USB 키보드 지원에 액세스하는 방법

1. `/usr/openwin/share/etc/keytables/keytable.map` 파일에서 `US6.kt` 항목을 `Estonia6.kt`로 변경합니다.

수정된 항목은 다음과 같이 나타납니다.

```
6          0          Estonia6.kt
```

2. `/usr/openwin/share/lib/locale/iso_8859_15/Compose` 파일에 다음 항목 중 하나를 추가합니다.

수정된 항목은 다음과 같이 나타납니다.

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

3. 변경 사항을 구현하려면 시스템을 재시동합니다.

▼ 캐나다 프랑스어 유형 6 USB 키보드 지원에 액세스하는 방법

1. `/usr/openwin/share/etc/keytables/keytable.map` 파일에서 `US6.kt` 항목을 `Canada6.kt`로 변경합니다.

수정된 항목은 다음과 같이 나타납니다.

```
6          0          Canada6.kt
```

2. 변경 사항을 구현하려면 시스템을 재시동합니다.

▼ 폴란드 프로그래머 유형 5 키보드 지원에 액세스하는 방법

1. `/usr/openwin/share/etc/keytables/keytable.map` 파일에서 `Poland5.kt` 항목을 `Poland5_pr.kt`로 변경합니다.

수정된 항목은 다음과 같이 나타납니다.

```
6          0      Poland5_pr.kt
```

2. 변경 사항을 구현하려면 시스템을 재시동합니다.

4장

지원되는 아시아권 로캘

이 장에서는 일본어, 인도어, 태국어에 대한 지역화 관련 정보를 제공합니다. 이 장의 단원은 다음과 같습니다.

- 77 페이지 “일본어 지역화”
- 81 페이지 “인도어 지역화”
- 108 페이지 “태국어 지역화”

일본어 지역화

이 절에서는 일본어 로캘 관련 정보를 설명합니다.

일본어 로캘

서로 다른 문자 코드화를 지원하는 네 가지 일본어 로캘을 현재 Solaris 환경에서 이용할 수 있습니다. ja와 ja_JP.eucJP 로캘은 일본어 EUC를 바탕으로 합니다.

ja_JP.eucJP 로캘은 UI-OSF Japanese Environment Implementation Agreement 버전 1.1을, ja 로캘은 이전 Solaris 릴리스의 전통적인 사양을 준수합니다. ja_JP.PCK 로캘은 PC-간지 코드(Shift_JIS)를 바탕으로 하며, ja_JP.UTF-8은 UTF-8을 바탕으로 합니다.

일본어 EUC와 문자 세트를 보여주는 맵은 eucJP(5) 설명서 페이지를 참조하십시오. PC-간지 코드와 문자 세트를 보여주는 맵은 PCK(5) 설명서 페이지를 참조하십시오.

일본어 문자 세트

지원되는 일본어 문자 세트는 다음과 같습니다:

- JIS X 0201-1976

- JIS X 0208-1990
- JIS X 0212-1990
- JIS X 0213-2000 (유니코드 4.0에서 정의된 문자만 해당)

JIS X 0212-1990은 ja_JP.PCK 로캘에서 지원되지 않습니다. JIS X 0213-2000은 ja_JP.UTF-8 로캘에서만 지원됩니다. JIS X 0213-2000에서 정의된 모든 문자를 사용할 수 있는 것은 아닙니다. 유니코드 4.0 문자 세트에 정의된 문자만 사용 가능합니다.

판매자 정의 문자(VDC) 및 사용자 정의 문자(UDC)도 지원됩니다. VDC는 JIS X 0208-1990 또는 JIS X 0212-1990의 미사용(예약된) 코드 지점을 점유합니다. UDC는 VDC에 대하여 할당된 코드 포인트를 제외하고 VDC와 동일한 코드 지점을 점유합니다.

일본어 글꼴

세 가지 일본어 글꼴 형식(bitmap, TrueType 및 Type1. 일본어 Type1 글꼴은 인쇄용 JIS X 0212만을 포함합니다. Type1 글꼴은 UDC에 의해서도 사용됩니다.

일본어 비트맵 글꼴은 다음 표에서 설명합니다.

표 4-1 일본어 비트맵 글꼴

전체 패밀리 이름	하위 패밀리	형식	판매업체	코드화
sun gothic	R, B	PCF(12,14,16,20,24)		JIS X 0208-1983, JIS X 0201-1976
sun minchou	R	PCF(12,14,16,20,24)		JIS X 0208-1983, JIS X 0201-1976
ricoh hg gothic b	R	PCF(10,12,14,16,18,20,24)	RICOH	JIS X 0208-1983, JIS X 0201-1976
ricoh hg mincho l	R	PCF(10,12,14,16,18,20,24)	RICOH	JIS X 0208-1983, JIS X 0201-1976
ricoh gothic	R	PCF(10,12,14,16,18,20,24)	RICOH	JIS X 0212-1990, JIS X 0213-2000
ricoh mincho	R	PCF(10,12,14,16,18,20,24)	RICOH	JIS X 0212-1990, JIS X 0213-2000
ricoh heiseimin	R	PCF(12,14,16,18,20,24)	RICOH	JIS X 0212-1990

일본어 트루 타입 글꼴은 다음 표에서 설명합니다.

표 4-2 일본어 트루 타입 글꼴

전체 패밀리 이름	하위 패밀리	형식	판매업체	코드화
ricoh hg gothic b	고정	트루 타입	RICOH	JIS X 0208-1983, JIS X 0201-1976
ricoh hg mincho l	고정	트루 타입	RICOH	JIS X 0208-.1983, JIS X 0201-1976
ricoh hg gothicb sun	고정, 비례	트루 타입	RICOH	JIS X 0201-176, JIS X 0208-1983, JIS X 0213-2000
ricoh hg minchol sun	고정, 비례	트루 타입	RICOH	JIS X 0201-1976, JIS X 0208-1983, JIS X 0213-2000
ricoh heiseimin	고정	트루 타입	RICOH	JIS X 0212-1990

일본어 입력 시스템

ATOK12는 현재 Solaris 환경의 기본 일본어 입력 시스템입니다. ATOK12는 일본어 로캘이 설치되어 있으면 모든 일본어 로캘과 UTF-8 로캘용으로 사용할 수 있습니다. Wnn6 일본어 입력 시스템도 모든 일본어 로캘에서 사용할 수 있습니다. 데스크탑 메뉴에서 입력 시스템을 전환할 수 있습니다. kkcvcv 일본어 입력 시스템은 일본어 Solaris 1.x BCP 지원용으로 사용할 수 있습니다.

다음 절차는 ATOK12 입력 방법으로 일본어 텍스트를 입력하는 방법에 대해 설명합니다.

▼ ATOK 입력 방법을 사용하는 방법

1. Control-스페이스바를 눌러 입력 변환을 설정하십시오.
2. 변환하려는 텍스트에 대하여 가나 문자를 입력하십시오.
예를 들어, 간지 henkan 대신 가나를 입력할 수 있습니다
3. 스페이스바를 눌러 가나 철자에 대신 사용 가능한 간지 변환 후보를 표시하십시오.
4. 선택하려는 변환 후보의 번호를 입력하십시오.
5. Return을 눌러 간지에 대한 완전한 가나 철자를 실행하십시오.
또는 아래쪽 화살표를 눌러 선택한 문자만 실행하십시오.
6. 스페이스바를 눌러 입력 변환을 해제하십시오.

일본어 단말기용 단말기 설정

문자 기반 단말기(TTY)에서 일본어 로캘을 사용하려면 라인 편집이 올바르게 작동되도록 단말기를 설정해 사용해야 합니다.

- 단말기가 CDE 단말기 에뮬레이터인 경우에는 (dtterm) 일본어 로캘(ja, ja_JP.PCK 또는 ja_JP.UTF-8)에서 인자 -defeucw 와 함께 stty(1)를 사용하십시오. 예를 들어, ja 로캘에서 다음을 입력합니다.

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

- 단말기가 CDE 단말기 에뮬레이터가 아니지만 터미널의 코드 세트가 현재 로캘의 코드 세트와 동일하다면 인자 -defeucw와 함께 stty(1)를 사용하십시오.
- 단말기의 코드 세트가 현재 로캘의 코드 세트와 같지 않다면 setterm(1)을 사용해 코드 변환을 활성화시킵니다. 예를 들어, ja 로캘에 있지만 단말기가 PCK (Shift_JIS code)를 필요로 하면 다음과 같이 입력합니다:

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

자세한 내용은 setterm(3CURSES) 설명서 페이지를 참조하십시오.

일본어 iconv 모듈

여러 일본어 코드 세트 변환이 iconv(1) 및 iconv(3)와 함께 지원됩니다. 자세한 내용은 iconv_ja(5) 설명서 페이지를 참조하십시오.

사용자 정의 문자 지원

사용자 정의 문자 유틸리티인 sdtudctool은 윤곽선 글꼴(Type1)과 비트맵 글꼴(PCF)을 모두 처리합니다. fontedit, type3creator 및 fontmanager와 같은 일부 유틸리티를 사용해 이전 릴리스의 구식 유틸리티로 작성된 UDC 글꼴을 이전할 수도 있습니다.

부분 로캘과 전체 로캘의 차이

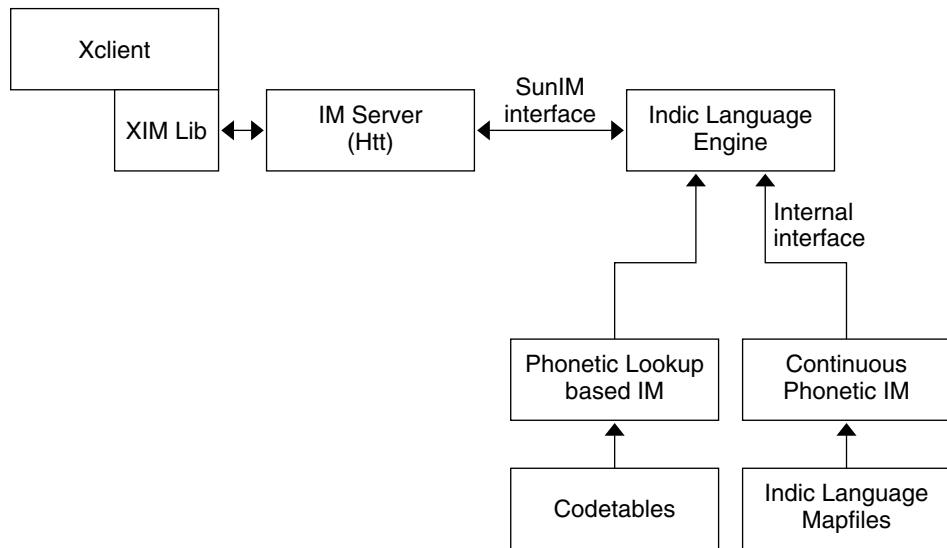
다음 구성 요소는 일본어 Languages CD가 있는 전체 로캘 환경에서만 사용할 수 있습니다:

- 번역된 메시지, 도움말 및 설명서 페이지
- Wnn6 일본어 입력 시스템
- 일본어 Solaris 1.x BCP 지원
- Mincho (min*) 서체 비트맵 글꼴
- JIS X 0212 Type1 인쇄용 글꼴
- 일본어 지향 단순 프린터 및 postprint 지원
- Kanji(1)와 같은 레거시 일본어 유틸리티

인도어 지역화

입력 방법(Shabdalipi)과 연속 음성 입력 방법을 바탕으로 하는 음성 조회는 UTF-8 로캘에서 지원되는 모든 인도어 언어에 대하여 사용 가능합니다. 입력 방법과 가상 키보드를 사용하여 모든 CDE 응용 프로그램에서 인도어 텍스트를 입력할 수 있습니다.

다음 데이터 흐름은 인도어 입력 프로세스의 작동을 예시합니다.



▼ 인도어 입력 방법을 사용하는 방법

1. 입력 상태 영역을 눌러 입력 방법 선택 메뉴를 표시하십시오.
2. 메뉴에서 입력 방법을 선택하십시오.
또는, F6 키를 눌러 사용 가능한 입력 방법 중에서 선택할 수 있습니다.
작성-hi 키 시퀀스를 입력하여 이전에 사용한 입력 방법을 선택할 수도 있습니다.
3. F5 키를 눌러 사용하려는 인도어 스크립트를 선택하십시오.
 - a. 키보드 기반(인도어 INSCRIPT 키보드) 입력 방법의 경우 82 페이지 “인도어 키보드”에 표시된 키보드 이미지를 사용하십시오.
 - b. 음성 조회 기반 입력 방법의 경우 대상 스크립트의 문자에 해당하는 첫번째 영어 음성 대응 문자를 입력하십시오.

매핑의 이해

85 페이지 “연속 음성 기반 입력 방법에 대한 매핑”의 이미지는 영어 토큰과 지원되는 각 대상 스크립트의 해당하는 코드 지점 사이의 매핑을 보여줍니다. 자음 범주는 영어 토큰과 스크립트의 자음 사이에 매핑이 이루어진다는 뜻입니다. 모음 범주는 영어 토큰과 스크립트의 모음 사이에 매핑이 이루어진다는 뜻입니다. 기타 범주는 자음과 모음의 특성을 나타내지 않는 문자의 매핑을 포함합니다(주변 문자에 따라 형태가 변하지 않음).

키워드 자음, 모음 및 기타는 이들 문자가 유니코드 표준의 일부라는 의미도 됩니다. 특수 자음, 특수 모음 또는 특수 기타 부분은 원칙적으로 이들 문자가 자음, 모음 또는 기타의 특성을 표시한다고 해도 그것들이 공식적으로 유니코드 표준의 부분이 아니고 글꼴 종속적임을 뜻합니다. 그러한 문자에는 유니코드 개인 사용자 영역의 코드 지점 값이 할당됩니다. 해당 문자들은 Solaris UTF-8 로캘에서 지원되고 다른 플랫폼에서는 매핑이 작동하지 않습니다.

이러한 맵 파일은 시스템의 맵 파일과 동일하지 않지만 이 논의의 맥락에서 불필요한 키워드를 제거하기 위해 약간 편집된 파일입니다.

모음과 특수 모음 부분에서는 독립 형태와 종속 형태가 문맥에 따라 동일한 영어 토큰에 대하여 표시됩니다. 107 페이지 “연속 음성 입력 방법이 작동하는 방법”을 참조하십시오.

malayalam 스크립트는 실제로는 특수 기타 범주인 특수 ‘CHILLU’ 부분을 포함합니다.

연속 음성 기반 입력 방법에 대한 매핑

다음 그림은 대상 인도어 스크립트의 음성 해당 문자와 영어의 기존 매핑을 보여줍니다. 사용하는 스크립트의 모든 매핑을 알 때까지 이 예를 참조로 사용하십시오. 여기서 제공된 매핑은 직관적이므로 예를 찾지 않고 대부분의 문자를 입력할 수 있습니다.

주 - 이 매핑에서 매핑의 일부로 포함된 ‘/’ 및 ‘\’와 같은 특수 문자는 ‘\’ 문자로 제어됩니다. 제어되지 않을 경우 ‘/’ 문자는 하나 이상의 토큰이 동일한 UTF-8 문자를 나타낼 때 구분자로 작용합니다.

그림 4-1, 그림 4-2 및 그림 4-3은 자음, 모음 및 기타에 대한 영어 대 Bengali 매핑을 보여줍니다.

CONSONANT

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

그림 4-1 Bengali 자음용 맵

VOWEL	Dependent form	Independent form
a	অ	আ
aa	আ	আ
i	ই	ই
ee ii	ঐ	ঐ
u	উ	উ
oo uu	ঊ	ঊ
r \ ^	ঋ	ঋ
rr \ ^	ঌ	ঌ
n \ ^	৳	৳
nn \ ^	৴	৴
e	এ	ই
ai	ঐ	ঐ
o	ও	ও
au	ঔ	ঔ

그림 4-2 Bengali 모음용 맵

OTHER

UM	ঊ
\ . N	ঋ
M	ঌ
H	঍
OU	ঔ
Rs	ঐ
Rs \ .	঑

그림 4-3 Bengali 기타용 맵

그림 4-4, 그림 4-5 및 그림 4-6은 자음, 모음 및 기타에 대한 영어 대 Gujarati 매핑을 보여줍니다.

CONSONANT

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	ઞ
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		

그림 4-4 Gujarati 자음용 맵

VOWEL	Dependent form	Independent form
a	अ	
aa	आ	।
i	इ	।
ee	ई	।
u	उ	।
oo uu	ऊ	।
r \ ^	ऋ	।
e	अ	।
E	अ	।
ai	अ	।
o	ओ	।
O	ओ	।
au	औ	।

그림 4-5 Gujarati 모음용 맵

OTHER	◌
NG	◌
M	◌
H	◌
OM	ॐ
RR	ॠ
kt	◌
av	॥

그림 4-6 Gujarati 기타용 맵

그림 4-7, 그림 4-8 및 그림 4-9 은 자음, 모음 및 기타에 대한 영어 대 Gurmukhi 매핑을 보여줍니다.

CONSONANT

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	ੜ
Th	ਠ	m	ਮ	f	ਫ਼
D	ਡ	y	ਯ		

그림 4-7 Gurmukhi 자음용 맵

VOWEL	Dependent form	Independent form
a	ਅ	
aa	ਆ	ਾ
i	ਇ	ਿ
ee ii	ਈ	ੀ
u	ਉ	ੁ
oo uu	ਊ	ੂ
E	ਏ	ਏ
ai	ਐ	ਐ
O	ਓ	ਓ
au	ਔ	ਔ

그림 4-8 Gurmukhi 모음용 맵

OTHER	
um	ੰ
\.N	ੰ
UH	ੰ
AD	ੰ
IR	ੲ
UR	ੳ
OM	ੴ

그림 4-9 Gurmukhi 기타용 맵

그림 4-10, 그림 4-11 및 그림 4-12은 자음, 모음 및 기타에 대한 영어 대 힌두어 매핑을 보여줍니다.

CONSONANT

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

그림 4-10 힌두어 자음용 맵

VOWEL	Dependent form	Independent form
a	अ	
aa	आ	।
i	इ	ि
ee	ई	ी
u	उ	ु
oo	ऊ	ू
r\^	ऋ	ॠ
rr\^	ॠ	ॡ
l\^	ऌ	ॢ
ll\^	ॡ	ॣ
EE	ए	ै
E	ए	ै
e	ए	ै
ai	ऐ	॥
OO	ऑ	॥
O	ओ	॥
o	ओ	॥
au	औ	॥

그림 4-11 힌두어 모음용 맵

OTHER			
OM	ॐ		U\~
\.C	.		A\^
M	.		A\~
H	.		\
\.N	.		\ \
V\^	ॡ		EH
U\^	.		

그림 4-12 힌두어 기타용 맵

그림 4-13, 그림 4-14 및 그림 4-15은 자음, 모음 및 기타에 대한 영어 대 Kannada 매핑을 보여줍니다.

CONSONANT

k | Kh | K ಕ ಀ
 g ಗ ಙ
 G | gh ಗ ಙ
 \~G ಳ ಴
 c | ch ಚ ಛ
 C | CH ಚ ಛ
 j ಜ ಝ
 J | jh ಜ ಝ
 \~J ಞ ಠ
 T ಡ ಢ
 Th ಡ ಢ
 D ಢ ಣ

Dh ಧ ಧ
 N ಢ ಣ
 t ತ ಠ
 th ತ ಠ
 d ದ ಢ
 dh ದ ಢ
 n ನ ಢ
 p ಪ ಪ
 p | ph ಪ ಪ
 b ಬ ಬ
 B | bh ಬ ಬ
 m ಮ ಮ

y ಯ ಯ
 R | r ರ ರ
 rx | rh ರ ರ
 l ಲ ಲ
 ll ಲ ಲ
 w | v ವ ವ
 S | sh ಶ ಶ
 Sh ಶ ಶ
 s ಸ ಸ
 \~h | h ಹ ಹ
 f ಫ ಫ

그림 4-13 Kannada 자음용 맵

VOWEL	Dependent form	Independent form
a	ಅ	
aa	ಆ	ೌ
i	ಇ	ಃ
ee	ಈ	ಃ
u	ಉ	ಃ
U oo	ಊ	ಃ
r\^	ಋ	
R\^	ೠ	ಃ
e	ಏ	ಃ
E	ಐ	ಃ
ai	ಏ	ಃ
o	ಓ	ಃ
O	ಔ	ಃ
au ou	ಔ	ಃ

그림 4-14 Kannada 모음용 맵

OTHER

M	೦
H	ಃ
OU	ೳ
LM	ೳ
RR \ ^	ಋ
\ ~N	ೠ

그림 4-15 Kannada 기타용 맵

그림 4-16, 그림 4-17 및 그림 4-18은 자음, 모음 및 기타에 대한 영어 대 Malayalam 매핑을 보여줍니다.

CONSONANT

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	SPECIAL CONSONANT	
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		

그림 4-16 Malayalam 자음용 맵

VOWEL	Dependent form	Independent form
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	ൌ
SPECIAL VOWEL		
ou	ഔ	ൗ

그림 4-17 Malayalam 모음용 맵

OTHER			CHILLU	
M	o		n \ ~	ൻ
H	ഃ		N \ ~	ൺ
rr \ ^	ഋ		l \ ~	ൽ
U	ൺ		L \ ~	ൾ
UU	൳		r \ ~	ർ

그림 4-18 Malayalam 기타용 맵

그림 4-19 및 그림 4-20는 자음 및 모음에 대한 영어 대 타밀어 매핑을 보여줍니다.

CONSONANT

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

그림 4-19 타밀어 자음용 맵

VOWEL	Dependent form	Independent form
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
OTHER		
H	ஃ	

그림 4-20 타밀어 모음용 맵

그림 4-21, 그림 4-22 및 그림 4-23은 자음, 모음 및 기타에 대한 영어 대 Telugu 매핑을 보여줍니다.

CONSONANT

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

그림 4-21 Telugu 자음용 맵

VOWEL	Dependent form	Independent form
a	అ	
A aa	ఆ	ౌ
i	ఇ	ః
I ia ee ii	ఈ	ః
u	ఉ	ః
ua U oo uu	ఊ	ృ
R	ఋ	ౠ
Ru	ౡ	ౣ
E ae ea	ఎ	ః
ai	ఐ	౦
o	ఒ	ః
oa oe O	ఓ	౦
ou au	ఔ	౦

그림 4-22 Telugu 모음용 맵

OTHER	
\ . N	౯
M	౦
H	ః
OU	౹
LM	౬
RR	ఋ
Nu	ౠ

그림 4-23 Telugu 기타용 맵

연속 음성 입력 방법이 작동하는 방법

각 인도어 스크립트의 경우 'virama' 또는 자음과 결합된 동등한 기호는 자음의 반형태 (또는 형태 결합이 준비된)를 제공합니다. 자음에 상응하는 여러 키 조합이 입력될 때마다 자음 + virama 형태가 출력되어 문자가 결합될 준비가 되었음을 나타냅니다.

자음은 초기 입력시 반형태를 나타내고 모음이 뒤에 오면 완전한 음절이나 변화형이 됩니다.

두 개의 연속된 자음은 반형태 결합 준비 상태로 남습니다. 반형태는 레이아웃 엔진에 의해 단일 결합 문자로 변환되거나 모든 언어에 대하여 통사적으로도 유효한 독립 형태로 남을 수 있습니다.

단어의 시작을 이루거나 뒤에 또 다른 모음이 오는 모음은 독립 형태로 나타납니다. 자음 바로 뒤에 오는 모음은 종속 형태로 나타납니다.

어떤 컨텍스트에서도 모양이 변하지 않는 문자를 기타라고 합니다. 이런 문자들은 자음도 모음도 아닙니다.

숫자와 문자의 일부를 이루지 않는 기타 구두점은 일대일로 매핑됩니다.

이러한 원칙을 사용하여 입력을 이러한 여러 범주로 구문 분석하고 언어 특정 유니코드 코드 지점으로 출력하는 구문 분석기가 작성됩니다. 연속 음성 입력 방법 엔진은 시스템의 다른 모듈에 의한 레이아웃이나 렌더링을 처리하지 않습니다.

태국어 지역화

현재 Solaris 환경은 세 가지 태국어 입력 레벨과 네 가지 태국어 키보드 레이아웃을 지원합니다.

태국어 입력 방법

다음 태국어 입력 방법은 이 릴리스에서 지원됩니다. 이러한 입력 방법은 문자 시퀀스 검사용 태국어 IT 표준에서 지정되었습니다.

1. 통과 레벨, 입력 검사 하지 않음.
2. 기본 입력 검사 레벨.
3. 엄격한 입력 검사 레벨.

시퀀스 검사가 없는 통과 레벨은 이전 Solaris 릴리스에서와 같이 이 릴리스에서 기본값입니다.

F2 기능 키를 사용하여 한 입력 레벨에서 다음 입력 레벨로 전환할 수 있습니다.

태국어 키보드 레이아웃

네 가지 다른 키보드 레이아웃이 태국어 입력 방법에 대하여 지원됩니다.

- **Kedmanee (TIS820-2531) 키보드 레이아웃.** Kedmanee 레이아웃은 컴퓨터 키보드가 아닌 타이프라이터용으로 설계되었습니다. 타이프라이터 키보드의 키 수가 제한되었다는 것은 일부 태국어 특수 문자는 이 레이아웃에서 사용할 수 없다는 뜻입니다. TIS820-2531은 컴퓨터 키보드와 함께 사용하기 위해 Kedmanee 레이아웃을 채택했습니다.



- TIS820-2538 키보드 레이아웃. 이 향상된 Kedmanee 레이아웃은 원래 Kedmanee 레이아웃에서 사용할 수 없었던 일부 태국어 특수 문자를 포함하는 TIS820-2531 레이아웃의 업데이트 버전입니다. 현재, TIS820-2538은 태국 산업 표준 협회에서 발행한 유일한 태국어 키보드 레이아웃 표준입니다.



- Pattajoti 키보드 레이아웃. Pattajoti 레이아웃도 타이프라이터용으로 설계되었지만 손가락 하중 배분이 더 우수합니다.



- 구성 가능한 키보드 레이아웃. 태국어 입력 방법용 사용자 정의 키보드 레이아웃.

태국어 입력 방식 보조 창

태국어 입력 방법 보조 창은 다음 기능과 유틸리티를 지원합니다.

- 입력 레벨 전환. 보조 팔레트의 입력 레벨 버튼을 눌러 통과, 기본 또는 엄격한 입력 레벨 중에서 선택할 수 있습니다.
- 태국어 가상 키보드 키보드 버튼을 눌러 태국어 문자를 입력하는 데 사용하는 태국어 가상 키보드를 표시할 수 있습니다.

UTF-8 로케일 지원의 개요

이 장은 UTF-8 로케일 지원의 개요를 제공합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- 111 페이지 “유니코드의 개요”
- 112 페이지 “유니코드 로케일: en_US.UTF-8 지원”
- 114 페이지 “데스크탑 입력 방법에 대하여”
- 146 페이지 “시스템 환경”
- 150 페이지 “코드 변환”
- 151 페이지 “DtMail 지원”
- 154 페이지 “프로그래밍 환경”

유니코드의 개요

유니코드는 컴퓨터 처리를 위한 텍스트 표현에 사용되는 문자 코드화 표준입니다. 유니코드는 국제 표준 ISO/IEC 10646-1:2000 및 ISO/IEC 10646-2:200을 완전 준수하고 ISO/IEC 10646와 동일한 모든 문자 및 코드화 포인트를 포함합니다. 유니코드 표준은 문자와 문자 사용에 대한 추가 정보를 제공합니다. 유니코드를 준수하는 모든 구현은 ISO/IEC 10646도 준수합니다.

유니코드는 다국어 일반 텍스트 인코딩의 일관성 있는 방식을 제공하고 국제 텍스트 파일 교환을 촉진합니다. 다국어 텍스트를 사용하는 컴퓨터 사용자, 비즈니스맨, 언어학자, 연구자, 과학자 등은 유니코드 표준이 업무를 대단히 간소화한다고 생각합니다. 정기적으로 수학 기호와 기타 기술 문자를 사용하는 수학자와 기술자도 유니코드 표준을 가치 있다고 여깁니다.

유니코드는, 17개의 16비트 플레인을 통해 최대 1,114,112개의 코드 포인트를 지원할 수 있습니다. 각 플레인은 65,536개의 다른 코드 포인트를 가집니다.

유니코드가 지원할 수 있는 백만 개 이상의 코드 포인트 가운데 버전 4.0은 현재 플레인 0, 1, 2 및 14에서 96,382개 문자를 정의합니다. 플레인 15와 16은 전용 사용 문자로 사용자 정의 문자라고도 합니다. 플레인 15와 16은 총 131,068개까지의 사용자 정의 문자를 지원할 수 있습니다.

유니코드는 다음 문자 코드화 체계 중 하나를 사용하여 코드화될 수 있습니다.

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

UTF-8은 ASCII 문자 코드 값을 그대로 보존하는, 유니코드 가변폭 인코딩 방식입니다. 이 형식은 Solaris 유니코드 로케에서 파일을 저장하는 방식으로 쓰입니다.

UTF-16은 유니코드의 16비트 인코딩 형식입니다. UTF-16에서는, 65,536까지의 문자는 하나의 16비트 값으로 인코딩됩니다. 65,536 이상부터 1,114,111까지의 코드 포인트에 해당되는 문자들은 두 개의 16비트 값이 한 쌍이 되어 인코딩됩니다.(surrogates라고도 합니다.)

UTF-32는 유니코드의 고정폭, 21비트 인코딩 방식으로 대개 32비트 저장 또는 데이터 유형으로 표현됩니다. 이 형식은 Solaris 유니코드 로케에서 프로세스 코드(와이드 캐릭터 코드)로 사용됩니다.

유니코드 표준 및 ISO/IEC 10646과 다양한 표현 형식에 대한 자세한 내용은 다음 소스를 참조하십시오.

- 유니코드 컨소시엄의 *The Unicode Standard, Version 4.0*
- ISO/IEC 10646-1:2000, Information Technology-Universal Multiple-Octet Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane
- ISO/IEC 10646-2: Information Technology-Universal Multiple-Octet Character Set (UCS) - Part 2: Secondary Multilingual Plane for Scripts and Symbols, Supplementary Plane for CJK Ideographs, Special Purpose Plane
- 유니코드 컨소시엄 웹 사이트는 <http://www.unicode.org/>입니다.

유니코드 로케: en_US.UTF-8 지원

유니코드/UTF-8 로케는 유니코드 4.0을 지원합니다. en_US.UTF-8 로케는 UTF-8을 코드 세트로 사용하여 다중 스크립트 처리 지원을 제공합니다. 이 로케는 다중 스크립트의 텍스트 입출력을 지원하며, Solaris Operating System에서 이 기능을 가진 첫번째 로케였습니다. 다른 UTF-8 로케의 기능은 en_us.UTF-8과 유사합니다. 다음 en_US.UTF-8의 논의는 이러한 로케와 동일하게 적용됩니다.

주 - UTF-8은 1992년 X/Open-Uniform Joint Internationalization Working Group (XoJIG)이 체계화하고 1996년 ISO 및 IEC가 Amendment 2 to ISO/IEC 10646-1:1993로 승인한, 파일 시스템 안정적인 유니코드/ISO/IEC 10646-1의 Universal Character Set Transformation Format입니다. 이 표준은 유니코드 Consortium, International Standards Organization 및 International Electrotechnical Commission에 의해 유니코드 4.0 및 ISO/IEC 10646-1의 일부로 채택되었습니다.

Solaris 환경의 유니코드 로케일은 유니코드 4.0 및 ISO/IEC 10646-1 및 10646-2에 정의된 모든 코드 포인트 값의 처리를 지원합니다. 지원되는 스크립트는 범 유럽 및 아시아 스크립트와 아랍어, 히브리어, 인도어 및 태국어와 같은 복잡한 텍스트 레이아웃 스크립트도 포함합니다.

주 - 일부 유니코드 로케일, 특히 아시아 로케일은 보다 많은 Kanji 또는 Hanzi 모양을 포함합니다.

제한된 글꼴 자원으로 인해 현재 Solaris 유니코드 로케일은 다음 문자 세트의 문자 모양만 포함합니다.

- ISO 8859-1 (영어, 프랑스어, 스페인어, 독일어와 같은 대부분의 서유럽 언어)
- ISO 8859-2 (체코어, 폴란드어, 헝가리어와 같은 대부분의 중부 유럽 언어)
- ISO 8859-4 (스칸디나비아어와 발트어)
- ISO 8859-5 (러시아어)
- ISO 8859-6 (아랍어, 더 많은 표시 형식 문자 모양 포함)
- ISO 8859-7 (그리스어)
- ISO 8859-8 (히브리어)
- ISO 8859-9 (터키어)
- TIS 620.2533 (태국어, 더 많은 표시 형식 문자 모양 포함)
- ISO 8859-15 (유럽 기호가 있는 대부분의 서부 유럽권 언어)
- GB 2312-1980 (간체 한자)
- JIS X 0201-1976, JIS X 0208-1990 (일본어)
- KS X 1001: 1992 부속서 3 (한국어)
- GB 18030 (간체 한자)
- HKSCS (정체 한자, 홍콩)
- Big5 (정체 한자, 대만)
- ISCII라고도 하는 IS 13194.1991(힌디어, 더 많은 표시 형식 문자 모양 포함)

만약 en_US.UTF-8 로케일에서, 해당 글리프가 없는 문자를 보려고 하면, 로케일은 아래와 같이 no-glyph라는 메시지를 표시합니다.



로켈은 설치시 선택 가능하며 시스템 기본 로켈로 지정될 수 있습니다.

64비트와 32비트의 Solaris 시스템에서, en_US.UTF-8과 동일한 수준의 로켈 지원이 제공됩니다.

주 - Motif 및 CDE 데스크탑 응용프로그램과 라이브러리는 en_US.UTF-8 로켈을 지원합니다. 그러나, XView™ 및 OLIT 라이브러리는 en_US.UTF-8 로켈을 지원하지 않습니다.

데스크탑 입력 방법에 대하여

CDE는 Xm 툴킷을 사용하여 국제화된 응용 프로그램에 대하여 지역화된 입력을 입력하는 기능을 제공합니다. XmText [Field] 위젯트는 각 로켈의 입력 방식과의 인터페이스가 가능합니다. 어떤 언어 환경에서는 텍스트를 오른쪽에서 왼쪽으로, 위에서 아래로 등등의 방식으로 쓰기 때문에 입력 방식을 국제화합니다. 동일한 응용프로그램 내에서 여러 글꼴을 적용하는 다른 입력 방식을 사용할 수 있습니다.

사전 편집 영역은 사전 편집 중인 문자열을 표시합니다. 텍스트 작성은 다음 4가지 모드로 이루어질 수 있습니다.

- OffTheSpot
- OverTheSpot (기본값)
- 루트
- 없음

OffTheSpot 모드에서의 위치는 상태 영역의 오른쪽에 있는 기본 창 영역 바로 아래입니다. OverTheSpot 모드에서 사전 편집 영역은 커서 포인트에 있습니다. 루트 모드에서 사전 편집 및 상태 영역은 클라이언트 창과 떨어져 있습니다.

자세한 내용은 VendorShell(3X) 설명서 페이지의 XmNpreeditType 자원 설명을 참조하십시오.

주 - 현재 Solaris 환경에서 원 아시아권 입력 방식은 간체/정체 한자, 일본어 및 한국어에 대하여 존재합니다. 이러한 메소드는 유니코드 로캘의 현재 다중 스크립트 입력 방식에 추가됩니다.

115 페이지 “입력 방식 액세스하기”는 선택한 입력 방법 설명, 사용법, 방법 전환법을 포함합니다.

스크립트 선택 및 입력 모드

Solaris 유니코드 로캘은 여러 스크립트를 지원합니다. 모든 유니코드 로캘에는 총 14개의 입력 모드가 있습니다.

- 영어/유럽어
- 키릴 문자
- 그리스어
- 아랍어
- 히브리어
- 태국어
- 일본어
- 한국어
- 중국어 간체
- 중국어 번체
- 정체 한자(홍콩)
- 인도어
- 유니 코드 16진 및 8진코드 입력 방법
- 테이블 조회 입력 방식

입력 방식 액세스하기

Compose 키 조합 또는 입력 모드 선택 창을 통해 특정 입력 모드로 전환이 가능합니다. 입력 모드 선택 창에 액세스하려면 응용프로그램 창의 맨아래 왼쪽 모서리에 있는 상태 영역을 누릅니다. 입력 모드 선택 창은 다음 그림과 같습니다.

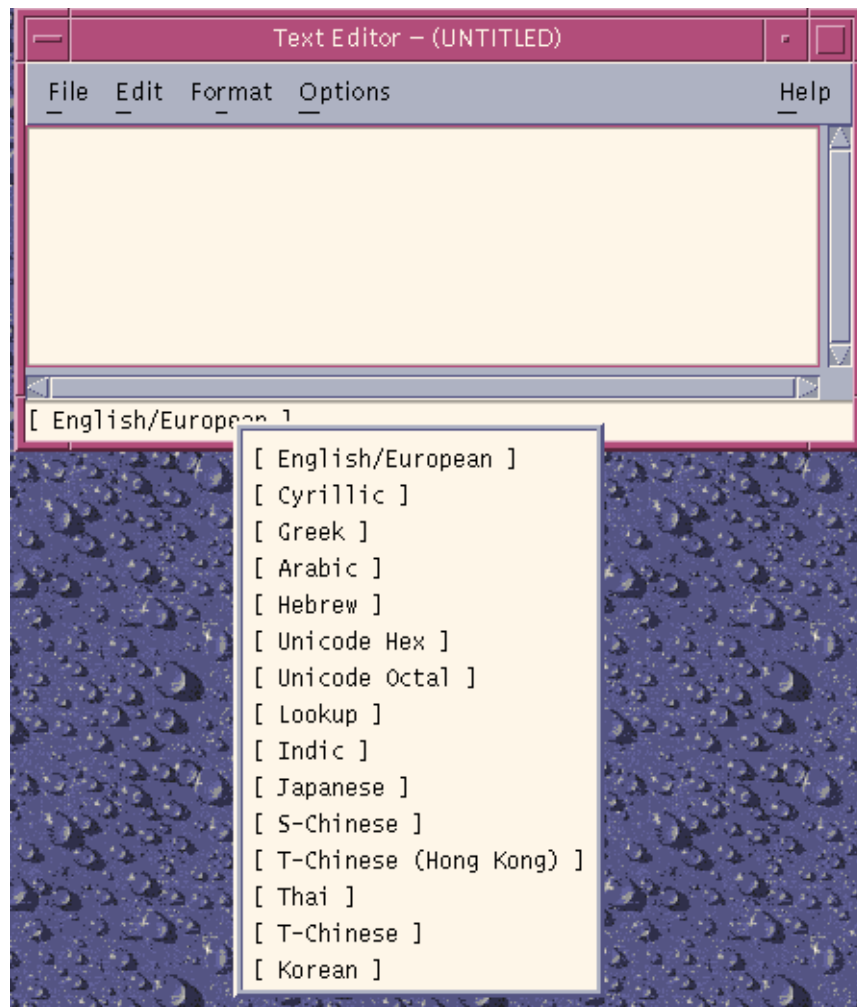


그림 5-1 입력 모드 선택 창

입력 모드 스위치 키 시퀀스

표 5-1에 나열된 키 시퀀스를 사용하여 현재 입력 모드를 새 입력 모드로 변경할 수 있습니다. 키 시퀀스 사용에 대한 유일한 제약은 아시아어 입력 모드 중 하나에 있을 경우 **Control**과 **스페이스바**를 동시에 눌러 영어/유럽어 입력 모드로 다시 전환해야 하는 것입니다. 영어/유럽어 입력 모드에 있으면 키 시퀀스를 사용하여 다른 입력 모드로 자유롭게 전환할 수 있습니다.

다음 키 시퀀스는 영어/유럽어 입력 모드에서 키릴 문자로 전환하는 방법을 보여줍니다.

1. Compose 키를 누릅니다.
2. C 키를 눌렀다 놓습니다.
3. C 키를 누릅니다.

표 5-1 입력 모드 스위치 키 시퀀스

키 시퀀스	입력 모드
Control-스페이스바	영어/유럽어
Compose c c	키릴 문자
Compose g g	그리스어
Compose a r	아랍어
Compose h h	히브리어
Compose t t	태국어
Compose h i	인도어
Compose i n	인도어
Compose j a	일본어
Compose k o	한국어
Compose s c	중국어 간체
Compose t c	중국어 번체
Compose h k	정체 한자(홍콩)
Compose u o	유니 코드 8진 코드 입력 방법
Compose u h	유니 코드 16진 코드 입력 방법
Compose l l	테이블 조회 입력 방식

영어/유럽어 입력 모드

영어/유럽어 입력 모드는 영어 알파벳과 구별 부호가 있는 문자(예를 들어, á, è, î, ò, ü), 유럽 스크립트의 문자(예를 들어, ÿ, ſ, ç)를 포함합니다.

이 입력 모드는 모든 응용프로그램에 대한 기본 모드입니다. 입력 모드는 GUI 응용프로그램 창의 맨아래 왼쪽 모서리에서 표시됩니다.

구별 표시가 있는 문자 또는 라틴-1, 라틴-2, 라틴-3, 라틴-4, 라틴-5 및 라틴-6의 특수 문자를 삽입하려면 다음 예제의 설명과 같이 Compose 키 시퀀스를 입력해야 합니다.

ç 문자를 표시하려면:

1. Compose 키를 눌렀다 놓습니다.
2. Shift와 A를 동시에 누릅니다. Shift-A를 놓습니다.
3. " 키를 눌렀다 놓습니다.

¡, 문자를 표시하려면:

1. Compose 키를 눌렀다 놓습니다.
2. ?를 눌렀다 놓습니다.
3. ?를 눌렀다 놓습니다.

키보드에 사용 가능한 Compose 키가 없으면 Control 키와 Shift 키를 동시에 눌러 해당 작업을 에뮬레이트할 수 있습니다.

로캘에서 유럽 통화 기호(유니코드 값 U+20AC)의 입력을 위해서는 다음 입력 시퀀스 중 하나를 사용할 수 있습니다.

- AltGraph를 누른채 E 키를 함께 누름
- AltGraph를 누른채 4 키를 함께 누름
- AltGraph를 누른채 5 키를 함께 누름

이러한 입력 시퀀스로 두 키를 동시에 누릅니다. 키보드에서 AltGraph 키 사용할 수 없을 경우 Compose e = 또는 Compose c =와 같은 특정한 대체 유로 기호 입력 시퀀스를 사용할 수 있습니다.

다음 표는 Solaris Operating System에 대하여 가장 공통적으로 사용되는 라틴-1, 라틴-2, 라틴-3, 라틴-4, 라틴-5 및 라틴-9의 작성 시퀀스를 보여줍니다.

다음 표는 공통 라틴-1 Compose 키 시퀀스를 나열합니다.

표 5-2 공통 Latin-1 Compose 키 시퀀스

>Compose 키를 누른 후, 아래 키를 눌렀다 놓음	아래 키를 눌렀다 놓습니다.	결과
스페이스바	스페이스바	중단 없는 공간
s	1	위첨자 1
s	2	위첨자 2
s	3	위첨자 3
!	!	전치 느낌표
x	o	통화 기호 ¢
p	!	문단 기호
/	u	mu u
'	"	예음 악센트
,	, (침표)	세디유

표 5-2 공통 Latin-1 Compose 키 시퀀스 (계속)

>Compose 키를 누른 후, 아래 키를 눌렀다 놓음	아래 키를 눌렀다 놓습니다.	결과
	"	분음 기호
-	^	장음 기호
o	o	각도 °
x	x	곱셈 부호 x
+	-	플러스-마이너스
-	-	짧은 하이픈-
-	:	분리 부호÷
-	a	서수(여성) ^a
-	o	서수(남성) ^a
-	, (선표)	부호 없음
.	.	가운데 점
1	2	상분수
1	4	상분수
3	4	상분수
<	<	왼쪽 이중 꺾쇠 인용 표시
>	>	오른쪽 이중 꺾쇠 인용 표시
✓	✓	전치 물음표
A	' (반대쪽 인용)	억음 기호?
A	' (단일 인용)	예음?
A	*	위쪽 링
A		분음 부호?
A	^	?곡절 부호
A	~	틸데
A	E	AE 복모음 Æ
C	, (선표)	C 세디유?
C	o	저작권 기호
D	-	대문자 eth ð
E	' (반대쪽 인용)	E 억음 기호?

표 5-2 공통 Latin-1 Compose 키 시퀀스 (계속)

>Compose 키를 누른 후, 아래 키를 눌렀다 놓음	아래 키를 눌렀다 놓습니다.	결과
E	'	E 예음 기호?
E		E 분음 기호?
E	^	E 곡절 기호
I	' (반대쪽 인용)	I 억음 기호?
I	'	I 예음 기호?
I	"	I 분음 기호?
I	^	I 곡절 기호
L	-	파운드 기호?
N	~	N 틸데
O	' (반대쪽 인용)	O 억음 기호?
O	'	O 예음 기호?
O	/	O 슬래시 Ø
O	"	O 분음 기호?
O	^	O 곡절 기호?
O	~	O 틸데
R	o	등록 상표
T	H	손
U	' (반대쪽 인용)	U 억음 기호?
U	'	U 예음 기호?
U		U 분음 기호?
U	^	U 곡절 기호?
Y	'	Y 예음 기호?
Y	-	옌 기호?
a	' (반대쪽 인용)	a 억음 기호?
a	'	a 예음 기호?
a	*	a 위쪽 링?
a		a 분음 기호
a	~	a 틸데

표 5-2 공통 Latin-1 Compose 키 시퀀스 (계속)

>Compose 키를 누른 후, 아래 키를 눌렀다 놓음	아래 키를 눌렀다 놓습니다.	결과
a	^	a 곡절 기호
a	e	ae 복모음 Æ
c	, (첨표)	c 세디유
c	/	센트 기호?
c	o	저작권 기호
d	-	eth ð
e	' (반대쪽 인용)	e 억음 기호
e	'	e 예음 기호
e		e 분음 기호
e	^	e 곡절 ?기호
i	' (반대쪽 인용)	i 억음 기호?
i	'	i 예음 기호?
i		i 분음 기호?
i	^	i 곡절 기호?
n	~	n 틸데
o	' (반대쪽 인용)	o 억음 기호?
o	'	o 예음 기호?
o	/	o 슬래시 ø
o		o 분음 기호?
o	^	o 곡절 기호?
o	~	o 틸데?
s	s	독일어 이중 s는 경음 S라고도 함
t	h	손
u	' (반대쪽 인용)	u 억음 기호?
u	'	u 예음 기호?
u		u 분음 기호?
u	^	u 곡절 기호?
y	'	y 예음 기호 y

표 5-2 공통 Latin-1 Compose 키 시퀀스 (계속)

>Compose 키를 누른 후, 아래 키를 눌렀다 놓음	아래 키를 눌렀다 놓습니다.	결과
y		y 분음 기호?
		분리된 수선?

다음 표는 공통 Latin-2 Compose 키 시퀀스를 나열합니다.

표 5-3 공통 Latin-2 Compose 키 시퀀스

Compose 키를 누른 후, 아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
k	k	kra
A	—	A 장음 기호
E	—	E 장음 기호
E	.	E 위쪽 점
G	,	G 세디유
I	—	I 장음 기호
I	~	I 틸데
I	a	I ogonek
K	,	K 세디유
L	,	L 세디유
N	,	N 세디유
O	—	O 장음 기호
R	,	R 세디유
T		T 스트로크
U	~	U 틸데
U	a	U ogonek
U	—	U 장음 기호
N	N	Eng
a	—	a 장음 기호
e	—	e 장음 기호
e	.	e 위쪽 점
g	,	g 세디유

표 5-3 공통 Latin-2 Compose 키 시퀀스 (계속)

Compose 키를 누른 후, 아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
i	—	i 장음 기호
i	~	i 틸데
i	a	i ogonek1
k	,	k 세디유
l	,	l 세디유
n	,	n 세디유
o	—	o 장음 기호
r	,	r 세디유
t		t 스트로크
u	~	u 틸데
u	a	u ogonek
u	—	E 장음 기호
n	n	eng

다음 표는 공통 Latin-3 Compose 키 시퀀스를 나열합니다.

표 5-4 공통 라틴-3 Compose 키 시퀀스

Compose 키를 누른 후, 아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
C	>	C 곡절 기호
C	.	C 위쪽 점
G	>	G 곡절 기호
G	.	G 위쪽 점
H	>	H 곡절 기호
J	>	j 곡절 기호
S	>	S 곡절 기호
U	u	U 약음 기호
c	>	c 곡절 기호
c	.	c 위쪽 점

표 5-4 공통 라틴-3 Compose 키 시퀀스 (계속)

Compose 키를 누른 후, 아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
g	>	g 곡절 기호
g	.	g 위쪽 점
h	>	h 곡절 기호
j	>	j 곡절 기호
s	>	s 곡절 기호
u	u	u 약음 기호

다음 표는 공통Latin-4 Compose 키 시퀀스를 나열합니다.

표 5-5 공통 Latin-4 Compose 키 시퀀스

>Compose 키를 누른 후, 아래 키를 눌렀다 놓음	아래 키를 눌렀다 놓습니다.	결과
k	k	kra
A	-	A 장음 기호
E	-	E 장음 기호
E	.	E 위쪽 점
G	,	G 세디유
I	-	I 장음 기호
I	~	I 틸데
I	a	Iogonek
K	,	K 세디유
L	,	L 세디유
N	,	N 세디유
O	-	O 장음 기호
R	,	R 세디유
T		T 스트로크
U	~	U 틸데
U	a	Uogonek
U	-	U 장음 기호
N	N	Eng

표 5-5 공통 Latin-4 Compose 키 시퀀스 (계속)

>Compose 키를 누른 후, 아래 키를 눌렀다 놓음	아래 키를 눌렀다 놓습니다.	결과
a	-	a 장음 기호
e	-	e 장음 기호
e	.	e 위쪽 점
g	,	g 세디유
i	-	i 장음 기호
i	~	i 틸데
i	a	i ogonek1
k	,	k 세디유
l	,	l 세디유
n	,	n 세디유
o	-	o 장음 기호
r	,	r 세디유
t		t 스트로크
u	~	u 틸데
u	a	u ogonek
u	-	E 장음 기호
n	n	eng

다음 표는 공통 Latin-5 Compose 키 시퀀스를 나열합니다.

표 5-6 공통 Latin-5 Compose 키 시퀀스

Compose 키를 누른 후, 아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
G	u	G 약음 기호
I	.	I 위쪽 점
g	u	g 약음 기호
i	.	i dotless

다음 표는 공통 Latin-9 Compose 키 시퀀스를 나열합니다.

표 5-7 공통 라틴-9 Compose 키 시퀀스

Compose 키를 누른 후, 아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
o	e	슬러 oe
O	E	슬러 OE
Y	“	Y 분음 기호

악센트 사용 불능 키를 가진 키보드를 사용 중이면 다음 Compose 키 시퀀스를 사용하십시오. “dead_acute” 및 그러한 키 이름은 /usr/openwin/include/X11/keysymdef.h에서 볼 수 있듯이 X_dead_acute 등의 X11 등록 keysym 이름에서 옵니다. SunFA_Circum 및 그러한 키 이름은 /usr/openwin/include/X11/Sunkeysym.h에서 볼 수 있듯이 SunXK_FA_Circum와 같은 Sun 정의 X11 keysym 이름에서 옵니다.

표 5-8 악센트 사용 불능 키 기반 Compose 키 시퀀스

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
dead_grave	스페이스바	억음 악센트
dead_acute	아포스트로피	예음 악센트
dead_acute	스페이스바	아포스트로피
dead_diaeresis	이중 인용	분음 기호
dead_diaeresis	스페이스바	분음 기호
dead_circumflex	스페이스바	곡절 악센트
dead_circumflex	슬래시	수직선
dead_circumflex	0	도 기호
dead_circumflex	1	수퍼스크립트 1
dead_circumflex	2	수퍼스크립트 2
dead_circumflex	3	수퍼스크립트 3
dead_circumflex	마침표	가운데 점
dead_circumflex	감탄 포인트	분리된 수선
dead_circumflex	마이너스	장음기호
dead_circumflex	밑줄	장음기호
dead_cedilla	쉼표	세디유
dead_cedilla	마이너스	기호 없음

표 5-8 악센트 사용 불능 키 기반 Compose 키 시퀀스 (계속)

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
dead_tilde	스페이스바	tilde
dead_grave	A	A 억음 기호
dead_acute	A	A 예음 기호
dead_circumflex	A	A 곡절 기호
dead_tilde	A	A 틸데
dead_diaeresis	A	A 분음 기호
dead_grave	a	a 억음 기호
dead_acute	a	a 예음 기호
dead_circumflex	a	a 곡절 기호
dead_tilde	a	a 틸데
dead_diaeresis	a	a 분음 기호
dead_cedilla	C	C 세디유
dead_cedilla	c	c 세디유
dead_grave	E	E 억음 기호
dead_acute	E	E 예음 기호
dead_circumflex	E	E 곡절 기호
dead_diaeresis	E	E 분음 기호
dead_grave	e	e 억음 기호
dead_acute	e	e 예음 기호
dead_circumflex	e	e 곡절 기호
dead_diaeresis	e	e 분음 기호
dead_grave	I	I 억음 기호
dead_acute	I	I 예음 기호
dead_circumflex	I	I 곡절 기호
dead_diaeresis	I	I 분음 기호
dead_grave	i	i 억음 기호
dead_acute	i	i 예음 기호
dead_circumflex	i	i 곡절 기호
dead_diaeresis	i	i 분음 기호

표 5-8 악센트 사용 불능 키 기반 Compose 키 시퀀스 (계속)

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
dead_tilde	N	N 틸데
dead_tilde	n	n 틸데
dead_grave	O	O 억음 기호
dead_acute	O	O 예음 기호
dead_circumflex	O	O 곡절 기호
dead_tilde	O	O 틸데
dead_diaeresis	O	O 분음 기호
dead_grave	o	o 억음 기호
dead_acute	o	o 예음 기호
dead_circumflex	o	o 곡절 기호
dead_tilde	o	o 틸데
dead_diaeresis	o	o 분음 기호
dead_cedilla	S	S 세디유
dead_cedilla	s	s 세디유
dead_grave	U	U 억음 기호
dead_acute	U	U 예음 기호
dead_circumflex	U	U 곡절 기호
dead_diaeresis	U	U 분음 기호
dead_grave	u	u 억음 기호
dead_acute	u	u 예음 기호
dead_circumflex	u	u 곡절 기호
dead_diaeresis	u	u 분음 기호
dead_acute	Y	Y 예음 기호
dead_acute	y	y 예음 기호
dead_diaeresis	y	y 분음 기호
SunFA_Grave	스페이스바	억음 악센트
SunFA_Grave	A	A 억음 기호
SunFA_Grave	a	a 억음 기호
SunFA_Grave	E	E 억음 기호

표 5-8 악센트 사용 불능 키 기반 Compose 키 시퀀스 (계속)

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
SunFA_Grave	e	e 억음 기호
SunFA_Grave	I	I 억음 기호
SunFA_Grave	i	i 억음 기호
SunFA_Grave	O	O 억음 기호
SunFA_Grave	o	o 억음 기호
SunFA_Grave	U	U 억음 기호
SunFA_Grave	u	u 억음 기호
SunFA_Acute	아포스트로피	예음 악센트
SunFA_Acute	스페이스바	아포스트로피
SunFA_Acute	A	A 예음 기호
SunFA_Acute	a	a 예음 기호
SunFA_Acute	C	C 예음 기호
SunFA_Acute	c	c 예음 기호
SunFA_Acute	E	E 예음 기호
SunFA_Acute	e	e 예음 기호
SunFA_Acute	I	I 예음 기호
SunFA_Acute	i	i 예음 기호
SunFA_Acute	L	L 예음 기호
SunFA_Acute	l	l 예음 기호
SunFA_Acute	N	N 예음 기호
SunFA_Acute	n	n 예음 기호
SunFA_Acute	O	O 예음 기호
SunFA_Acute	o	o 예음 기호
SunFA_Acute	R	R 예음 기호
SunFA_Acute	r	r 예음 기호
SunFA_Acute	S	S 예음 기호
SunFA_Acute	s	s 예음 기호
SunFA_Acute	U	U 예음 기호
SunFA_Acute	u	u 예음 기호

표 5-8 악센트 사용 불능 키 기반 Compose 키 시퀀스 (계속)

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
SunFA_Acute	Y	Y 예음 기호
SunFA_Acute	y	y 예음 기호
SunFA_Acute	Z	Z 예음 기호
SunFA_Acute	z	z 예음 기호
SunFA_Cedilla	쉽표	세디유
SunFA_Cedilla	마이너스	기호 없음
SunFA_Cedilla	C	C 세디유
SunFA_Cedilla	c	c 세디유
SunFA_Cedilla	G	G 세디유
SunFA_Cedilla	g	g 세디유
SunFA_Cedilla	K	K 세디유
SunFA_Cedilla	k	k 세디유
SunFA_Cedilla	L	L 세디유
SunFA_Cedilla	l	l 세디유
SunFA_Cedilla	N	N 세디유
SunFA_Cedilla	n	n 세디유
SunFA_Cedilla	R	R 세디유
SunFA_Cedilla	r	r 세디유
SunFA_Cedilla	S	S 세디유
SunFA_Cedilla	s	s 세디유
SunFA_Cedilla	T	T 세디유
SunFA_Cedilla	t	t 세디유
SunFA_Circum	스페이스바	꼭절 악센트
SunFA_Circum	0	도 기호
SunFA_Circum	1	수퍼스크립트 1
SunFA_Circum	2	수퍼스크립트 2
SunFA_Circum	3	수퍼스크립트 3
SunFA_Circum	감탄 포인트	분리된 수선
SunFA_Circum	마이너스	장음기호

표 5-8 악센트 사용 불능 키 기반 Compose 키 시퀀스 (계속)

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
SunFA_Circum	밑줄	장음기호
SunFA_Circum	마침표	가운데 점
SunFA_Circum	슬래시	수직선
SunFA_Circum	A	A 곡절 기호
SunFA_Circum	a	a 곡절 기호
SunFA_Circum	C	C 곡절 기호
SunFA_Circum	c	c 곡절 기호
SunFA_Circum	E	E 곡절 기호
SunFA_Circum	e	e 곡절 기호
SunFA_Circum	G	G 곡절 기호
SunFA_Circum	g	g 곡절 기호
SunFA_Circum	H	H 곡절 기호
SunFA_Circum	h	h 곡절 기호
SunFA_Circum	I	I 곡절 기호
SunFA_Circum	i	i 곡절 기호
SunFA_Circum	J	J 곡절 기호
SunFA_Circum	j	j 곡절 기호
SunFA_Circum	O	O 곡절 기호
SunFA_Circum	o	o 곡절 기호
SunFA_Circum	S	S 곡절 기호
SunFA_Circum	s	s 곡절 기호
SunFA_Circum	U	U 곡절 기호
SunFA_Circum	u	u 곡절 기호
SunFA_Diaeresis	겹 따옴표	분음 기호
SunFA_Diaeresis	스페이스바	분음 기호
SunFA_Diaeresis	A	A 분음 기호
SunFA_Diaeresis	a	a 분음 기호
SunFA_Diaeresis	E	E 분음 기호
SunFA_Diaeresis	e	e 분음 기호

표 5-8 악센트 사용 불능 키 기반 Compose 키 시퀀스 (계속)

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
SunFA_Diaeresis	I	I 분음 기호
SunFA_Diaeresis	i	i 분음 기호
SunFA_Diaeresis	O	O 분음 기호
SunFA_Diaeresis	o	o 분음 기호
SunFA_Diaeresis	U	U 분음 기호
SunFA_Diaeresis	u	u 분음 기호
SunFA_Diaeresis	y	y 분음 기호
SunFA_Diaeresis	Y	Y 분음 기호
SunFA_Tilde	스페이스바	틸데
SunFA_Tilde	A	A 틸데
SunFA_Tilde	a	a 틸데
SunFA_Tilde	N	N 틸데
SunFA_Tilde	n	n 틸데
SunFA_Tilde	O	O 틸데
SunFA_Tilde	o	o 틸데

아랍어 입력 모드

아랍어 입력 모드로 전환하려면 **Compose a r**을 누르거나 입력 모드 선택 창에서 아랍어를 선택합니다. 입력 모드 선택 창 액세스에 대한 자세한 내용은 115 페이지 “입력 방식 액세스하기”을 참조하십시오.

다음 그림은 아랍어 키보드 레이아웃을 보여줍니다.

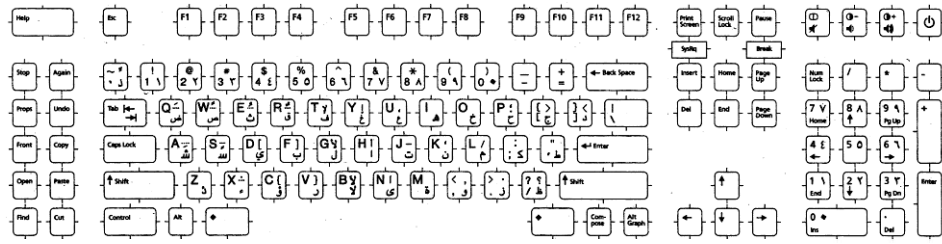


그림 5-2 아랍어 키보드

키릴 문자 입력 모드

키릴 문자 입력 모드로 전환하려면 **Compose cc**를 누르거나 입력 모드 선택 창에서 키릴 문자를 선택합니다. 입력 모드 선택 창 액세스에 대한 자세한 내용은 [115 페이지 “입력 방식 액세스하기”](#)을 참조하십시오.

키릴 문자(러시아어) 키보드 레이아웃이 다음 그림에 나타납니다.

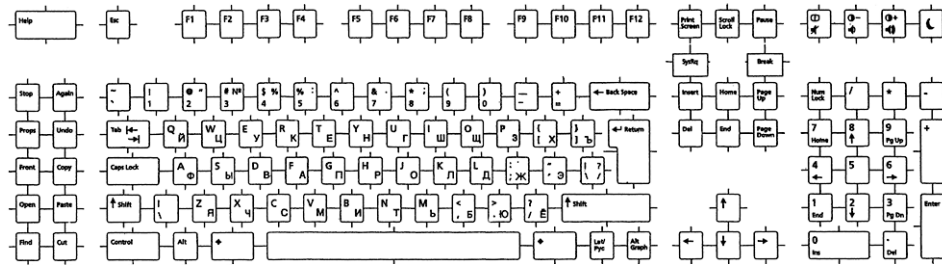


그림 5-3 키릴 문자(러시아) 키보드

키릴 문자 입력 모드로 전환한 후 영어 또는 유럽어 텍스트를 입력할 수 없습니다. 영어/유럽어 입력 모드로 다시 전환하려면 **Control**과 스페이스바를 동시에 누르거나 상태 영역을 눌러 입력 모드 선택 창에서 영어/유럽어 입력 모드를 선택합니다. [115 페이지 “입력 방식 액세스하기”](#)를 참조하십시오.

해당 입력 모드 전환 키 시퀀스를 입력하여 다른 입력 모드로 전환할 수도 있습니다.

그리스어 입력 모드

그리스어 입력 모드로 전환하려면 Compose g g를 누르거나 입력 모드 선택 창에서 아랍어를 선택합니다. 입력 모드 선택 창 액세스에 대한 자세한 내용은 115 페이지 “입력 방식 액세스하기”을 참조하십시오.

그리스어 입력 모드로 전환한 후 영어 또는 유럽어 텍스트를 입력할 수 없습니다. 영어/유럽어 입력 모드로 다시 전환하려면 Control과 스페이스바를 동시에 누르거나 상태 영역을 눌러 입력 모드 선택 창에서 영어/유럽어 입력 모드를 선택합니다. 그리스어 유럽 키보드 레이아웃이 다음 그림에 나타납니다.

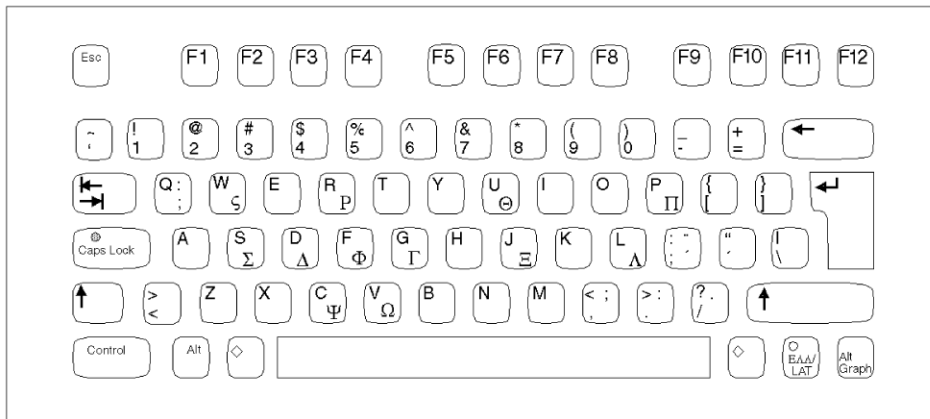


그림 5-4 그리스 유럽 키보드

다음 그림은 그리스 UNIX 키보드를 보여줍니다.

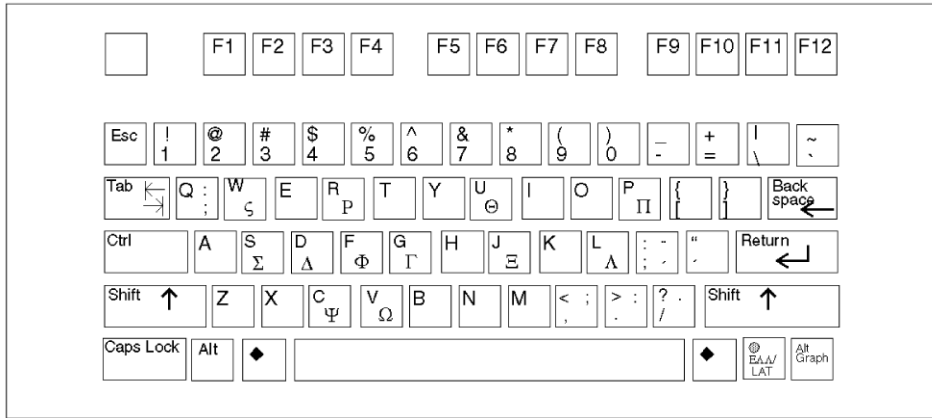


그림 5-5 그리스어 UNIX 키보드

다음 Compose 키 시퀀스가 그리스어 입력 모드에서 지원됩니다. 일부 Compose 키 시퀀스는 악센트 사용 불능 키로 시작합니다. 약자 “ordfemenine” 은 여성 서수 표시 키를 나타냅니다.

표 5-9 그리스어 입력 모드의 Compose 키 시퀀스

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
세미콜론	a	tonos를 가진 소문자 Greek_alpha
세미콜론	e	tonos를 가진 소문자 Greek_epsilon
세미콜론	h	tonos를 가진 소문자 Greek_eta
세미콜론	i	tonos를 가진 소문자 Greek_iota
세미콜론	o	tonos를 가진 소문자 Greek_omicron
세미콜론	y	tonos를 가진 소문자 Greek_upsilon
세미콜론	v	tonos를 가진 소문자 Greek_omega
세미콜론	A	tonos를 가진 대문자 Greek_alpha
세미콜론	E	tonos를 가진 대문자 Greek_epsilon

표 5-9 그리스어 입력 모드인 Compose 키 시퀀스 (계속)

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
세미콜론	H	tonos를 가진 대문자 Greek_eta
세미콜론	I	tonos를 가진 대문자 Greek_iota
세미콜론	O	tonos를 가진 대문자 Greek_omicron
세미콜론	Y	tonos를 가진 대문자 Greek_upsilon
세미콜론	V	tonos를 가진 대문자 Greek_omega
dead_acute	Greek_alpha	tonos를 가진 소문자 Greek_alpha
dead_acute	Greek_epsilon	tonos를 가진 소문자 Greek_epsilon
dead_acute	Greek_eta	tonos를 가진 소문자 Greek_eta
dead_acute	Greek_iota	tonos를 가진 소문자 Greek_iota
dead_acute	Greek_omicron	tonos를 가진 소문자 Greek_omicron
dead_acute	Greek_upsilon	tonos를 가진 소문자 Greek_upsilon
dead_acute	Greek_omega	tonos를 가진 소문자 Greek_omega
dead_acute	Greek_ALPHA	tonos를 가진 대문자 Greek_alpha
dead_acute	Greek_EPSILON	tonos를 가진 대문자 Greek_epsilon
dead_acute	Greek_ETA	tonos를 가진 대문자 Greek_eta
dead_acute	Greek_IOTA	tonos를 가진 대문자 Greek_iota
dead_acute	Greek_OMICRON	tonos를 가진 대문자 Greek_omicron
dead_acute	Greek_UPSILON	tonos를 가진 대문자 Greek_upsilon

표 5-9 그리스어 입력 모드의 Compose 키 시퀀스 (계속)

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
dead_acute	Greek_OMEGA	tonos를 가진 대문자 Greek_omega
dead_acute	a	tonos를 가진 소문자 Greek_alpha
dead_acute	e	tonos를 가진 소문자 Greek_epsilon
dead_acute	h	tonos를 가진 소문자 Greek_eta
dead_acute	i	tonos를 가진 Greek_iota
dead_acute	o	tonos를 가진 소문자 Greek_omicron
dead_acute	y	tonos를 가진 소문자 Greek_upsilon
dead_acute	v	tonos를 가진 소문자 Greek_omega
dead_acute	A	tonos를 가진 대문자 Greek_alpha
dead_acute	E	tonos를 가진 대문자 Greek_epsilon
dead_acute	H	tonos를 가진 대문자 Greek_eta
dead_acute	I	tonos를 가진 대문자 Greek_iota
dead_acute	O	tonos를 가진 대문자 Greek_omicron
dead_acute	Y	tonos를 가진 대문자 Greek_upsilon
dead_acute	V	tonos를 가진 대문자 Greek_omega
colon	i	dialylika를 가진 소문자 Greek_iota
colon	y	dialylika를 가진 소문자 Greek_upsilon
콜론	I	dialylika를 가진 대문자 Greek_iota

표 5-9 그리스어 입력 모드의 Compose 키 시퀀스 (계속)

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
colon	Y	dialytika를 가진 대문자 Greek_upsilon
dead_diaeresis	i	dialytika를 가진 소문자 Greek_iota
dead_diaeresis	y	dialytika를 가진 소문자 Greek_upsilon
dead_diaeresis	I	dialytika를 가진 대문자 Greek_iota
dead_diaeresis	Y	dialytika를 가진 대문자 Greek_upsilon
dead_diaeresis	Greek_iota	dialytika를 가진 소문자 Greek_iota
dead_diaeresis	Greek_upsilon	dialytika를 가진 소문자 Greek_upsilon
dead_diaeresis	Greek_IOTA	dialytika를 가진 대문자 Greek_iota
dead_diaeresis	Greek_UPSILON	dialytika를 가진 대문자 Greek_upsilon
세미콜론	세미콜론	그리스어 tonos
콜론	콜론	분음 기호/dialytika
ordfeminine	0	플러스-마이너스 기호
ordfeminine	1	섹션 기호
ordfeminine	2	수퍼스크립트 2
ordfeminine	3	수퍼스크립트 3
ordfeminine	5	분리된 수선
ordfeminine	6	저작권 기호
ordfeminine	7	기호 없음
ordfeminine	8	짧은 하이픈
ordfeminine	9	도 기호
ordfeminine	하이픈	구어체 분수 1/2
ordfeminine	역 슬래시	파운드 기호
ordfeminine	braceleft	수정자 글자 반전 쉼표

표 5-9 그리스어 입력 모드의 Compose 키 시퀀스 (계속)

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
ordfeminine	braceright	수정자 글자 아포스트로피
ordfeminine	bracketleft	왼쪽 이중 꺾쇠 인용 표시
ordfeminine	bracketright	오른쪽 이중 꺾쇠 인용 표시
SunFA_Acute	a	tonos를 가진 소문자 Greek_alpha
SunFA_Acute	e	tonos를 가진 소문자 Greek_epsilon
SunFA_Acute	h	tonos를 가진 소문자 Greek_eta
SunFA_Acute	i	tonos를 가진 소문자 Greek_iota
SunFA_Acute	o	tonos를 가진 소문자 Greek_omicron
SunFA_Acute	y	tonos를 가진 소문자 Greek_upsilon
SunFA_Acute	v	Greek_omega with tonos
SunFA_Acute	A	tonos를 가진 대문자 Greek_alpha
SunFA_Acute	E	tonos를 가진 대문자 Greek_epsilon
SunFA_Acute	H	tonos를 가진 대문자 Greek_eta
SunFA_Acute	O	tonos를 가진 대문자 Greek_omicron
SunFA_Acute	I	tonos를 가진 대문자 Greek_iota
SunFA_Acute	Y	tonos를 가진 대문자 Greek_upsilon
SunFA_Acute	V	tonos를 가진 대문자 Greek_omega
SunFA_Acute	Greek_alpha	tonos를 가진 소문자 Greek_alpha
SunFA_Acute	Greek_epsilon	tonos를 가진 소문자 Greek_epsilon

표 5-9 그리스어 입력 모드의 Compose 키 시퀀스 (계속)

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
SunFA_Acute	Greek_eta	tonos를 가진 소문자 Greek_eta
SunFA_Acute	Greek_iota	tonos를 가진 소문자 Greek_iota
SunFA_Acute	Greek_omega	tonos를 가진 소문자 Greek_omega
SunFA_Acute	Greek_omicron	tonos를 가진 소문자 Greek_omicron
SunFA_Acute	Greek_upsilon	tonos를 가진 소문자 Greek_upsilon
SunFA_Acute	Greek_ALPHA	tonos를 가진 대문자 Greek_alpha
SunFA_Acute	Greek_EPSILON	tonos를 가진 대문자 Greek_epsilon
SunFA_Acute	Greek_ETA	tonos를 가진 대문자 Greek_eta
SunFA_Acute	Greek_IOTA	tonos를 가진 대문자 Greek_iota
SunFA_Acute	Greek_OMICRON	tonos를 가진 대문자 Greek_omicron
SunFA_Acute	Greek_UPSILON	tonos를 가진 대문자 Greek_upsilon
SunFA_Acute	Greek_OMEGA	tonos를 가진 대문자 Greek_omega
SunFA_Diaeresis	i	dialylika를 가진 소문자 Greek_iota
SunFA_Diaeresis	y	dialytika를 가진 소문자 Greek_upsilon
SunFA_Diaeresis	I	dialylika를 가진 대문자 Greek_iota
SunFA_Diaeresis	Y	dialytika를 가진 대문자 Greek_upsilon
SunFA_Diaeresis	Greek_iota	dialylika를 가진 소문자 Greek_iota
SunFA_Diaeresis	Greek_upsilon	dialytika를 가진 소문자 Greek_upsilon

표 5-9 그리스어 입력 모드의 Compose 키 시퀀스 (계속)

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
SunFA_Diaeresis	Greek_IOTA	dialytika를 가진 대문자 Greek_iota
SunFA_Diaeresis	Greek_UPSILON	dialytika를 가진 대문자 Greek_upsilon

표 5-10 3개 키를 통한 그리스어 Compose 키 시퀀스 입력 모드

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
세미콜론	콜론	y	dialytika와 tonos를 가 진 소문자 Greek_upsilon
콜론	세미콜론	y	dialytika와 tonos를 가 진 소문자 Greek_upsilon
세미콜론	콜론	i	dialytika와 tonos를 가 진 소문자 Greek_iota
콜론	세미콜론	i	dialytika와 tonos를 가 진 소문자 Greek_iota
dead_acute	dead_diaeresis	y	dialytika와 tonos를 가 진 소문자 Greek_upsilon
dead_diaeresis	dead_acute	y	dialytika와 tonos를 가 진 소문자 Greek_upsilon
dead_acute	dead_diaeresis	i	dialytika와 tonos를 가 진 소문자 Greek_iota
dead_diaeresis	dead_acute	i	dialytika와 tonos를 가 진 소문자 Greek_iota
dead_acute	dead_diaeresis	Greek_upsilon	dialytika와 tonos를 가 진 소문자 Greek_upsilon
dead_diaeresis	dead_acute	Greek_upsilon	dialytika와 tonos를 가 진 소문자 Greek_upsilon
dead_acute	dead_diaeresis	Greek_iota	dialytika와 tonos를 가 진 소문자 Greek_iota
dead_diaeresis	dead_acute	Greek_iota	dialytika와 tonos를 가 진 소문자 Greek_iota

표 5-10 3개 키를 통한 그리스어 Compose 키 시퀀스 입력 모드 (계속)

아래 키를 눌렀다 놓습니다. 아래 키를 눌렀다 놓습니다. 아래 키를 눌렀다 놓습니다.				결과
SunFA_Acute	SunFA_Diaeresis	i		dialytika와 tonos를 가진 소문자 Greek_iota
SunFA_Diaeresis	SunFA_Acute	i		dialytika와 tonos를 가진 소문자 Greek_iota
SunFA_Acute	SunFA_Diaeresis	y		dialytika와 tonos를 가진 소문자 Greek_upsilon
SunFA_Diaeresis	SunFA_Acute	y		dialytika와 tonos를 가진 소문자 Greek_upsilon
SunFA_Acute	SunFA_Diaeresis	Greek_iota		dialytika와 tonos를 가진 소문자 Greek_iota
SunFA_Diaeresis	SunFA_Acute	Greek_iota		dialytika와 tonos를 가진 소문자 Greek_iota
SunFA_Acute	SunFA_Diaeresis	Greek_upsilon		dialytika와 tonos를 가진 소문자 Greek_upsilon
SunFA_Diaeresis	SunFA_Acute	Greek_upsilon		dialytika와 tonos를 가진 소문자 Greek_upsilon

표 5-11 4개 키를 통한 그리스어 Compose 키 시퀀스 입력 모드

아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	아래 키를 눌렀다 놓습니다.	결과
세미콜론	콜론	세미콜론	콜론	그리스어 dialytika tonos
콜론	세미콜론	콜론	세미콜론	그리스어 dialytika tonos

히브리어 입력 모드

히브리어 입력 모드로 전환하려면 Compose h h를 누르거나 입력 모드 선택 창에서 히브리어를 선택합니다. 입력 모드 선택 창 액세스에 대한 자세한 내용은 115 페이지 “입력 방식 액세스하기”을 참조하십시오.

다음 그림은 히브리어 키보드 레이아웃을 보여줍니다.



그림 5-6 히브리어 키보드

일본어 입력 모드

일본어 입력 모드로 전환하려면 Compose ja를 누르거나 입력 모드 선택 창에서 일본어를 선택합니다. 입력 모드 선택 창 액세스에 대한 자세한 내용은 115 페이지 “입력 방식 액세스하기”을 참조하십시오.

일본어 입력 시스템을 사용하려면 하나 이상의 일본어 로케일을 설치하고 시스템을 재부팅해야 합니다. 일본어 로케일을 설치한 후 모든 UTF-8 로케일에서 ATOK12를 사용할 수 있습니다. Wnn6은 ja_JP.UTF-8을 제외한 UTF-8 로케일에서 사용할 수 없습니다.

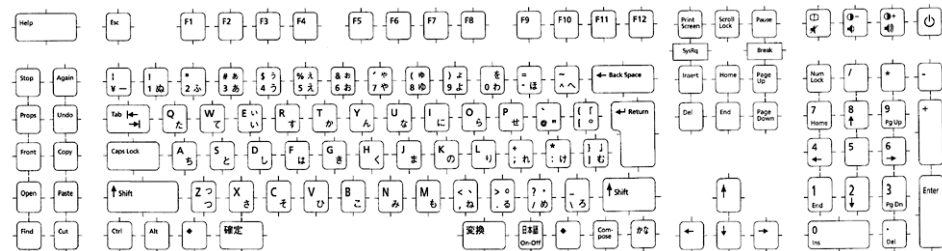


그림 5-7 일본어 키보드

한국어 입력 모드

한국어 입력 모드로 전환하려면 Compose k o를 누르거나 입력 모드 선택 창에서 한국어를 선택합니다. 입력 모드 선택 창 액세스에 대한 자세한 내용은 115 페이지 “입력 방식 액세스하기”을 참조하십시오.

한국어 입력 시스템에 사용하려면 시스템에 하나 이상의 한국어 로케를 설치해야 합니다. 한국어 입력 시스템 사용 방법에 대한 자세한 내용은 한국어 Solaris 사용 설명서를 참조하십시오.

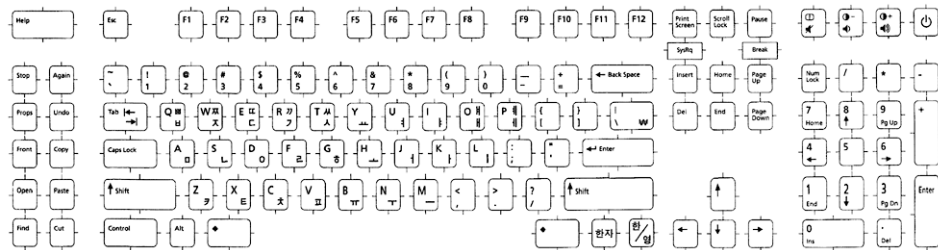


그림 5-8 한국어 키보드

간체 한자 입력 모드

간체 한자 입력 모드로 전환하려면 Compose s c를 누르거나 입력 모드 선택 창에서 S-한자를 누르십시오. 115 페이지 “입력 방식 액세스하기”.

간체 한자 입력 시스템을 사용하려면 시스템에 하나 이상의 간체 한자 로케를 설치해야 합니다. 간체 한자 입력 시스템 사용 방법에 대한 자세한 내용은 *Simplified Chinese Solaris User's Guide*를 참조하십시오.

정체 한자 입력 모드

정체 한자 입력 모드로 전환하려면 Compose t c를 누르거나 입력 모드 선택 창에서 T-중국어를 선택합니다. 입력 모드 선택 창 액세스에 대한 자세한 내용은 115 페이지 “입력 방식 액세스하기”.

원 정체 한자 입력 시스템에 액세스하려면 시스템에 하나 이상의 정체 한자 로케를 설치해야 합니다. 정체 한자 입력 시스템 사용 방법에 대한 자세한 내용은 *Traditional Chinese Solaris User's Guide*를 참조하십시오.

정체 한자(홍콩) 입력 모드

정체 한자 입력 모드로 전환하려면 **Compose h c**를 누르거나 입력 모드 선택 창에서 정체 한자(홍콩)를 선택합니다. 입력 모드 선택 창 액세스에 대한 자세한 내용은 115 페이지 “입력 방식 액세스하기”을 참조하십시오.

원 정체 한자(홍콩) 입력 시스템에 액세스하려면 시스템에 하나 이상의 정체 한자(홍콩) 로케를 설치해야 합니다.

유니코드 16진법 입력 모드

유니코드 16진법 입력 모드로 전환하려면 **Compose u h**를 누르거나 입력 모드 선택 창에서 유니코드 16진법을 선택합니다. 8진법 숫자 시스템으로 전환하려면 **Compose u o**를 누르거나 유니코드 8진법을 선택합니다. 입력 모드 선택 창 액세스에 대한 자세한 내용은 115 페이지 “입력 방식 액세스하기”을 참조하십시오.

입력 모드를 사용하려면, 문자의 16진법 또는 8진법 코드 포인트 값을 알아야 합니다. 코드 포인트와 문자 사이의 매핑에 대한 자세한 내용은 *The 유니코드 Standard, Version 4.0*을 참조하십시오.

유니코드 16진법 코드 입력 모드에서 문자를 입력하려면 4 개의 16진법 숫자를 입력합니다. 일부 예제의 16진 값은 다음과 같습니다.

- 반전 느낌표는 00A1
- 그리스어 작은 문자 베타는 03B2
- 한국어 한글 음절은 AC00
- 일본어 가타카나 문자 A는 30A1
- 통합 한 문자는 4E58

16진법 숫자에 대하여 A, B, C, D, E, F의 대소문자 모두를 사용할 수 있습니다. 16진법 숫자 대신 8진법 숫자 시스템을 선호할 경우 8진법 숫자 0~7을 입력할 수 있습니다. Delete 또는 Backspace 키를 사용하여 숫자를 입력할 수 있습니다.

테이블 조회 입력 모드

테이블 조회 모드로 전환하려면 Compose 11를 누르거나 입력 모드 선택 창에서 조회를 선택합니다. 입력 모드 선택 창 액세스에 대한 자세한 내용은 115 페이지 “입력 방식 액세스하기”을 참조하십시오.

두번째 조회 창은 그룹 전용 표시에 대한 후보를 표시하여 한 번에 최대 80개 후보를 표시합니다. 다음 후보 세트에 대해서는 Control n을 누르고 이전 후보 세트에 대해서는 Control p를 누릅니다.

시스템 환경

이 절은 로케일 환경 변수, TTY 환경 설치, 32비트 및 64비트 STREAMS 모듈, 단말기 지원을 설명합니다.

로케일 환경 변수

시스템에 en_US.UTF-8 로케일이 설치되어 있는지 확인하십시오. 다양한 범주의 현재 로케일 설정을 확인하려면 locale 유틸리티를 사용합니다.

```
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=
```

en_US.UTF-8 로케일 데스크탑 환경을 사용하려면 먼저 로케일을 선택하십시오. TTY 환경에서는 먼저 다음 C-셸 예와 같이 LANG 환경 변수를 en_US.UTF-8로 설정하여 로케일을 선택합니다.

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

LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, LC_NUMERIC, LC_MONETARY, 및 LC_TIME 범주가 설정되지 않았거나 en_US.UTF-8로 설정되었는지 확인합니다. 범주가 설정되면 보다 낮은 우선 순위의 LANG 환경 변수를 무시합니다. 환경 변수 계층에 대한 자세한 내용은 setlocale(3C) 매뉴얼 페이지를 참조하십시오.

CDE 데스크탑에서 en_US.UTF-8 환경을 시작할 수도 있습니다. CDE 로그인 화면의 옵션 -> 언어 메뉴에서, en_US.UTF-8를 선택하십시오.

TTY 환경 설정

사용하는 단말기 및 단말기 에뮬레이터에 따라 특정한 코드 세트 특정 STREAMS 모듈을 스트림으로 푸시해야 할 수 있습니다.

STREAMS 모듈 및 일반적인 스트림에 대한 자세한 내용은 *STREAMS Programming Guide*를 참조하십시오.

다음 표는 단말기 환경의 en_US.UTF-8 로케일이 지원하는 64비트 STREAMS 모듈을 나열합니다. 자세한 내용은 *Solaris 64-bit Developer's Guide*를 참조하십시오.

표 5-12 en_US.UTF-8이 지원하는 STREAMS 모듈

32비트 STREAMS 모듈	설명
/usr/kernel/strmod/sparcv9/u8lat1	UTF-8과 ISO8859-1(서부 유럽어) 사이의 코드 변환 STREAMS 모듈
/usr/kernel/strmod/sparcv9/u8lat2	UTF-8과 ISO8859-2(동부 유럽어) 사이의 코드 변환 STREAMS 모듈
/usr/kernel/strmod/sparcv9/u8koi8	UTF-8과 KOI8-R(키릴 문자) 사이의 코드 변환 STREAMS 모듈

주 - Solaris 10 릴리스부터 32비트 커널은 SPARC sun4u 플랫폼에서 더 이상 지원되지 않습니다. 표 5-12는 x86 플랫폼에 대한 32비트 커널에만 적용됩니다. 자세한 내용은 릴리스 노트를 참조하십시오.

다음 표는 en_US.UTF-8이 지원하는 64비트 STREAMS 모듈을 나열합니다.

표 5-13 en_US.UTF-8이 지원하는 64비트 STREAMS 모듈

64비트 STREAMS 모듈	설명
/usr/kernel/strmod/sparcv9/u8lat1	UTF-8과 ISO8859-1(서부 유럽어) 사이의 코드 변환 STREAMS 모듈
/usr/kernel/strmod/sparcv9/u8lat2	UTF-8과 ISO8859-2(동부 유럽어) 사이의 코드 변환 STREAMS 모듈
/usr/kernel/strmod/sparcv9/u8koi8	UTF-8과 KOI8-R(키릴 문자) 사이의 코드 변환 STREAMS 모듈

▼ STREAMS 커널 모듈을 로드하는 방법

1. 루트 사용자로 64비트 Solaris 시스템을 실행하는지 32비트 시스템을 실행하는지 결정합니다.

```
system# isainfo -v
```

- 64비트 Solaris 시스템은 다음 정보를 반환합니다.

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

- 32비트 Solaris 시스템은 다음 정보를 반환합니다.

```
32-bit sparc applications
```

- 32비트 x86 시스템은 다음 정보를 반환합니다.

```
32-bit i386 applications
```

2. 시스템이 이미 STREAMS 모듈에 있는지 결정합니다..

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

u8lat1과 같은 STREAMS 모듈이 이미 설치되었으면 출력은 다음과 같습니다.

```
system# modinfo | grep u8lat1
```

```
89 ff798000 4b13 18 1 u8lat1 (UTF-8 <--> ISO 8859-1 module)
```

3. 모듈이 아직 로드되지 않았으면 modload(1M) 명령을 사용하여 로드하십시오.

- 32비트 시스템에서 다음을 입력합니다.

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

- 64비트 시스템에서 다음을 입력합니다.

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

적절한 u8lat1 STREAMS 모듈이 커널에 로드됩니다. 이제 스트림으로 푸시할 수 있습니다.

▼ STREAMS 커널 모듈을 로드 해제하는 방법

1. 루트로 커널 모듈이 로드되었는지 확인합니다.

예를 들어, u8lat1이 로드되었는지 확인하려면 다음을 입력합니다.

```
system# modinfo | grep u8lat1
```

```
89 ff798000 4b13 18 1 u8lat1 (UTF-8 <--> ISO 8859-1 module)
```

2. 커널을 로드 해제하려면 modunload(1M) 명령을 사용합니다.

u8lat1 모듈을 로드 해제하려면 다음을 입력합니다.

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

▼ 라틴-2 단말기 및 STREAMS 모듈을 설정하는 방법

1. 두번째 명령줄에서 보이듯이 strchg(1M)를 사용합니다.

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

```

u8lat1
ptem
^D
system% strchg -f /tmp/mystreams
strchg(1)를 사용할 때 장치의 루트 또는 소유자인지 확인하십시오.

```

2. 현재 구성을 조사하려면 **stconf** 명령을 실행합니다.

```

system% stconf
ttcompat
ldterm
u8lat1
ptem
pts
system%

```

3. 원래 구성을 재설정하려면 **strchg** 명령을 실행합니다.

```

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

```

dtterm, xterm 및 UTF-8 문자의 입출력이 가능한 단말기

Solaris Operating System 이전 버전과 달리 UTF-8 코드 세트의 입출력을 지원하는 dtterm 및 xterm 단말기 에뮬레이터 및 기타 단말기는 스트림에 추가 STREAMS 모듈이 있을 필요가 없습니다. ldterm 모듈은 이제 코드 독립적이고 stty(1) 유틸리티로 단말기 환경을 설정하는 경우 유니코드/UTF-8을 지원합니다.

유니코드 로케일의 적합한 단말기 환경을 설정하려면 stty(1) 유틸리티를 사용하십시오.

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

현재 설정을 질의하려면 다음과 같이 stty 유틸리티의 -a 옵션을 사용합니다.

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

주 - /usr/ucb/stty는 국제화되지 않았기 때문에 대신 /bin/stty를 사용하십시오.

라틴-1, 라틴-2 또는 KOI8-R에 대한 단말기 지원

라틴-1 (ISO8859-1), 라틴-2 (ISO8859-2) 또는 KOI8-R만 지원하는 단말기의 경우 다음 STREAMS 구성이 있어야 합니다.

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

이 구성은 라틴-1을 지원하는 단말기에만 해당됩니다. 라틴-2 단말기의 경우 STREAMS 모듈 u8lat1을 u8lat2로 교체합니다. KOI8-R 단말기에 대하여 모듈을 u8koi8로 바꾸십시오.

커널로 이미 STREAMS 모듈을 로드했는지 확인하십시오.

~/ .cshrc에 설정 저장하기

필요한 STREAMS 모듈이 커널과 함께 로드되었다고 가정하면 편의를 위해 .cshrc 파일(C 셸 예제)에 다음 행을 저장할 수 있습니다.

```
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
```

이러한 행이 .cshrc 파일에 있으면 STREAMS 모듈을 사용할 때마다 모든 명령을 입력할 필요가 없습니다. 두번째 _EOF가 파일의 첫번째 열에서 시작해야 함을 주의하십시오.

코드 변환

유니코드 로캘 지원은 iconv 및 sdtconvtool 유틸리티를 통해 많은 국가의 주요한 코드 세트 사이의 다양한 코드 변환을 추가합니다.

현재 Solaris 환경에서 유틸리티 geniconvtbl을 사용하여 사용자 정의 코드 변환이 가능합니다. geniconvtbl 유틸리티를 사용하여 작성된 사용자 정의 코드 변환은 iconv(1) 및 iconv(3) 모두와 함께 사용될 수 있습니다. 이 유틸리티에 대한 자세한 내용은 geniconvtbl(1) 및 geniconvtbl(4) 설명서 페이지를 참조하십시오.

iconv, iconv_open 및 sdtconvtool에 적용할 수 있는 사용 가능한 fromcode 및 tocode 이름이 [부록 A](#)의 표에서 표시됩니다. iconv 코드 변환에 대한 자세한 내용은 iconv(1) 및 sdtconvtool(1) 설명서 페이지를 참조하십시오. 사용 가능한 코드 변환에 대한 자세한 내용은 iconv(5) 설명서 페이지를 참조하십시오. [부록 A](#)도 참조하십시오.

주 - UCS-2, UCS-4, UTF-16 및 UTF-32는 모두 문자가 문자 스트림의 처음에 나타날 경우 유니코드 4.0 및 ISO/IEC 10646-1:2000 표준에 정의된 Byte Order Mark (BOM)을 인식하는 유니코드/ISO/IEC 10646 표현 형식입니다. UCS-2BE, UCS-4BE, UTF-16BE, UTF-32BE와 같은 다른 형식은 모두 BOM 문자를 인식하지 않는 고정 너비 유니코드/ISO/IEC 10646 표현 형식이고 또한 빅 엔디언 바이트 순서를 가정합니다. 반면 UCS-2LE, UCS-4LE, UTF-16LE, UTF-32LE 같은 표현 형식은 리틀 엔디언 바이트 정렬을 가정합니다. 이러한 형식은 또한 BOM 문자를 인식하지 않습니다.

ISO8859-* 및 KOI8-*의 관련 스크립트 및 언어에 대해서는 <http://czyborra.com/charsets/iso8869.html>을 참조하십시오.

DtMail 지원

스크립트의 적용 범위 증가의 결과 en_US.UTF-8 로케일에서 실행되는 Solaris DtMail은 MIME 이름이 표시하는 다음 문자 세트를 지원합니다.

- US-ASCII (7비트 US ASCII)
- UTF-8 (UCS 전송 형식 8비트)
- UTF-7 (UCS 전송 형식 7비트)
- ISO-8859-1 (라틴-1)
- ISO-8859-2 (라틴-2)
- ISO-8859-3 (라틴-3)
- ISO-8859-4 (라틴-4)
- ISO-8859-5 (라틴/키릴 문자)
- ISO-8859-6 (라틴/아랍어)
- ISO-8859-7 (라틴/그리스어)
- ISO-8859-8 (라틴/히브리어)
- ISO-8859-9 (라틴-5)
- ISO-8859-10 (라틴-6)
- ISO-8859-13 (라틴-7/발트어)
- ISO-8859-14 (라틴-8/켈트어)
- ISO-8859-15 (라틴-9)
- ISO-8859-16 (라틴-10)
- KOI8-R (키릴 문자)
- ISO-2022-JP 및 EUC-JP (일본어)
- ISO-2022-KR 및 EUC-KR (한국어)
- ISO-2022-CN (간체 한자)
- ISO-8859-13 (라틴-7/발트어)
- ISO-8859-14 (라틴-8/켈트어)
- KOI8-U (키릴 문자/우크라이나어)
- Shift_JIS (Shift JIS의 일본어)
- GB2312 (EUC의 간체 한자)

- TIS-620 (태국어)
- UTF-16 (UCS 전송 형식 16비트)
- UTF-16BE (UTF-16 빅 엔디언)
- UTF-16LE (UTF-16 리틀 엔디언)
- Windows-1250
- Windows-1251
- Windows-1252
- Windows-1253
- Windows-1254
- Windows-1255
- Windows-1256
- Windows-1257
- Windows-1258
- Big5 (정체 한자)
- UTF-16 (UCS 전송 형식 32비트)
- UTF-32BE (UTF-32 빅 엔디언)
- UTF-32LE (UTF-32 리틀 엔디언)

이 지원은 사용자가 DtMail의 단일 인스턴스로 세계 모든 지역의 다양한 문자 세트로 인코딩된 모든 종류의 전자 우편을 볼 수 있도록 지원합니다. DtMail은 전자 우편과 함께 제공되는 MIME 문자 세트와 내용 전송 인코딩을 찾아 수신된 전자 우편을 암호 해독합니다. Windows-125x MIME 문자 세트가 지원됩니다.

전자 우편을 보내는 경우 수신자 우편 사용자 에이전트(우편 클라이언트)가 이해하는 MIME 문자 세트를 지정해야 하고 그렇지 않으면 en_US.UTF-8 로케일이 제공하는 기본 MIME 문자 세트를 사용할 수 있습니다. 새 메시지 창에서 **Control Y**를 누르거나 형식 메뉴 버튼을 누른 다음 문자 세트 변경 버튼을 눌러 보내는 전자 우편의 문자 세트를 전환할 수 있습니다. 다음 사용 가능 문자 세트 이름이 보내기 버튼의 맨 위 왼쪽 모서리에 표시됩니다.

전자 우편 메시지 헤더나 메시지 본문이 지정된 MIME 문자 세트가 표현할 수 없는 문자를 포함하면 시스템이 자동으로 모든 문자를 표현할 수 있는 UTF-8로 문자 세트를 전환합니다.

메시지가 7비트 US-ASCII 문자 세트의 문자만 포함하면 전자 우편의 기본 MIME 문자 세트는 US-ASCII입니다. 모든 우편 사용자 에이전트는 문자 또는 정보의 상실 없이 그러한 전자 우편 메시지를 해석할 수 있습니다.

메시지가 스크립트 혼합의 문자를 포함하면 기본 MIME 문자 세트는 UTF-8입니다. UTF-8의 모든 8비트 문자는 할당된 인쇄 가능 인코딩으로 인코딩됩니다. MIME, 등록된 MIME 문자 세트, 할당된 인쇄 가능 인코딩에 대한 자세한 내용은 RFCs 2045, 2046, 2047, 2048, 2049, 2279, 2152, 2237, 1922, 1557, 1555 및 1489를 참조하십시오.

0L20L2L02

L02

0L2

L020L20L20L2

L020L20L20L2



모내기 다른 이름으로 저장 모내기 다른 이름으로 저장 모내기 다른 이름으로 저장

하그 서버서
간트 - 48-1

프로그래밍 환경

국제화 응용 프로그램은 en_US.UTF-8 로케일을 자동으로 활성화해야 합니다. 그러나 응용프로그램 자원 파일의 적합한 FontSet/XmFontList 정의가 필수입니다.

국제화 응용 프로그램에 대한 자세한 내용은 *Creating Worldwide Software: Solaris International Developer's Guide*, 제2판을 참조하십시오.

X 응용프로그램과 함께 사용되는 FontSet

X 응용 프로그램과 함께 사용되는 FontSet에 대한 자세한 내용은 112 페이지 “유니코드 로케일: en_US.UTF-8 지원”을 참조하십시오.

각 문자 세트에는 Solaris 데스크탑 환경의 연관된 글꼴 세트가 있습니다.

다음은 현재 Solaris 환경에서 지원되는 라틴-1 글꼴입니다.

```
-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
```

-dt-interface user-* 및 -dt-application-* 별명을 포함하는 CDE 공통 글꼴 별명에 대한 자세한 설명은 *Common Desktop Environment: Internationalization Programmer's Guide*를 참조하십시오.

en_US.UTF-8 로케일에서 utf도 X 논리적 글꼴 설명 이름의 스타일 필드의 추가 속성으로 로케일의 공통 글꼴 별명에 포함됩니다. 따라서 적합한 글꼴 세트를 가지려면 다음 예제에서와 같이 추가 스타일이 글꼴 세트 생성에 포함되어야 합니다.

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

CDE/Motif 응용프로그램의 글꼴 목록 정의

FontSet 정의에서와 같이 응용프로그램의 XmFontList 자원 정의는 로케일이 지원하는 추가 스타일 속성도 포함해야 합니다.

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


Complex Text Layout

Complex Text Layout(CTL) 확장으로 Motif APIs가 논리적이고 실제적인 텍스트 표현 간의 복잡한 변형을 필요로 하는 쓰기 시스템을 지원할 수 있습니다. 아랍어, 히브리어, 태국어에는 그러한 변형이 필요합니다. CTL Motif는 합자, 분음 기호, 세그먼트 순서와 같은 문자 형성을 제공합니다. 정적 및 동적 텍스트 위젯의 변형에 대한 지원도 동적 텍스트 위젯에 대한 양방향 텍스트 기능과 동적 탭 표시와 함께 제공됩니다. 텍스트 렌더링은 표현 레이어를 통해 처리되기 때문에 다른 위젯 라이브러리는 CTL을 지원하도록 쉽게 확장될 수 있습니다.

이 장은 다음 내용으로 구성되어 있습니다.

- 157 페이지 “CTL 기술의 개요”
- 158 페이지 “CTL 구조의 개요”
- 158 페이지 “X 라이브러리 기반 응용프로그램에 대한 CTL 지원”
- 159 페이지 “XOC 자원”
- 159 페이지 “CTL 기술 지원을 위한 Motif 변경 사항”
- 168 페이지 “CTL 응용프로그램 개발하기”

CTL 기술의 개요

새 기능을 활용하려면 사용자는 Portable Layout Services (PLS) 라이브러리와 적절한 언어 엔진을 가져야 합니다. CTL은 PLS를 언어 엔진에 대한 인터페이스로 사용하고 언어 엔진을 사용하여 텍스트가 렌더링되기 전에 텍스트를 변형합니다. CTL을 지원하는 응용프로그램은 CTL 설명서의 설명과 같이 추가 자원을 포함해야 합니다.

특정적으로, XomCTL은 기본적인 로캘 종속적 PSL 모듈 변형에 의해 제공되는 다음과 같은 복잡한 언어 형성과 재정렬 기능을 지원합니다.

- 위치 변화
- 결합(다대일)과 문자 합성(일대다)
- 구별 기호

- 양방향성
- 대칭적 교체
- 숫자 형성
- 문자열 검증

CTL 구조의 개요

CTL 구조는 그림 6-1과 같이 구성됩니다. 스택 상부에 있는 Dt Apps는 텍스트 렌더링을 위해 Motif CTL 기능을 사용합니다. Motif는 PLS를 사용하여 로켈 특정 언어 엔진과 인터페이스를 이루고 위치 변화, 숫자 모양 등을 지원하는 변환을 수행합니다.

CTL 구조는 로켈 특정 엔진을 가진 새로운 언어를 지원합니다. 다시 말해서, 타이어와 베트남어에 대한 지원은 Motif 또는 Dt Apps의 변경 없이 추가될 수 있습니다.

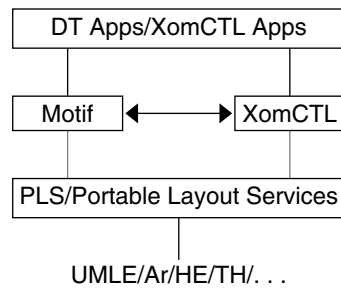


그림 6-1 CTL 구조

X 라이브러리 기반 응용프로그램에 대한 CTL 지원

XomCTL(X 라이브러리 출력 모듈의 Complex Text Layout 지원)은 모든 순수 X 윈도우 응용프로그램(X 기반 단말기 에뮬레이터)이 CTL 지원을 갖도록 합니다. XomCTL은 X11 단순 글꼴 지원을 포함하는 완전 기능 개방형 소스 XI18N 구현을 제공합니다.

XOC 자원

다음 XOC 자원은 현재 Solaris 환경에서 지원됩니다.

XNText	사용자가 CTL 작업이 수행되어야 하는 텍스트 버퍼를 설정할 수 있도록 합니다.
XNTextLayoutNumGlyphs	텍스트 버퍼에 있는 텍스트의 글리프 수를 알려줍니다.
XNTextLayoutModifier	Motif의 XmNLayoutModifier와 동일합니다.
XNTextLayoutProperty	PLS 특성, 입력 대 출력 및 출력 대 입력과 동일합니다.
XNTextLayoutMapInpToOut	PLS 특성, 입력 대 출력 및 출력 대 입력과 동일합니다.
XNTextLayoutMapOutToInp	PLS 특성, 입력 대 출력 및 출력 대 입력과 동일합니다.

이 자원에 대한 설명은 X/Open 또는 PLS 이식 가능 서비스의 사양에서 얻을 수 있습니다.

CTL 기술 지원을 위한 Motif 변경 사항

Motif의 다음 변경 사항은 CTL 기술을 지원합니다:

XmNlayoutDirection	객체 레이아웃을 제어합니다.
XmStringDirection	시스템이 문자열의 문자들을 표현하는 방향을 지정합니다.
XmRendition	XmRendition에 가상 자원을 추가합니다.
XmText 및 XmTextField	XmRendition과 연관된 텍스트의 레이아웃 동작에 영향을 미칩니다.
XmTextFieldGetLayoutModifier	rendition 레이아웃 객체의 레이아웃 수정자 문자열을 반환합니다.
XmTextGetLayoutModifier	위젯트와 관련된 rendition의 현재 레이아웃 객체 설정 값을 반환합니다.
XmTextFieldSetLayoutModifier	rendition에 결합된 레이아웃 객체용 레이아웃 수정자 값을 설정합니다.

XmTextSetLayoutModifier	위젯트와 관련된 rendition의 레이아웃 객체 설정을 수정합니다.
XmStringDirectionCreate	복합 문자열을 작성합니다.

XmNlayoutDirection 자원

XmNlayoutDirection 자원은 객체 레이아웃을 제어합니다. 이 자원은 아래와 같은 방식으로 LayoutObject의 방향 값과 상호 작용합니다.

XmNlayoutDirection 개요, 특히 XmStringDirection 및 XmNlayoutDirection 간의 상호 작용에 대한 설명은 *Motif Programmer's Guide (Release 2.1)*의 11.3절을 참조하십시오.

레이아웃 방향 결정하기

XmNlayoutDirection이 XmDEFAULT_DIRECTION로 지정되면 위젯트의 레이아웃 방향은 만들어질 때 제어하는 의사 XOC에서 설정됩니다. 동적 텍스트(XmText 및 XmTextField)의 경우 의사 XOC는 위젯트에 대하여 사용되는 XmRendition와 연관된 것입니다. 정적 텍스트(XmList, XmLabel, XmLabelG)의 경우 레이아웃 방향은 방향을 지정하는 첫번째 복합 문자열 구성 요소에서 설정됩니다. 지정은 두 가지 중 한 가지 방법으로 수행됩니다:

- 구성 요소는 유형 XmSTRING_COMPONENT_LAYOUT_PUSH 또는 XmSTRING_COMPONENT_DIRECTION입니다.
- 구성 요소는 연관된 XmRendition 및 LayoutObject의 유형 XmSTRING_COMPONENT_LOCALE_TEXT, XmSTRING_COMPONENT_WIDECHAR_TEXT, 또는 XmSTRING_COMPONENT_TEXT입니다.

XmNlayoutDirection이 XmDEFAULT_DIRECTION으로 지정되지 않고 XmNlayoutModifier @ls orientation 값이 레이아웃 수정자 문자열에서 명시적으로 지정되지 않으면 XmNlayoutDirection 값은 XOC 및 해당하는 LayoutObject에 전달됩니다.

XmNlayoutDirection 및 XmNlayoutModifier @ls orientation 값이 명시적으로 지정되어 있으면 동작이 혼합됩니다. XmNlayoutDirection은 위젯트 객체 레이아웃을 제어하며 XmNlayoutModifier @ls orientation 값은 레이아웃 변환을 제어합니다.

기존 POSIX 로케일 모델에 대한 논리적 확장으로 컨텍스트 종속적 및 양방향 텍스트 변형을 처리하기 위한 이식 가능 함수의 설명은 *CAE Specification: Portable Layout Services: Context-dependent and Directional Text*(The Open Group: Feb 1997; ISBN 1-85912-142-X; 문서 번호 C616)를 참조하십시오. 이 설명서는 복합 텍스트 언어에 대한 지원을 제공하고자 하는 시스템 및 응용프로그램 프로그래머들을 위한 것입니다.

XmStringDirection 자원

XmStringDirection은 시스템이 문자열의 문자를 표시하는 방향을 지정하는 데 사용되는 유형입니다.

XmNlayoutDirection 자원은 해당 문자열의 방향을 지정하는 구성 요소가 없는 모든 복합 문자열(XmString)에 대하여 기본 rendering 방향을 설정합니다. 따라서 레이아웃 방향을 설정하려면 XmNlayoutDirection 자원을 위한 적절한 값을 설정해야 합니다. 특정 방향 구성 요소를 사용해 복합 문자열을 작성할 필요는 없습니다. 응용프로그램이 XmString을 렌더링하면 해당 응용프로그램은 방향이 명시된 문자열이 작성되었는지 확인합니다(XmStringDirection). 문자열에서 아무런 방향 구성 요소도 제공하지 않으면 응용프로그램은 현재 위젯트를 위한 XmNlayoutDirection 자원의 값을 확인하고 해당 값을 XmString을 위한 기본 렌더링 방향으로 사용합니다.

XmRendition 자원

CTL은 새로운 다음 표에 나열된 의사 자원을 XmRendition에 추가합니다. 의사 자원의 설명은 표를 참조합니다.

표 6-1 XmRendition의 새로운 자원

이름	클래스/유형	액세스	기본값
XmNfontType	XmCFontType/XmFontType	CSG	XmAS_IS
XmNlayoutAttrObject	XmClayoutAttrObject/String	CG	NULL
XmNlayoutModifier	XmClayoutModifier/String	CSG	NULL

XmNfontType

Rendition 글꼴 객체의 유형을 지원합니다. CTL의 경우 이 자원의 값은 XmFONT_IS_XOC값이어야만 합니다. 값이 일치하지 않으면 XmNlayoutAttrObject 및 XmNlayoutModifier 자원이 무시됩니다.

이 자원의 값이 XmFont_IS_XOC이고 XmNfont 자원이 지정되지 않으면 작성될 때 XmNfontName 자원의 값이 XmNlayoutAttrObject 자원이 지정하는 로케일 또는 현재 로케일에서 XOC 객체로 변환됩니다. 또, XmNlayoutModifier 자원의 값은 XOC와 연관된 모든 레이아웃 객체에 전달됩니다.

XmNlayoutAttrObject

레이아웃 AttrObject 인자를 지정합니다. 자원은 XmRendition과 관련되어 있는 XOC와 관련된 레이아웃 객체를 작성하는데 사용됩니다. 이 문자열의 구문 및 의미에 대해서는 레이아웃 서비스 m_create_layout() 사양을 참조하십시오. 레이아웃 수정자 방향 출력값과 XmNlayoutDirection 위젯트 자원 간의 상호 작용에 대한 설명은 XmNfontType 설명을 참조하십시오.

XmNlayoutModifier

XmRendition용으로 XOC와 함께 사용된 레이아웃 객체에 전달될 레이아웃 값을 지정합니다. 문자열의 구문과 의미에 대한 내용은 *CAE Specification*을 참조하십시오.

XmRendition{Retrieve, Update}을 사용하여 이 자원을 설정하면 문자열은 이 rendition과 연관된 XOC와 연관된 레이아웃 객체에 전달됩니다. 이 메커니즘을 통해 레이아웃 서비스를 동적으로 구성할 수 있습니다. Orientation, Context, TypeOfText, TextShaping 또는 ShapeCharset이 변경되면 예상치 못한 동작이 나타날 수 있습니다.

추가 레이아웃 동작

XmNlayoutModifier는 XmRendition과 연관된 텍스트의 레이아웃 동작에 영향을 미칩니다. 예를 들어, 숫자의 레이아웃 기본 처리가 NUMERALS_NOMINAL인 경우 XmNlayoutModifier를 @ls numerals=nominal:national 또는 @ls numerals=:national로 설정하여 NUMERALS_NATIONAL로 변경합니다.

레이아웃 값은 다음 그룹으로 구분할 수 있습니다:

- 코드화 설명 - TypeOfText, TextShaping, ShapeCharset (및 로캘 코드 세트)
TypeOfText는 기본적으로 정렬을 세그먼트화하고 불투명한 블록으로 그려질 수 있습니다. 이 값을 rendition 객체를 통해 동적으로 수정하는 것은 대개 불필요한 일이며 예측하지 못한 동작을 유발할 수 있습니다.
- 레이아웃 동작 - Orientation, Context, ImplicitAlg, Swapping 및 Numerals. Orientation 및 Context는 동적으로 수정해선 안 됩니다. ImplicitAlg, Swapping 및 Numerals는 안전하게 수정할 수 있습니다.
- 편집 동작 - CheckMode

XmText and XmTextField 자원

Xm CTL은 Motif 2.0 CStext 위젯트 다음 패턴화되어 시각적으로 작동하는 이동 및 삭제 활동의 병렬 목록을 추가하여 XmText 및 XmTextField를 확장합니다. 표준 Motif 2.1 Text 및 TextField는 논리적 순서와 물리적 순서를 구분하지 않습니다. next 및 forward는 "오른쪽 방향"을 의미하고 previous 및 backward는 "왼쪽 방향"을 의미합니다. 그러나 CStext는 적절한 구분을 하고 엄격하게 물리적인 이름(예를 들어, left-character(), delete-right-word() 등)로 새로운 동작 집합을 정의합니다. 이 활동 루틴 모두는 위젯트의 XmNlayoutDirection에 민감하게 정의되며 적절한 다음 또는 이전 활동을 호출합니다.

Xm CTL 확장은 CStext 확장보다 다소 복잡합니다. Xm CTL 확장은 중립적 고정화를 포함한 의사 XOC에 의한 결정에 따라 위젯트의 전체 방향보다 커서를 둘러싼 물리적 문자의 구체적 방향에 민감합니다.

새로운 자원 이름을 사용하여 rendition 태그를 제공하고 배열을 제어하기 위한 선택 정책을 제어할 수 있습니다.

새로운 Xm CTL 활동 세트는 대략 {Move, Delete, Kill}, {Left, Right}, {Character, Word}의 조합으로 이루어지며 아래에 나열되어 있습니다. 활동 목록은 다음 표에 나와 있습니다.

표 6-2 Xm CTL의 새로운 자원

이름	클래스/유형	액세스	기본값
XmNrenditionTag	XmCRenditionTag/XmRString	CSG	XmFONTLIST_DEFAULT_TAG
XmNalignment	XmCAlignment/XmRAlignment	CSG	XmALIGNMENT_BEGINNING
XmNeditPolicy	XmCEditPolicy/XmREditPolicy	CSG	XmEDIT_LOGICAL

XmNrendition 태그

위젯트에 대하여 사용되는, XmNrenderTable 자원에 있는 XmRendition의 rendition 태그를 지정합니다.

XmNalignment

위젯트에서 사용되는 텍스트 배열을 지정합니다. XmALIGNMENT_END 및 XmALIGNMENT_CENTER만 지원됩니다.

XmNeditPolicy

위젯트, XmEDIT_LOGICAL 또는 XmEDIT_VISUAL에 대하여 사용되는 편집 정책을 지정합니다. XmEDIT_VISUAL의 경우에는는 선택, 커서 이동 및 삭제가 시각적 스타일로 표시됩니다. 자원을 설정하면 표준 키보드 이동 및 삭제 이벤트에 대한 번역도 새 "시각적" 활동 목록이나 기존 논리적 활동으로 변경됩니다.

문자 방향 활동 루틴

forward-cell() 및 backward-cell() 활동은 지정된 방향의 문자 방향을 질의합니다. 방향이 왼쪽에서 오른쪽 방향이면 해당하는 next-/forward- 또는 previous-/backward- 변수를 호출합니다.

문자 방향 추가 동작

이 동작은 중첩 수준에 대하여 레이아웃 서비스 변환 OutToInp 및 Property 버퍼를 사용하여 문자의 방향을 결정합니다. 따라서 위젯트의 동작은 로캘 지향 변환에 따라 달라집니다. OutToInp의 정보 또는 특히, Property 버퍼가 정확하지 않으면 위젯트는 예기치 않은 동작이 일어날 수 있습니다. 아울러, 로캘 지향 모듈이 본 사양의 범위를 벗어나면 양방향 편집 동작은 동일한 텍스트, 응용프로그램, 자원 값 및 LayoutObject 구성의 경우에도 플랫폼에 따라 차이가 있을 수 있습니다.

시각적 모드 활동은 셀 기반 동작의 표시입니다. 논리적 모드 활동은 논리적 문자 기반 동작을 유발할 수 있습니다. 예를 들어, delete-right-character() 작업은 디스플레이 셀에 해당하는 입력 버퍼 문자들을 삭제합니다. 즉, LayoutObject 변환 "속성" 바이트 "새 셀 표시 기호"가 1인 한 개의 입력 버퍼 문자 및 "새 셀 표시 기호"가 0인 뒤따르는 모든 문자입니다.

Property 버퍼에 대한 자세한 정보는 CAE Specification의 m_transform_layout() 사양을 참조하십시오.

마찬가지로 `backward-character()`의 경우 삽입점은 입력 버퍼에서 한 문자 뒤로 이동되며 커서는 관련된 출력 버퍼 문자에 해당하는 시각적 위치로 복귀합니다. 따라서 복합적인 디스플레이 셀을 가로질러 이동하려면 여러 번 키 입력을 해야 함을 의미합니다; 커서는 삽입 지점이 “새로운 셀 표시 기호”가 0인 구별 기호 또는 합자 조각과 같은 입력 버퍼 문자에 걸쳐 이동함에 따라 실제로 표시 위치를 변경하지 않습니다.

이것은 삭제 작업이 논리/입력 버퍼 측이나 물리적/출력측의 디스플레이 셀 수준에서 일어난다는 것을 의미합니다. 입력 및 출력 버퍼 간의 1 대 1 대응이 성립되지 않기 때문에 엄격한 물리적 문자 대 문자 삭제를 위한 모드는 없습니다. 주어진 물리적 문자는 예를 들어 논리적 문자의 한 단편만을 나타낼 수 있습니다.

XmText 활동 루틴

다음 목록은 XmText 활동 루틴을 설명합니다.

`left-character(extend)`

XmNeditPolicy가 XmEDIT_LOGICAL이고 인자가 없이 호출되면 삽입 커서가 논리적으로 한 문자 뒤로 이동합니다. 삽입 커서가 줄의 시작 부분에 있으면 삽입 커서는 이전 줄의 논리적인 마지막 문자로 이동합니다. 그렇지 않으면 삽입 커서 위치는 변경되지 않습니다.

XmNeditPolicy가 XmEDIT_VISUAL이면 커서는 커서 위치의 왼쪽으로 이동합니다. 삽입 커서가 줄의 시작 부분에 있으면 삽입 커서는 이전 줄의 끝 문자로 이동합니다.

`left-character()`가 `extend` 인자와 함께 호출되면 삽입 커서는 인자가 없는 경우처럼 이동하고 현재 선택을 확장합니다.

`left-character()` 활동은 합리적인 값 `XmCR_MOVING_INSERT_CURSOR`을 가진 `XmNmotionVerifyCallback` 프로시저에 대한 호출을 생성합니다. `extend` 인자를 통해 호출되면 `XmNgainPrimaryCallback` 프로시저에 대한 호출을 생성할 수 있습니다. 자세한 정보는 *Motif Programmer's Reference*의 콜백 설명을 참조하십시오.

`right-character(extend)`

XmNeditPolicy가 XmEDIT_LOGICAL이고 아무런 인자가 없이 호출되면 삽입 커서가 논리적으로 한 문자 앞으로 이동합니다. 삽입 커서가 줄의 논리적인 끝 부분에 있으면 삽입 커서는 다음 줄의 논리적인 시작 부분으로 이동합니다.

XmNeditPolicy가 XmEDIT_VISUAL이면 커서는 커서 위치의 오른쪽으로 이동합니다. 삽입 커서가 줄의 끝 부분에 있으면 삽입 커서는 다음 줄의 시작 부분으로 이동합니다.

`extend`의 인자를 통해 호출되면 XmNeditPolicy는 인자가 없는 경우와 마찬가지로 삽입 커서를 이동시키고 현재의 선택을 확장합니다.

`right-character()` 활동은 합리적인 값 `XmCR_MOVING_INSERT_CURSOR`를 가진 `XmNmotionVerifyCallback` 프로시저에 대한 호출을 생성합니다. `extend` 인자를 통해 호출되면 이 활동은 `XmNgainPrimaryCallback` 프로시저에 대한 호출을 생성할 수 있습니다. 자세한 정보는 *Motif Programmer's Reference*의 콜백 설명을 참조하십시오.

right-word(extend)

XmNeditPolicy가 XmEDIT_LOGICAL이고 인자 없이 호출되면 삽입 커서가 논리적인 연속된 문자(있을 경우)의 논리적 시작 문자로 이동합니다. 그렇지 않으면 기존 단어의 논리적 끝 부분으로 이동시킵니다. 삽입 커서가 줄의 논리적인 끝 부분이나 줄의 논리적인 마지막 단어에 있으면 해당 커서를 다음 줄의 논리적인 첫 번째 단어로 이동시킵니다. 그렇지 않으면 기존 단어의 논리적 끝 부분으로 이동시킵니다.

XmNeditPolicy가 XmEDIT_VISUAL이고 인자가 없이 호출되면 삽입 커서를 줄 끝의 오른쪽 또는 뒤에 있는 첫 번째 흰색 공백 문자 뒤 비 흰색 공백 문자로 이동시킵니다.

extend의 인자를 통해 호출되면 인자가 없는 경우와 마찬가지로 삽입 커서를 이동시키고 현재의 선택을 확장합니다.

left-word() 활동은 합리적인 값 XmCR_MOVING_INSERT_CURSOR을 가진 XmNmotionVerifyCallback 프로시저에 대한 호출을 생성합니다. extend 인자를 통해 호출되면 이 활동은 XmNgainPrimaryCallback 프로시저에 대한 호출을 생성할 수 있습니다. 자세한 정보는 *Motif Programmer's Reference*의 콜백 설명을 참조하십시오.

delete-left-character()

XmNeditPolicy가 XmEDIT_LOGICAL이면 delete-previous-char()에 상응합니다. XmNeditPolicy가 XmEDIT_VISUAL이면 정상 모드에서 널이 아닌 선택이 있는 경우 해당 선택을 삭제합니다. 그렇지 않으면 삽입 커서의 남은 문자를 삭제합니다. 추가 모드에서 널이 아닌 선택이 있으면 커서는 선택에서 사라지지 않고 XmNpendingDelete가 참으로 설정되며 해당 선택을 삭제합니다. 그렇지 않으면 선택에 영향을 미칠 수 있는 삽입 커서의 왼쪽 문자를 삭제합니다.

delete-left-character() 활동은 합리적인 값 XmCR_MODIFYING_TEXT_VALUE을 가진 XmNmodifyVerifyCallback 프로시저 및 합리적인 값 XmCR_VALUE_CHANGED을 가진 XmNvalueChangedCallback 프로시저에 대한 호출을 생성합니다.

delete-right-character()

XmNeditPolicy가 XmEDIT_VISUAL이면 delete-next-character()에 상응합니다. XmNeditPolicy가 XmEDIT_LOGICAL이면 정상 모드에서 널이 아닌 선택이 있는 경우 해당 선택을 삭제합니다. 그렇지 않으면 삽입 커서의 오른쪽 문자를 삭제합니다. 추가 모드에서 널이 아닌 선택이 있으면 커서는 선택에서 사라지지 않고 XmNpendingDelete가 참으로 설정되며 해당 선택을 삭제합니다. 그렇지 않으면 삽입 커서의 오른쪽 문자를 삭제합니다. 이 활동은 선택에 영향을 미칠 수 있습니다.

delete-left-character() 활동은 합리적인 값 XmCR_MODIFYING_TEXT_VALUE을 가진 XmNmodifyVerify-Callback 프로시저 및 합리적인 값 XmCR_VALUE_CHANGED을 가진 XmNvalue-ChangedCallback 프로시저에 대한 호출을 생성합니다.

문자 구성, 묶음 기호 및 분음 부호를 지원하기 위해 일부 셀 기반 루틴이 구축됩니다. 다시 말해, 두 개 이상의 문자가 단일 표시 셀을 점유하는 도형 문자로 표현됩니다.

XmText 셀 활동 루틴은 다음 목록에서 설명합니다.

backward-cell (extend)

삽입 커서를 한 셀 뒤로 이동합니다. XmNeditPolicy 가 XmEDIT_LOGICAL이면 삽입 커서는 현재의 셀보다 논리적으로 앞선 셀의 시작 부분으로 이동합니다. 그렇지 않으면 현재 셀의 시작 부분으로 이동합니다.

XmNeditPolicy가 XmEDIT_VISUAL이면 커서는 커서 왼쪽 셀의 시작 부분으로 이동합니다. prev-cell() 활동은 합리적인 값 XmCR_MOVING_INSERT_CURSOR을 가진 XmNmotionVerifyCallback 프로시저에 대한 호출을 생성합니다. extend 인자를 통해 호출되면 XmNgainPrimaryCallback 프로시저에 대한 호출을 생성할 수 있습니다. 자세한 정보는 *Motif Programmer's Reference*의 콜백 설명을 참조하십시오.

forward-cell (extend)

삽입 커서를 논리적인 다음 셀의 시작 부분으로 이동시킵니다. 그렇지 않으면 커서를 셀의 끝 부분으로 이동시킵니다. XmNeditPolicy가 XmEDIT_LOGICAL이면 커서는 한 셀 앞으로 이동합니다.

XmNeditPolicy가 XmEDIT_VISUAL이면 커서는 커서 위치 오른쪽의 셀의 시작 위치로 이동하고(있을 경우) 그렇지 않으면 현재 셀의 끝 부분으로 이동합니다.

forward-cell() 활동은 합리적인 값 XmCR_MOVING_INSERT_CURSOR을 가진 XmNmotionVerifyCallback 프로시저에 대한 호출을 생성합니다. extend 인자를 통해 호출되면 XmNgainPrimaryCallback 프로시저에 대한 호출을 생성할 수 있습니다. 자세한 정보는 *Motif Programmer's Reference*의 콜백 설명을 참조하십시오.

XmTextFieldGetLayoutModifier 자원

XmTextFieldGetLayoutModifier()는 해당 rendition에 연결된 레이아웃 객체의 상태를 반영하는 레이아웃 수정자 문자열을 반환합니다.

XmTextFieldGetLayoutModifier() 용 구문:

```
#include <Xm/TextF.h>
string XmTextFieldGetLayoutModifier(Widget widget)
```

XmTextFieldGetLayoutModifier()는 위젯트에 연관된 rendition의 현재 레이아웃 객체 설정의 값에 액세스합니다. 편리한 함수를 이용해 레이아웃 객체 수정자 값이 변경되면 XmTextFieldGetLayoutModifier 함수가 변경된 값 뿐 아니라 레이아웃 객체의 전체 상태를 반환합니다.

XmTextFieldGetLayoutModifier()는 문자열 값 형태로 레이아웃 객체 수정자 값을 반환합니다.

XmTextGetLayoutModifier 자원

XmTextGetLayoutModifier()는 해당 rendition에 연결된 레이아웃 객체의 상태를 반영하는 레이아웃 수정자 문자열을 반환합니다.

XmTextGetLayoutModifier() 용 구문:

```
#include <Xm/Text.h>
String XmTextGetLayoutModifier (Widget widget)
```

XmTextGetLayoutModifier는 위젯과 연관된 rendition의 현재 레이아웃 객체 설정의 값에 액세스합니다. 편리한 함수를 사용해 레이아웃 객체 수정자 값을 변경하면 XmTextGetLayoutModifier 함수는 변경된 값 뿐 아니라 전체 레이아웃 객체의 상태를 반환합니다.

XmTextGetLayoutModifier는 문자열 값의 형태로 레이아웃 객체 수정자 값을 반환합니다.

XmTextFieldSetLayoutModifier 자원

XmTextFieldSetLayoutModifier()는 해당 rendition에 연결된 레이아웃 객체의 동작을 변경하는 레이아웃 수정자 값을 설정합니다.

XmTextFieldSetLayoutModifier()용 구문:

```
#include <Xm/TextF.h> \
void XmTextFieldSetLayoutModifier (Widget \
widgetstring layout_modifier)
```

XmTextFieldSetLayoutModifier는 위젯과 관련된 rendition의 레이아웃 객체 설정을 수정합니다. 편리한 함수를 사용해 레이아웃 객체 수정자 값이 설정되면 입력 매개 변수에 지정된 속성들만이 변경됩니다; 그 밖의 속성은 바뀌지 않습니다.

XmTextSetLayoutModifier 자원

XmTextSetLayoutModifier()은 해당 rendition의 레이아웃 객체의 동작을 변경하는 레이아웃 수정자 값을 설정합니다.

XmTextSetLayoutModifier()용 구문:

```
#include <Xm/Text.h>
void XmTextSetLayoutModifier (Widget widget, string layout_modifier)
```

XmTextSetLayoutModifier는 위젯과 관련된 rendition의 레이아웃 객체 설정을 수정합니다. 레이아웃 객체 수정자 값이 이 편리한 기능을 사용하여 설정되면 입력 매개 변수에 지정된 속성만 변경되고 나머지 속성은 변하지 않습니다.

XmStringDirectionCreate 자원

XmStringDirectionCreate는 복합 문자열을 작성합니다.

XmTextSetLayoutModifier()용 구문:

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

XmStringDirectionCreate는 단일 구성 요소를 가진 복합 문자열을 주어진 값의 방향으로 작성합니다. 한편, XmNlayoutDirection 자원은 문자열을 위한 방향을 지정하는 구성 요소를 갖고 있지 않은 임의의 복합 문자열(XmString)을 위한 기본 렌더링 방향을 설정합니다. 따라서 레이아웃 방향을 설정하려면 XmNlayoutDirection 자원을 위한 적절한 값을 설정해야 합니다. 특정 방향 구성 요소를 갖는 복합 문자열을 만들 필요는 없습니다.

응용프로그램이 XmString을 렌더링하면 해당 응용프로그램은 방향이 명시된 문자열이 작성되었는지 확인합니다(XmStringDirection). 문자열에서 아무런 방향 구성 요소도 제공하지 않으면 응용프로그램은 현재 위젯트를 위한 XmNlayoutDirection 자원의 값을 확인하고 해당 값을 XmString을 위한 기본 렌더링 방향으로 사용합니다.

UIL 인자

다음 표는 UIL 인자 이름 및 유형을 보여줍니다.

표 6-3 UIL

UIL 인자 이름	인자 유형
XmNlayoutAttrObject	String
XmNlayoutModifier	String
XmNrenditionTag	String
XmNalignment	Integer
XmNeditPolicy	Integer

CTL 응용프로그램 개발하기

다음 절에서는 CTL 응용프로그램 개발법을 설명합니다.

레이아웃 방향 제어

복합 문자열의 방향은 영어, 스페인어, 프랑스어 및 독일어와 같이 왼쪽에서 오른쪽으로 쓰는 언어나 히브리어 및 아랍어와 같은 오른쪽에서 왼쪽으로 쓰는 언어의 텍스트 모두에 데이터 구조가 유용하도록 저장됩니다. Motif 응용프로그램에서는 VendorShell 또는

MenuShell에서 XmNlayoutDirection 자원을 사용해 레이아웃 방향을 설정할 수 있습니다. 관리자 및 프리미티브 위젯(Gadgets 포함) 역시 XmNlayoutDirection 자원을 갖고 있습니다. 기본 값은 동일한 자원을 갖고 있는 가장 가까운 상위로부터 상속됩니다.

XmText 위젯의 경우 가로 방향은 물론 세로 방향을 지정해야 합니다. layoutDirection을 XmRIGHT_TO_LEFT로 설정하면 문자열 방향은 오른쪽에서 왼쪽으로 설정되지만 커서는 세로 방향으로 움직입니다. 세로 방향이 중요하고 위에서 아래 방향 할당이 필요할 경우 XmRIGHT_TO_LEFT TOP_TO_BOTTOM을 지정하도록 하십시오. 이 설정은 구성 요소가 먼저 오른쪽에서 왼쪽으로 배치된 다음 위에서 아래로 배치되도록 지정하여 원하는 동작을 얻습니다.

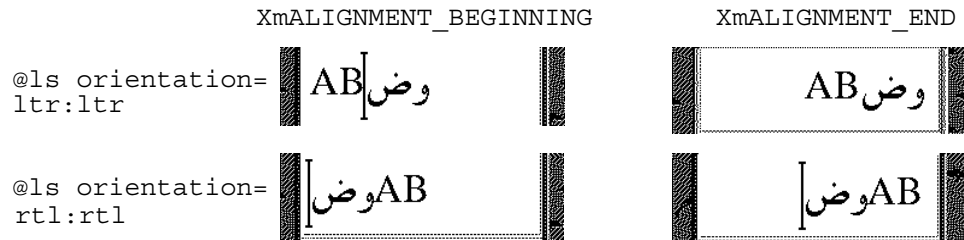
XmText 및 TextField 위젯의 동작은 XmRendition의 XmNalignment 및 XmNlayoutModifier 자원의 영향도 받습니다. 이 자원들은 XmNlayoutDirection과 함께 텍스트 위젯의 레이아웃 동작을 제어합니다. 이 동작은 그림 6-2에 표시되어 있습니다.

그림에서 사용된 입력 문자열:

A B و ض

다음 그림을 위한 XmNlayoutModifier 문자열 @ls orientation= 설정 값은 왼쪽 열에 표시되어 있습니다.

Layout Direction: XmLEFT_TO_RIGHT



Layout Direction: XmRIGHT_TO_LEFT

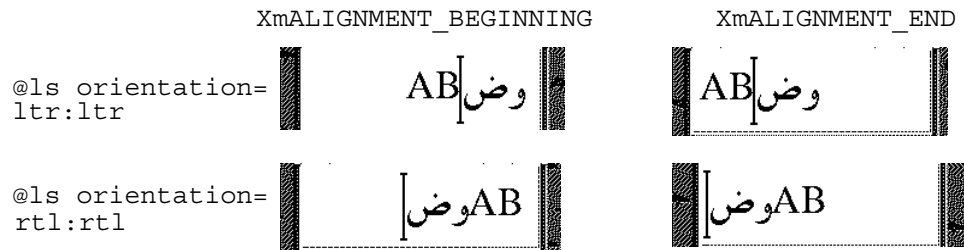


그림 6-2 레이아웃 방향

그림에서 보이는 바와 같이, XmNalignment는 레이아웃 방향에 따라 텍스트 오른쪽을 비울 것인지, 아니면 왼쪽을 비울 것인지를 지정합니다. XmNlayoutModifier는 텍스트를 세그먼트로 나눈 뒤 방향 값에 따라 왼쪽에서 오른쪽 또는 오른쪽에서 왼쪽으로 배열합니다. 다시 말해 XmNlayoutDirection이 XmRIGHT_TO_LEFT 이고 XmNalignment 값이 XmALIGNMENT_BEGINNING 이면 문자열 오른쪽이 비워집니다.

예 6-1 Rendition 만들기

다음 코드는 태그가 "ArabicShaped"인 렌더링을 사용하여 XmNlabelString이 유형 XmCHARSET_TEXT인 XmLabel을 만듭니다. Rendition은 "ar"(아랍어 로케일의 로케일 이름)의 XmNlayoutAttrObject와 출력 버퍼에 대하여 "iso8859-6"의 NUMERALS_CONTEXTUAL 및 ShapeCharset 값의 Numerals을 지정하는 레이아웃 수정자 문자열로 만듭니다.

로케일 특정 레이아웃 모듈은 16비트 유니코드 코드 세트를 사용하여 입력 텍스트를 코드화된 물리적 문자의 출력 버퍼로 변환합니다. 명시적 레이아웃 로케일은 지정되어 있기 때문에 이 텍스트는 런타임 로케일 설정에 관계 없이 올바르게 렌더링됩니다. 이 예에서는 입력이 ISO 8859-6으로 코드화됩니다.

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

예 6-1 Rendition 만들기 (계속)

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

예 6-2 Rendition 편집하기

다음 코드는 단일 Rendition을 사용하여 TextField 위젯트와 RenderTable을 만듭니다. XmNlayoutAttrObject 및 XmNlayoutModifier 의사 자원은 지정되지 않은 채로 남았으므로 NULL로 설정됩니다. 이 값은 Rendition 명령과 관련된 레이아웃 객체가 존재하는 경우 기본 로케일에 속한다는 것을 의미합니다.

이 예제가 올바르게 작동하려면 로케일 코드 세트가 ISO 8859-6이고 로케일 지향 레이아웃 모듈이 IMPLICIT_BASIC 알고리즘을 지원할 수 있는 로케일로 설정되어야 합니다. Rendition의 LayoutObject의 ImplicitAlg 값은 Rendition의 XmNlayoutModifier 의사 자원에 의해 수정됩니다.

```
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);
```

예 6-2 Rendition 편집하기 (계속)

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

자원 파일에서 렌더 테이블 만들기

응용프로그램의 국제화가 올바르게 이루어지려면 **Rendition**과 렌더 테이블을 지정해야 합니다. 렌더 테이블이 파일에서 지정되면 프로그램 바이너리는 주어진 로컬의 요건에 독립적으로 만들어지므로 로컬 요건에 맞추어 쉽게 사용자 정의할 수 있습니다.

렌더 테이블은 다음 구문에 의해 리소스 파일에서 지정됩니다: *resource_spec*: [*tag* [, *tag* *]

이 때 *tag*는 *rendition*의 *XmNtag* 자원에 적합한 일부 문자열입니다.

이 줄은 지정된 바와 같이 한 개 이상의 *rendition*을 포함한 최초의 렌더 테이블을 만듭니다. *rendition*은 지정된 태그에 부착됩니다:

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

다음 예제에서는 리소스 파일을 사용해 설정할 수 있는 *XmRendition*과 관련된 CTL 자원들을 보여줍니다. 레이아웃 객체가 효과를 발휘하려면 *fontType*을 *FONT_IS_XOC*로 설정해야 합니다. *@ls*를 사용해 지정된 *layoutModifier*는 *rendition* 객체에 의해 레이아웃 객체에 전달됩니다.

*layoutModifier*을 사용하여 레이아웃 객체에서 설정될 수 있는 자원의 완전한 목록은 *CAE Specification: Portable Layout Services: Context-dependent and Directional Text*, The Open Group: Feb 1997; ISBN 1-85912-142-X; 문서 번호 C616을 참조하십시오.

예 6-3 응용프로그램에서 렌더 테이블 만들기

렌더 테이블을 만들기 전에 먼저 응용프로그램이 테이블의 일부인 *rendition* 중 최소 한 개를 작성해야 합니다. *XmRenderTableAddRenditions()* 함수는 이름이 암시하듯이 새 *rendition*으로 렌더 테이블을 증대시키는 데에도 사용됩니다. 새 렌더 테이블을 만들려면 기존 렌더 테이블 대신 *NULL* 인자로 *XmRenderTableAddRenditions()* 기능을 호출하십시오.

다음 코드는 *XmFONT_IS_XOC*에 설정된 *XmNfontType*으로 만든 *rendition*을 사용해 렌더 테이블을 만듭니다.

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

예 6-3 응용프로그램에서 렌더 테이블 만들기 (계속)

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

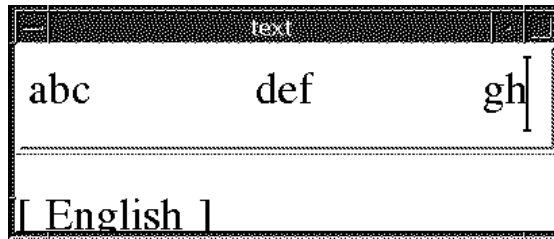
수평 탭

복합 문자열에는 텍스트의 위치를 제어하는 탭 문자가 들어있을 수 있습니다. 표시에서 이들 문자를 해석하기 위해 위젯트는 해당 복합 문자열에 효과를 주는 `rendition`에 대하여 탭 정지 목록을 참조합니다. 하지만 동적 위젯트인 `TextField` 및 `XmText`는 `rendition`의 탭 자원을 사용하지 않습니다. 대신 위젯트는 $8 * (\text{width of character } 0)$ 공식을 사용해 탭 너비를 계산합니다.

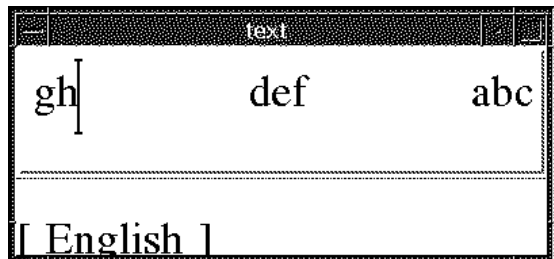
탭 측정치는 복합 문자열 표시의 왼쪽 여백으로부터의 거리입니다. 레이아웃 방향이 오른쪽에서 왼쪽으로인 경우 거리는 오른쪽 여백으로부터 측정됩니다. 텍스트의 방향에 관계 없이(아랍어 오른쪽에서 왼쪽, 또는 영어 왼쪽에서 오른쪽), 탭은 레이아웃 방향(`XmNlayoutDirection`)에 의해 지정된 대로 오른쪽이나 왼쪽에 공백을 삽입합니다.

탭 뒤의 텍스트는 항상 탭 스톱에 맞추어집니다. 탭 스톱은 위젯트의 시작 위치부터 계산되며 `XmNlayoutDirection`의 영향을 받습니다. 탭의 동작과 텍스트 방향성 및 위젯트의 `XmNlayoutDirection` 과의 상호 작용은 다음 그림에 나타나 있습니다.

이 그림을 위한 입력은 `abc\tdef\tgh`입니다.



Layout Direction: XmLEFT_TO_RIGHT



Layout Direction: XmRIGHT_TO_LEFT

그림 6-3 탭 이동 동작

마우스 선택

사용자는 마우스 버튼 1로 기본 선택을 합니다. 이 버튼을 누르면 기존 선택이 선택 해제되고 삽입 커서 및 앵커를 버튼을 누른 텍스트 위치로 이동합니다. 마우스 버튼 1을 누른 채 끝면 앵커와 포인터 위치 사이의 모든 텍스트가 선택되고 해당 범위 밖의 텍스트는 선택 해제됩니다.

선택된 텍스트는 자원의 영향을 받으며 XmNeditPolicy 자원은 XmEDIT_LOGICAL 또는 XmEDIT_VISUAL로 설정할 수 있습니다. 만일 XmNeditPolicy가 XmEDIT_LOGICAL로 설정되어 있고 선택된 텍스트가 양방향이면 선택된 텍스트는 연속적이지 않은 세그먼트의 모음이 됩니다. 논리 버퍼의 텍스트가 디스플레이와 1 대 1 대응이 이루어지지 않기 때문입니다.

결과적으로 양방향 텍스트의 논리적 문자의 연속적 버퍼는 렌더링될 때 연속적인 문자 스트림으로 만들어지지 않습니다. 거꾸로, XmNeditPolicy가 XmEDIT_VISUAL로 설정되면 선택된 텍스트는 시각적으로 연속적일 수 있지만 논리적 버퍼에서 세그먼트화되지 않습니다. 따라서 동일한 커서 포인트에서의 양방향 텍스트의 선택, 삭제 및 삽입 순서는 항상 동일한 결과를 가져오지는 않습니다.

키보드 선택

선택 작업은 마우스와 키보드로 수행할 수 있습니다. Shift와 화살표 키를 조합하여 텍스트를 선택할 수 있습니다.

선택된 텍스트는 자원의 영향을 받으며 `XmNeditPolicy` 자원은 `XmEDIT_LOGICAL` 또는 `XmEDIT_VISUAL`로 설정할 수 있습니다. `XmNeditPolicy`가 `XmEDIT_LOGICAL`로 설정되고 선택된 텍스트가 양방향이면 선택된 텍스트는 시각적으로 연속적인 것으로 보이지 않습니다. 논리 버퍼의 텍스트가 디스플레이와 1 대 1 대응이 이루어지지 않기 때문에 양방향 텍스트의 논리적 문자의 연속적 버퍼는 렌더링될 때 연속적인 문자 스트림을 만들어내지 않습니다.

거꾸로, `XmNeditPolicy`가 `XmEDIT_VISUAL`로 설정되면 선택된 텍스트는 시각적으로 연속적일 수 있지만 논리적 버퍼에서 세그먼트화되지 않습니다. 따라서 동일한 커서 포인트에서의 양방향 텍스트의 선택, 삭제 및 삽입 순서는 항상 동일한 결과를 가져오지는 않습니다.

텍스트 자원 및 기하학

다음 텍스트 자원은 기하학과 관련이 있습니다.

- 렌더 테이블 `XmNrenderTable`은 글꼴이나 글꼴 집합 및 텍스트를 표시하기 위한 기타 속성을 선택하기 위해 위젯트가 사용합니다.

`Text` 및 `Textfield` 위젯트는 `XmNfontType` 과 같이 글꼴과 관련된 표현 자원만을 사용합니다. 이러한 위젯트는 `XmNlayoutAttrObject`와 같은 레이아웃 객체의 속성도 지정할 수 있습니다. 이러한 위젯트는 일반적으로 로컬 식별자와 이 `XmRendition`과 관련된 레이아웃 객체를 통과하도록 레이아웃 값을 지정하는 `XmNlayoutModifier`를 포함합니다.

- 텍스트가 위젯트보다 넓을 경우 단어 경계에서 줄을 바꿀지를 지정하는 자원 (`XmNwordWrap`).

단어 경계에서 행을 바꾸어도 텍스트에 새 행이 삽입되지 않습니다. 아랍어와 같은 흘림체 언어의 경우에 단어 길이가 위젯트 길이보다 크면 단어는 다음 행으로 줄바꿈됩니다. 그러나 다음 행의 첫번째 문자는 논리적 버퍼에서 이전 문자에 독립적으로 형성됩니다.

이식 관련 지침

Complex Text Layout (CTL)용으로 활성화된 새 Motif 라이브러리는 /usr/dt/lib/libXm.so.4 에 있습니다. 응용프로그램이 libXm.so.3에 연결되면 CTL을 지원하지 않습니다. `ldd app_name`은 응용프로그램이 연결할 라이브러리를 보여줍니다. CTL을 활성화하기 위해 기존 응용프로그램을 이식하려면 다음 단계를 따라야 합니다.

1. `-DSUN_CTL`을 Makefile 에 추가하십시오.
이 플래그는 중요한 것으로 CTL을 지원하기 위해 필요한 데이터 구조를 포함하고 있습니다. 이 값은 컴파일 과정에서 설정해야 합니다.
2. 기존 응용프로그램을 다시 컴파일하십시오.
컴파일을 다시 하면 자동으로 CTL을 사용할 수 있는 Motif 라이브러리 `libXm.so.4`에 연결됩니다.
3. `XmText.translations` 자원을 응용프로그램 리소스 파일에 추가하십시오. 리소스가 없으면 로케일의 레이아웃 엔진이 실행되지 않습니다.
4. 설명서에 첨부된 예제 응용프로그램을 참조하십시오.

주 - `fontName` 자원에서 자신의 로케일에 적합한 사용 가능한 글꼴 이름을 사용하십시오.

예를 들어, `XmTextField` 또는 `XmText` 위젯트에서 셀 기반 문자 이동(태국어)을 원하면 해당 위젯트의 번역을 다음과 같이 설정하십시오:

```
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\  

```


mp를 사용한 인쇄 필터 강화

이 장은 mp 유틸리티를 사용한 인쇄 강화에 대해 설명합니다. 이 장에서는 다음 내용을 설명합니다.

- 177 페이지 “UTF-8에 대한 인쇄”
- 178 페이지 “mp 인쇄 필터 강화 개요”
- 180 페이지 “mp.conf 구성 파일을 사용하는 지역화”
- 186 페이지 “Locale-Dependent prolog Files”
- 186 페이지 “프롤로그 파일 추가 및 사용자 정의”
- 186 페이지 “포스트스크립트 파일 사용자 정의”
- 189 페이지 “.xpr Files”

UTF-8에 대한 인쇄

UTF-8로 작성된 텍스트 파일을 포함하는 다양한 입력 파일 형식을 인쇄할 수 있는 향상된 mp 인쇄 필터가 현재 Solaris 환경에서 사용 가능합니다. 이 인쇄 필터는 트루타입과 유형 1 크기 조절 가능 글꼴 및 Solaris 시스템에서 사용할 수 있는 X11 비트맵 글꼴을 사용합니다. 또한 프린터 상주 글꼴을 사용할 수도 있고 X 인쇄 서버 클라이언트 역할을 할 수 있습니다.

유틸리티를 사용하여 표준 포스트스크립트™로 출력한 뒤 포스트스크립트 프린터로 전송할 수 있습니다. mp 유틸리티는 X 인쇄 서버 클라이언트로 구성할 때 모든 페이지 설명 언어를 출력할 수도 있고 mp는 인쇄 서버의 지원을 받습니다.

유틸리티를 사용하려면 다음 명령을 입력하십시오.

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

또한 mp 유틸리티는 stdin 스트림을 받아들이기 때문에 필터로 사용할 수도 있습니다.

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

이 유틸리티를 라인 프린터용 인쇄 필터로 설정할 수 있습니다. 예를 들어, 다음 명령 시퀀스는 프린터 서비스 LP에 프린터 lp1이 mp 형식의 파일만을 받아들인다고 알려줍니다. 이 명령은 또한 프린터 lp1을 /dev/ttya 포트에 설치합니다. 자세한 내용은 lpadm(1M) 설명서 페이지를 참조하십시오.

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

lpfilter(1M)을 사용하여 필터에 대한 유틸리티를 다음과 같이 추가할 수 있습니다.

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

이 명령은 필터 설명 파일인 *pathname*을 통해 변환기(이 경우에는 mp)를 사용할 수 있음을 LP에 알려줍니다. *pathname* contains the following information:

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

필터는 /usr/bin/mp를 사용하여 기본 유형 파일 입력을 포스트스크립트 출력으로 변환합니다.

UTF-8 텍스트 파일을 인쇄하려면 다음 명령을 사용하십시오.

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

자세한 내용은 mp(1) 설명서 페이지를 참조하십시오.

mp 인쇄 필터 강화 개요

mp 인쇄 필터는 현재 Solaris 릴리스에서 향상되었습니다. 최신 mp는 로케ルの 출력 파일을 만드는 세 개의 다른 모드에서 내부적으로 작동하여 국제 텍스트를 인쇄합니다. 사용 가능한 모드는 다음과 같습니다.

- 로케ल 특정 글꼴 구성 파일 mp.conf와 작동
- 로케ल 특정 포스트스크립트 프롤로그 파일 prolog.ps와 작동
- Xprt (X 인쇄 서버) 클라이언트로 작동

다음 절은 특정 인쇄 방법을 사용하는 시기와 이러한 인쇄 방법에 대하여 mp가 어떤 구성 및 지원 파일을 사용하는지 설명합니다.

로케ल 특정 글꼴 구성 파일 mp.conf와 함께 mp 사용하기

만약 -D나 -p가 명령줄에서 주어지지 않았고, prolog.ps 파일이 /usr/openwin/lib/locale/\$LANG/print 또는 /usr/lib/lp/locale/\$LANG/mp에도 없다면 로케ल 특정 글꼴 구성 파일인

mp.conf를 기본적으로 사용합니다. prolog.ps 파일은 파일의 포스트스크립트 내장 글꼴을 사용하여 mp를 인쇄하게 합니다. prolog.ps 가 로캘이 존재하더라도 -M 옵션을 사용하면 prolog.ps 파일이 무시되고 대신 mp.conf 파일이 사용됩니다.

이 방법은 /usr/lib/lp/locale/\$LANG/mp/mp.conf 글꼴 구성 파일을 사용합니다. 대체 글꼴을 사용해야 하는 경우를 제외하고 이 파일을 변경하지 않아도 됩니다. 이 파일은 트루타입, 유형 1 또는 pcf 글꼴로 구성될 수 있습니다.

로캘 특정 포스트스크립트 프롤로그 파일과 함께 mp 사용하기

/usr/lib/lp/locale/C/ 디렉토리는 이 인쇄 모드에 공통된 .ps 인쇄 페이지 레이아웃 파일을 포함합니다. 파일을 사용자 정의하는 방법에 대한 설명은 186 페이지 “프롤로그 파일 추가 및 사용자 정의”에서 제공됩니다.

만약 -D 또는 -P가 명령줄에서 주어지지 않았고, /usr/openwin/lib/locale/\$LANG/print/prolog.ps가 존재하면 prolog.ps 파일이 출력을 위해 미리 결정됩니다. .ps 프롤로그 페이지의 인쇄 스타일에 따라 레이아웃 파일도 출력을 위해 미리 결정됩니다.

이 인쇄 방법은 포스트스크립트 글꼴 파일만 사용합니다. prolog.ps 파일의 사용자 정의는 186 페이지 “프롤로그 파일 추가 및 사용자 정의”에서 설명됩니다.

Xprt(X 인쇄 서버) 클라이언트로 mp 사용하기

mp를 Xprt 클라이언트로 사용하면 mp는 Xprt 인쇄 서비스가 지원하는 네트워크에 연결된 모든 프린터의 출력을 인쇄할 수 있습니다. Xprt 클라이언트로서 mp는 포스트스크립트 및 PCL의 여러 버전을 지원합니다.

Xprt 클라이언트는 다음 규칙을 바탕으로 Xprt 서버에 대하여 연결을 시도합니다

- -D *printer_name@machine[:dispNum]* 또는 -P *printer_name@machine[:dispNum]* 옵션이 mp 명령과 함께 사용되면 mp는 *machine[:dispNum]* with *printer_name*의 Xprt 인쇄 서비스에 연결하려 합니다.

*machine[:dispNum]*에 대하여 위에서 시도한 연결이 실패하거나 -D 또는 -P에 주어진 인자가 *printer_name*이면 mp 명령은 *printer_name* 인자를 지원하는 Xprt 서버에 대하여 XPSERVERLIST를 검사합니다. 예를 들면 다음과 같습니다.

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

- 위 규칙을 사용하여 서버를 발견하지 못하면 mp는 *machine[:dispNum]*로 설정된 XPDISPLAY 환경 변수에 대하여 검사합니다. 예를 들면 다음과 같습니다.

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

- XPDISPLAY 변수가 설정되지 않거나 변수가 유효하지 않으면 mp가 기본 표시 2100에 연결하려 합니다. 기본 표시 값도 유효하지 않으면 mp는 오류 메시지와 함께 종료됩니다.

/usr/lib/lp/locale/C/mp 디렉토리에는 Xprt 클라이언트에 대한 .xpr 인쇄 페이지 예제 레이아웃 파일이 있습니다. 이들은 300dpi 프린터를 위한 예제 파일입니다. 대상 프린터에 dpi 값이 다르면 예제 파일의 dpi 값은 자동으로 대상 프린터의 해상도로 변환됩니다.

mp.conf 구성 파일을 사용하는 지역화

구성 파일은 글꼴 항목 또는 글꼴 그룹 항목을 추가 또는 변경하는 유연성을 제공합니다.

시스템 기본 구성 파일은 /usr/lib/lp/locale/\$LANG/mp/mp.conf이며 여기서 \$LANG은 인쇄가 이루어지는 로케일의 로케일 환경 변수입니다. -u *config.file path* 옵션과 함께 개인 구성 파일을 지정할 수 있습니다.

호환용 문자로 코드화된 슬러 또는 변수 그림 문자를 **표현 형식**이라고 합니다. mp.conf 파일은 로케일의 코드 포인트를 그 코드 포인트를 인쇄하는데 사용되는 글꼴의 코드화된 표현 형식으로 매핑하는데 주로 사용됩니다.

중간 코드 포인트는 넓은 문자이거나 이식 가능 레이아웃 서비스(PLS) 레이어의 출력일 수 있습니다. 복잡한 텍스트 레이아웃 인쇄를 위해서는 중간 코드 포인트가 PLS 출력이어야 합니다. mp가 생성한 기본 중간 코드는 PLS 출력입니다.

현재 지원되는 글꼴 형식은 PCF(Portable Compiled Format), TrueType 및 Type1 형식입니다. 시스템 상주 및 프린터 상주 Type1 글꼴이 모두 지원됩니다. mp.conf 구성 파일의 형식과 내용에 대하여 다음 정보를 기억하십시오.

- 행은 유효한 키보드(지시어)로 시작해야 합니다.
- 키보드에 대한 인자는 키워드와 동일한 행에 나타나야 합니다.
- # 문자로 시작되는 행은 행 끝까지 주석으로 취급됩니다.
- 0x로 시작되는 숫자 인자는 16진수 숫자로 해석됩니다.

mp.conf 파일의 다른 부분은 다음을 포함합니다.

- 글꼴 별명 짓기
- 글꼴 그룹 정의
- 로케일에서 중간 코드 범위에서 글꼴 그룹에 이르는 매핑
- 중간 코드 포인트를 글꼴 코드화의 표현 형식으로 매핑하는 공유 객체와 각 글꼴을 연관

글꼴 별명 짓기

mp.conf 파일의 글꼴 별명 짓기 부분은 인쇄에 사용되는 각 글꼴의 별명을 정의하는데 사용됩니다. 이 절의 각 행은 다음과 같은 형식입니다.

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

font-alias-name

글꼴 이름 별명 짓기의 일반적인 규약은 글꼴의 코드화/스크립트 Roman, Bold, Italic 또는 BoldItalic (R, B, I 또는 BI)인지를 나타내는 문자가 뒤에 오도록 지정하는 것입니다.

예를 들어, /usr/openwin/lib/X11/fonts/75dpi/courR18.pcf.Z, 이것은 iso88591 로마 글꼴이기 때문에 별명 iso88591R이 할당될 수 있습니다.

font-type

가능한 값은 .pcf 글꼴에 대하여 PCF, Adobe Type1 글꼴에 대하여 Type1, TrueType 글꼴에 대하여 TrueType을 지정합니다. 이러한 세 가지 글꼴 종류만 이 mp.config 파일에서 구성될 수 있습니다.

font-path

글꼴 파일에 절대 경로 이름을 부여합니다. 유형1 프린터 상주 글꼴에 대해서는 Helvetica와 같은 글꼴 이름을 지정합니다.

예를 들어,

```
FontNameAlias prnHelveticaR Type1 Helvetica  
입니다.
```

글꼴 그룹 정의

동일한 유형의 글꼴을 조합하여 글꼴 그룹을 형성할 수 있습니다. 글꼴 그룹의 형식은 다음과 같습니다.

<i>keyword</i>	FontGroup.
<i>fontgroupname</i>	글꼴의 그룹 이름
<i>GroupType</i>	글꼴 유형동일한 유형의 글꼴(PCF, Type1, TrueType)에 대해서만 글꼴 그룹을 만드십시오.
<i>Roman</i>	글꼴 그룹의 로마 글꼴 이름
<i>Bold</i>	글꼴 그룹의 굵은체 글꼴 이름
<i>Italic</i>	글꼴 그룹의 기울임꼴 글꼴 이름
<i>BoldItalic</i>	글꼴 그룹의 굵은 기울임 글꼴 이름.

그룹을 만들려면 로마 글꼴 항목만 필요합니다. 굵은체, 기울임꼴 및 굵은 기울임 글꼴은 선택적입니다. 다른 유형의 글꼴은 예를 들어, 우편/뉴스 기사의 헤더 행을 표시하는데 사용됩니다. 로마 글꼴만 정의되면 다른 글꼴 대신 사용됩니다.

매핑 부분

mp.conf 파일의 매핑 부분은 로켈에서 중간 코드 범위를 글꼴 그룹에 매핑합니다. 이 절의 각 행에 대한 형식은 다음과 같습니다.

<i>keyword</i>	MapCode2글꼴.
<i>range_start</i>	0x로 시작되는 4바이트 16진수 값. 하나 이상의 글꼴 그룹에 매핑되는 코드 범위의 시작을 나타냅니다.
<i>range_end</i>	매핑될 코드 범위의 끝을 나타냅니다. 단일 중간 코드 포인트가 대상 글꼴에 매핑될 때는 '-' 값만 가능합니다.

group 표현 형식이 인쇄될 Type1, PCF 또는 TrueType 글꼴 그룹.

연관 부분

mp.conf 파일의 연관 부분은 각 글꼴을 글꼴 코드화의 표현 형식에 중간 코드 포인트를 매핑하는 공유 객체와 연관시킵니다. 이 절의 각 행에 대한 형식은 다음과 같습니다.

<i>keyword</i>	CnvCode2Font.
<i>font alias name</i>	글꼴에 정의된 별명
<i>mapping function</i>	중간 코드를 받아들이고 글꼴 코드화의 표현 형식을 반환하면 이것은 그림 문자 색인을 얻고 그림 문자를 그리는 데 사용됩니다.
파일 경로 매핑 함수	매핑 기능을 포함하는 .so 파일 이름dumpcs를 사용하여 EUC 로컬의 중간 코드 세트를 찾을 수 있습니다.

주 - mp(1)이 사용하는 현재 트루타입 엔진은 형식 4와 PlatformID 3 cmap만 다룰 수 있습니다. Microsoft .tff 파일만 구성할 수 있습니다. 추가로, 문자 맵 코드화인 트루타입 글꼴 엔진이 제대로 작동하기 위해 유니코드 또는 기호여야 합니다. Solaris 환경의 .tff 글꼴 대부분은 이러한 제약을 준수하기 때문에 mp.conf 파일 내에서 Solaris 소프트웨어의 모든 트루타입 글꼴을 매핑할 수 있습니다.

PCF 유형1 X Logical Fonts Description (XLFD)에 해당하는 글꼴에 매핑하는 공유 객체를 작성할 수 있습니다. 그런 다음 중간 코드 범위에서 XLFD이 지정하는 코드화로 매핑하는 공유 객체를 작성할 수 있습니다. 예를 들면 다음과 같습니다.

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

해당 PCF 글꼴은 다음과 같습니다.

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

이 글꼴은 ISO 8859-8로 코드화되므로 공유 객체는 중간 코드와 해당 ISO 8859-8 코드 포인트 사이를 매핑해야 합니다.

XLFD가 있는 트루타입 글꼴의 경우

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

해당 글꼴은 다음과 같습니다.

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

이전 트루타입 글꼴의 cmap 코드화가 유니코드로 되어 있기 때문에 중간 코드와 유니코드 사이를 매핑해야 합니다. 트루타입 글꼴의 예제에서 히브리 문자(PLS 레이어가 만듦)에 해당하는 en_US.UTF-8 로캘 예제 중간 코드가 0xe50000e9라고 가정합니다. 글꼴이 유니코드로 코드화되기 때문에 0xe50000e9를 전달할 때 출력이 유니코드의 표현 형식에 해당하는 방식으로 해당 .so 모듈 내에서 함수를 설계하십시오. 현재 예제는 0x000005d9입니다.

매핑 함수에 대한 함수 프로토타입은 다음과 같아야 합니다.

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

다음은 mp.conf에서 사용할 수 있는 선택적 키워드/값 쌍입니다.

```
PresentationForm WC/PLSOutput
```

기본값은 PLSOutput입니다. 사용자가 WC를 지정하면 생성되는 중간 코드는 넓은 문자입니다. CTL 인쇄의 경우 이 기본값이 사용되어야 합니다.

로캘이 비CTL 로캘이고 키보드 값이 PLSOutput이면 해당 값은 무시되고 대신 mp(1)가 넓은 문자 코드를 생성합니다.

로캘이 CTL을 지원하는 경우 다음 표에 나열된 선택적 키워드/값 쌍을 사용할 수 있습니다. 이러한 변수는 테이블의 중간 열에 주어진 모든 가능한 값을 가정할 수 있습니다.

표 7-1 선택적 키워드/값 쌍

선택적 키워드	선택적 값	기본값
방향	ORIENTATION_LTR/	ORIENTATION_LTR
	ORIENTATION_RTL/	
	ORIENTATION_CONTEXTUAL	
숫자	NUMERALS_NOMINAL/	NUMERALS_NOMINAL
	NUMERALS_NATIONAL/	
	NUMERALS_CONTEXTUAL	
TextShaping	TEXT_SHAPED/	TEXT_SHAPED
	TEXT_NOMINAL/	
	TEXT_SHFORM1/	
	TEXT_SHFORM2/	
	TEXT_SHFORM3/	
	TEXT_SHFORM4	

▼ 프린터 상주 글꼴을 추가하는 방법

다음 절차의 예제는 새로운 PCF, 트루타입 또는 유형1 프린터 상주 글꼴을 구성 파일에 추가하는 방법을 보여줍니다.

이 절차를 완료하여 현재 구성된 글꼴을 대체합니다. 첫번째 두 단계에서 범위 0x00000021 - 0x0000007f에 문자를 표시하는 데 사용되는 PCF 글꼴은 트루타입 글꼴로 대체됩니다.

1. 새 글꼴을 추가하기 전에 다음과 같이 현재 구성된 글꼴에 해당하는 구성 파일에서 다양한 구성 요소를 찾으십시오.

```
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
예를 들어,
/usr/openwin/lib/locale/ja/X11/fonts/TT/HG-MinchoL.ttf 글꼴을
en_US.UTF-8 로캘에 매핑할 수 있습니다. HG-MinchoL.ttf은 유니코드 트루타
입 글꼴 파일이기 때문에 .so 모듈 매핑 기능을 사용하여 직접 받는 ucs-2 코드 포
인트로 돌아옵니다.
unsigned short _ttfjis0201(unsigned short ucs2) {
    return(ucs2);
}
```

a. `ttfjis0201.c` 파일에 매핑을 저장합니다.

b. 공유 객체 파일을 작성합니다.

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

2. `/usr/openwin/lib/locale/ja/X11/fonts/75dpi/gotmrk20.pcf.z`와 같
은 PCF 파일을 매핑하려면
`/usr/openwin/lib/locale/ja/X11/fonts/75dpi/fonts.dir` 파일의
XLFD에 해당하는 다음 코드화를 확인하십시오.

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

a. `jisx0201`은 코드화이기 때문에 `ucs-2`에서 `jisx0201`로 매핑되는 공유 객체를
준비하십시오. `.so` 모듈을 작성하기 위한 매핑 테이블을 얻으십시오. 유니코드
로캘의 경우 `ftp.unicode.org/pub/MAPPINGS/` 디렉토리의 유니코드에 대

한 문자 세트 매핑을 찾으십시오.

b. 이 매핑을 사용하여 `xu2jis0201.c` 파일을 작성하십시오.

```
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. 매핑 파일을 작성할 때 모든 `usc-2`를 `jisx0201` 케이스에 포함시키십시오.

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

▼ 공유 객체 파일을 작성하는 방법

다음 절차의 예제는 공유 객체 파일을 작성하는 방법을 보여줍니다.

1. 글꼴을 추가하려면 `mp.conf` 파일의 섹션에 해당하는 다음 예제의 행을 편집하십시오.

다음 예제는 트루타입 글꼴을 추가하는 방법을 보여줍니다. `.so` 경로는 `xu2jis0201.so` 파일을 가리킵니다.

```
FontNameAlias    jis0201R TrueType /home/fn/HG-Minchol.ttf
FontGroup        jis0201 TrueType jis0201R
MapCode2Font     0x0020      0x007f jis0201
CnvCode2Font     jis0201R      _ttfjis0201 <.so path>
```

주 - PCF 글꼴을 추가하려면 키워드를 트루타입에서 PCF로 변경하십시오.

2. 변경된 `mp.conf` 파일로 `mp` 명령을 호출하여 범위 `0x0020-0x007f`를 새 글꼴에서 인쇄하십시오.

다른 일본어 문자 범위도 동일하게 매핑합니다. 예를 들어 `so` 파일은, `0x0000FF61`에서 `0x0000FF9F` 범위를 매핑할 수 있습니다.

주 - 역방향 호환성을 유지하려면

`/usr/openwin/lib/locale/$LANG/print/prolog.ps` 파일을 사용하여 현재 로케일에서 출력을 작성할 수 있습니다. `prolog.ps` 파일을 사용할 때 구성 파일이 필요합니다.

`/usr/lib/lp/locale/en_US.UTF-8/mp` 디렉토리에서 예제 `mp.conf` 파일을 찾을 수 있습니다.

프롤로그 파일 추가 및 사용자 정의

프롤로그 파일은 다음 두 가지 주요한 범주로 나눌 수 있습니다.

- 포스트스크립트 프롤로그 파일 (.ps)
- X 인쇄 서버 클라이언트 프롤로그 파일 (.xpr)

포스트스크립트 파일 사용자 정의

포스트스크립트 파일은 다음 범주로 나뉩니다.

- 공통 프롤로그 파일
- 인쇄 레이아웃 프롤로그 파일

Locale-Dependent prolog Files

prolog.ps 파일은 흔히 사용되지 않는 글꼴을 설정하기 위해 사용됩니다. 응용프로그램은 인쇄를 위해 미리 정의된 포스트스크립트 글꼴 이름을 사용합니다. 프롤로그 파일은 데스크 세트 캐린더 관리자 및 mp에 대하여 최소한 다음 글꼴 이름을 정의해야 합니다.

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

다음 예제는 지정된 특정한 로컬 문자를 인쇄하기 위해 다음 글꼴을 사용합니다.

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

Solaris 지역화 키트는 일본어 환경에 대해 예제 prolog.ps 파일을 제공합니다. 대신, 이 파일은 /usr/openwin/lib/locale/ja/print/ 디렉토리에 있습니다.

다음 예제는 기존 prolog.ps 파일에서 복합 글꼴을 추가 또는 변경하는 방법을 보여줍니다.

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

예를 들어, LC_Base-Font라는 복합 글꼴을 정의할 수 있습니다. LC_Base-Font는 로컬 문자 세트와 Base-Font를 포함하는 Foo-Fine 글꼴의 복합입니다. 글꼴을 추가 또는 변경하기 위해서는 포스트스크립트 프로그래밍에 대한 깊이 있는 지식이 필요하지 않습니다.

prolog.ps 파일을 작성하는 최선의 방법은 예제 버전을 연구하는 것입니다. prolog.ps의 예에서는 두 가지 루틴 즉,makecodeset12 및 makeEUCfont이 작성되어야 합니다. 루틴 makecodeset12는 로컬 글꼴 코드화 정보를 설정합니다. 이 루틴은 로컬에 따라 다릅니다. 루틴 makeEUCfont는 기반 글꼴과 로컬 글꼴을 조합하여 복합 글꼴을 형성합니다. 프롤로그 파일의 작성자는 makecodeset12 및 makeEUCfont를 작성하기 위해 포스트스크립트를 잘 알고 있어야 합니다.

prolog.ps 파일 지원은 역방향 호환성을 위해서만 유지됩니다. 로컬에 대한 인쇄 출력을 생성하기 위해 새로운 prolog.ps 파일을 작성하지 마십시오. 대신 mp.conf를 사용하십시오.

prolog.ps 파일의 경로는 다음과 같습니다.

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

공통 포스트스크립트 프롤로그 파일

공통 프롤로그 파일은 mp.common.ps 입니다.

모든 다른 페이지 레이아웃 프롤로그 파일은 이 파일을 포함해야 합니다.

mp.common.ps 파일은 /usr/lib/lp/locale/C/mp/ 디렉토리에 상주합니다. 이 파일은 글꼴을 표준 코드화에서 ISO 8859-1 코드화로 다시 코드화하는 포스트스크립트 루틴을 포함합니다. .reencodeISO 루틴은 인쇄 레이아웃 프롤로그 파일에서 호출되어 글꼴 코드화를 변경합니다. 보통 이 프롤로그 파일은 사용자 정의가 필요하지 않습니다. 사용자가 자신의 프롤로그 파일을 작성 중이면 환경 변수 PROLOGUE가 수정된 프롤로그 파일을 포함하는 디렉토리를 가리키도록 설정합니다.

인쇄 레이아웃 프롤로그 파일

인쇄 레이아웃 프롤로그 파일, mp.*.ps 파일은 인쇄용 페이지 레이아웃을 제어하는 루틴을 포함합니다. 사용자 이름, 인쇄 날짜, 페이지 번호가 있는 인쇄 페이지의 헤더 및 푸터 외에 프롤로그 파일은 다른 정보를 제공할 수 있습니다. 예를 들어, 프롤로그 파일은 유효 인쇄 영역 크기와 가로 및 세로 인쇄 모드를 제공할 수 있습니다.

인쇄 레이아웃 프롤로그 파일은 다음과 같습니다.

- 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

표준 기능 세트가 모든 프롤로그 파일에서 정의되어야 합니다. 함수는 새 인쇄 페이지가 시작, 인쇄 페이지가 종료 또는 새 열이 종료될 때 호출됩니다. 이러한 함수의 구현은 인쇄 출력의 인쇄 속성을 정의합니다.

다음 포스트스크립트 변수는 mp 바이너리에 의해 런타임에 정의됩니다. 모든 인쇄 레이아웃 파일은 사용자 이름, 주제, 인쇄 시간과 같은 동적 정보를 인쇄하기 위해 이러한 변수를 사용할 수 있습니다. 변수에서 취해진 정보는 일반적으로 인쇄 페이지의 헤더 또는 푸터에 나타납니다.

<i>User</i>	시스템 passwd 파일에서 얻은 mp를 실행하는 사용자의 이름
<i>MailFor</i>	인쇄할 항목의 유형 이름을 보유하는데 사용되는 변수. 이 변수의 가능한 값은 다음과 같습니다. <ul style="list-style-type: none"> ■ Listing for-- 입력이 텍스트 파일일 때 ■ Mail for-- 입력이 우편 파일일 때 ■ Article from-- 입력이 새 그룹의 항목일 때
<i>Subject</i>	우편 및 뉴스 헤더에서 취해진 주제 - s 옵션을 사용하여 일반 텍스트 파일은 물론 우편 및 뉴스 파일에 주제를 강제할 수 있습니다.
<i>Timenow</i>	헤더와 푸터에 나타나는 인쇄 시간이 정보는 localtime() 함수에서 취해집니다.

다음 함수는 프롤로그 파일의 인쇄 레이아웃에서 구현됩니다. 이러한 모든 함수는 부속 함수를 사용할 수 있습니다.

endpage	<p>사용법: page_number endpage</p> <p>인쇄된 페이지의 맨아래에 도달하면 호출됩니다. 이 함수는 페이지의 그래픽 컨텍스트를 복원하고 showpage를 실행합니다. 일부 prolog 파일에서는 헤더 및 푸터 정보가 컬럼별 모드가 아닌 페이지별 모드에서 표시됩니다. 이 함수를 구현하여 헤더 및 푸터 회색조 lozenges를 표시하는 부속 함수를 호출할 수 있습니다.</p>
newpage	<p>사용법: page_number newpage</p> <p>새 페이지가 시작할 때 실행될 루틴 또는 명령수직 인쇄 모드 설정, 인쇄 그래픽 컨텍스트 저장 및 페이지 좌표 변환은 이러한 루틴 기능의 일부입니다.</p>
endcol	<p>사용법: page_number col_number endcol</p>

헤더 및 푸터 정보를 표시하고 새 인쇄 위치로 이동하는 등에 사용됩니다.

새 인쇄 레이아웃 프롤로그 파일 추가를 위해 인쇄 레이아웃 프롤로그 파일 내부에서 다음 변수를 정의해야 합니다.

NumCols 인쇄 페이지의 열 수. 기본값은 2입니다.
PrintWidth 인치 단위 인쇄 영역 너비. 기본값은 6입니다.
PrintHeight 인치 단위 인쇄 영역 높이. 기본값은 9입니다.

.xpr Files

파일은 기본적으로 `/usr/lib/lp/locale/C/mp/` 에 위치합니다. `.xpr` 파일은 `mp.common.ps` 파일을 제외하고 각 포스트스크립트 프롤로그 레이아웃 파일에 해당합니다. `MP_PROLOGUE` 환경 변수를 정의하여 대체 프롤로그 디렉토리를 정의할 수 있습니다.

파일은 키워드/값 쌍으로 작동합니다. #로 시작하는 행은 주석으로 간주됩니다. 공백은 명시적으로 진술된 경우를 제외하고 다른 토큰을 분리합니다. 각 `.xpr` 파일의 세 개의 주 부분은 다음 키워드 쌍으로 묶입니다.

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

특정 키워드/값 쌍은 세 영역에서 사용됩니다. 각 영역은 다음 절에서 설명됩니다.

STARTCOMMON/ENDCOMMON 키워드

STARTCOMMON 키워드 뒤, ENDCOMMON 키워드 앞에 나타나는 모든 키워드/값 쌍은 인쇄 페이지의 일반 등록 정보를 정의합니다. 키워드에 대한 다른 유효한 값은 슬래시(/) 문자를 사용하여 구분됩니다.

ORIENTATION 0/1

0은 수직으로 인쇄가 이루어짐을 뜻하고 1은 수평으로 이루어짐을 뜻합니다.

PAGELength **부호없는 정수**

논리적 페이지당 행 수를 나타내는 값

LINELENGTH **부호없는 정수**

행당 단일 열 문자 수를 나타내는 값

NUMCOLS *unsigned-integer*

물리적 페이지당 논리적 페이지 수

HDNGFONTSIZE 부호없는 정수

10포인트 단위의 머리글 글꼴 포인트 크기

BODYFONTSIZE 부호없는 정수

10포인트 단위의 본문 글꼴 포인트

PROLOGDPI 부호없는 정수

현재 .xpr 파일이 작성된 인치당 도트 눈금

YTEXTBOUNDARY 부호없는 정수

y 좌표는 페이지 또는 논리적 페이지(열)의 텍스트 인쇄 경계를 설정합니다. 경계는 텍스트 인쇄가 예상 영역 내에 있는지 확인하는 추가 검사로 사용됩니다. 경계는 해당 글꼴에서 얻은 문자 높이 정보가 잘못되어있기 때문에 복잡한 텍스트 레이아웃 및 EUC 인쇄에 필요합니다.

STARTTEXT 부호없는 정수부호없는 정수

물리적 페이지의 첫번째 논리적 페이지에서 실제 텍스트 인쇄가 시작되는 10포인트 x/y 포인트

PAGESTRING 0/1

1은 페이지 문자열이 머리글의 페이지 수 전에 추가되어야 함을 나타냅니다.

0은 페이지 수만 표시됨을 나타냅니다.

EXTRAHDNGFONT font string 1, font string 2, ... font string n

글꼴 문자열은 X 논리적 글꼴 설명입니다. 쉼표 구분 글꼴 이름 목록에서 키워드 EXTRAHDNGFONT를 구분하는 토큰은 스페이스나 탭이 아닌 " 문자입니다. 글꼴은 머리글이 인쇄될 때 내장 글꼴에 대한 환경 설정이 주어집니다. 일반적으로, EXTRABODYFONT는

/usr/openwin/server/etc/XpConfig/C/print/models/<model name>/fonts 디렉토리에 구성된 프린터 상주 글꼴을 할당하는데 사용됩니다.

fonts.dir 파일은 프린터 상주 글꼴의 XLFD를 포함합니다.

.xpr 파일에서 글꼴은 일반적으로 다음 예제에서 나타나듯이 지정됩니다.

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

%d"가 있을 경우 mp(1)이 .xpr 파일의 현재 머리글 글꼴의 포인트 크기로 바꿉니다. x 해상도와 y 해상도는 *로 지정됩니다. 평균 너비 필드는 0으로 설정되어 가능한 경우 확장 가능 글꼴의 선택을 나타냅니다. 보다 특정한 글꼴 이름을 제공할 수도 있습니다.

EXTRABODYFONT 글꼴 문자열 1, 글꼴 문자열 2, ... 글꼴 문자열 n

글꼴이 페이지 본문 인쇄에 사용된다는 것을 제외하고 EXTRAHDNGFONT와 동일합니다.

XDISPLACEMENT 부호있는/부호없는 정수

x 방향으로 페이지 내용을 이동하기 위해 페이지에 적용될 x 좌표 변위를 제공합니다. 이 변위는 +ve 또는 -ve 값일 수 있습니다.

YDISPLACEMENT 부호있는/부호없는 정수

이동이 y 방향으로 일어난다는 것을 제외하고 x 변위와 동일합니다.

두 키워드는 페이지에서 인쇄된 내용을 이동해야 하는 비표준 여백 너비를 가진 일부 프린터를 다룰 때 유용합니다.

STARTPAGE/ENDPAGE 키워드

이 부분의 키워드 값 쌍은 STARTPAGE 및 ENDPAGE 키워드로 묶입니다. 이 부분은 물리적 페이지에 적용될 그리기 및 머리글 정보를 포함합니다. 물리적 페이지는 많은 논리적 페이지를 포함할 수 있지만 키워드 사이에 포함되는 모든 그리기 루틴은 물리적 페이지에 한 번만 적용됩니다.

유효한 그리기 엔티티는 LINE 및 ARC입니다. XDrawLine() 및 XDrawArc() 함수는 키워드 값에서 실행됩니다.

이 부분 내의 크기는 PROLOGDPI 단위로 매핑됩니다. 각도는 도 단위입니다.

LINE x1 y1 x2 y2	x/y 부호없는 좌표는 선 연결을 위한 포인트 쌍을 정의합니다.
ARC x y 너비 높이 각도1 각도2	x 및 y는 모두 원호 원점을 표현하는 부호없는 정수입니다. 너비 및 높이는 원호의 너비와 높이를 표현하는 부호없는 정수입니다.
USERSTRINGPOS x y	부호없는 좌표는 사용자 정보가 머리글에 인쇄되는 위치를 표시합니다.
TIMESTRINGPOS x y	부호없는 좌표는 인쇄 시간이 머리글에 인쇄되는 위치를 표시합니다.
PAGESTRINGPOS x y	부호없는 좌표는 각 인쇄된 페이지에 대한 페이지 문자열을 인쇄하는 위치를 표시합니다.
SUBJECTSTRINGPOS x y	부호없는 좌표는 페이지의 주제를 인쇄하는 위치를 표시합니다.

STARTFORCEDPAGE/ENDFORCEDPAGE 부분

-n 옵션이 mp에 주어지면 STARTPAGE/ENDPAGE 부분 내에서 모든 장식이 주어집니다. 그러나 -n 옵션이 주어져도 STARTFORCEDPAGE/ENDFORCEDPAGE 부분 내에 포함된 모든 것이 인쇄됩니다.

STARTCOLUMN/ENDCOLUMN 부분

모든 키워드는 이 부분의 항목이 물리적 페이지에 대한 NUMCOLS 횟수에 적용된다는 것을 제외하고 191 페이지 "STARTPAGE/ENDPAGE 키워드"의 설명과 동일합니다. NUMCOLS이 3이면 물리적 페이지의 인쇄 가능 영역이 셋으로 분할되고 선, 원호 또는 머리글 장식이 페이지당 세 번 나타납니다.

STARTFORCEDCOLUMN/ENDFORCEDCOLUMN 부분

-n 옵션이 mp에 주어지면 STARTCOLUMN/ENDCOLUMN 부분 내에서 주어진 모든 장식이 인쇄되지 않습니다. 그러나 -n 옵션이 주어지지더라도 STARTFORCEDCOLUMN/ENDFORCEDCOLUMN 부분 내에 포함된 모든 것이 인쇄됩니다.

새 .xpr 파일 작성하기

새 .xpr 프롤로그 파일을 작성할 때 기본값과 다른 값만 지정해야 합니다.

다음 표 목록은 이러한 값이 STARTCOMMON/ENDCOMMON 부분에 대한 .xpr 파일에 지정되지 않더라도 다른 키워드에 대한 mp 프로그램 기본값입니다.

표 7-2 STARTCOMMON/ENDCOMMON 키워드 값

키워드	값
ORIENTATION	0
PAGELength	60
LINELENGTH	80
YTEXTBOUNDARY	3005
NUMCOLS	01
HDNGFONTsize	120
PROLOGDPI	300
STARTTEXT	135 280
PAGESTRING	0

STARTPAGE/ENDPAGE 및 STARTCOLUMN/ENDCOLUMN에 묶이는 다른 두 부분을 위해 기본값이 필요하지 않습니다.

장식이 없는 페이지를 작성하려면 세로 형식으로 물리적 페이지당 네 개의 논리적 페이지를 사용합니다. 다음 부분 및 값을 지정합니다.

```
STARTCOMMON
NUMCOLS 04
LINELENGTH 20
ENDCOMMON
```

장식이 없는 페이지를 작성할 때는 다음 두 부분을 지정할 필요가 없습니다.

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


매개 변수는 인쇄된 페이지에 장식을 두지 않을 때 필요합니다. 모든 좌표는 PROLOGDPI 키워드를 지정하지 않는 한 기본적으로 300dpi 범위입니다. 대상 프린터 해상도가 다른 .xpr 파일은 프로그램에 의해 해당 해상도에 맞도록 조정됩니다.

.xpr 파일을 작성하기 전에 먼저 용지 크기를 알아야 합니다. 미국 용지의 경우 해상도 300dpi 프린터에 대하여 8.5x11인치, 2550X3300이 전체 크기입니다. 대부분의 프린터는 용지의 맨 위 왼쪽 모서리부터 인쇄되지 않습니다. 대신, 물리적 용지 주변에 일부 여백 공간이 할당됩니다. 이것은 0,0에서 인쇄를 하려해도 인쇄는 페이지의 맨 위 왼쪽 모서리에서 이루어지지 않음을 뜻합니다. 새 .xpr 파일을 작성할 때 이 제한을 고려하십시오.

부록 A

iconv 코드 변환

이 부록은 현재 Solaris 운영 환경에서 사용 가능한 유니코드 관련 코드 변환 모듈을 나열합니다.

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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)

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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)

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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)

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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)

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-1 사용 가능한 유니코드 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

주 - UTF-EBCDIC은 새로운 IBM 코드 페이지 이름입니다. 현재 Solaris 환경은 UTF-8과 UTF-EBCDIC 코드 페이지 사이의 양방향 변환도 지원합니다.

다음 표는 현재 Solaris 환경에서 사용 가능한 유니코드 및 IBM/Microsoft EBCDIC 및 PC iconv 코드 변환 모듈을 나열합니다.

표 A-2 사용 가능한 유니코드 및 IBM/Microsoft EBCDIC 및 PC 코드 페이지 관련 iconv 코드 변환 모듈

원본 코드(기호)	대상 코드(기호)
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

표 A-2 사용 가능한 유니코드 및 IBM/Microsoft EBCDIC 및 PC 코드 페이지 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
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

표 A-2 사용 가능한 유니코드 및 IBM/Microsoft EBCDIC 및 PC 코드 페이지 관련 iconv 코드 변환 모듈 (계속)

원본 코드(기호)	대상 코드(기호)
UTF-8	CP1255
UTF-8	CP1256
UTF-8	CP1257
UTF-8	CP1258

다음 표는 UTF-8에 대한 사용 가능한 iconv 코드 변환 IBM 및 Microsoft EBCDIC/PC 코드 페이지를 나열합니다.

표 A-3 UTF-8에 대한 사용 가능한 iconv 코드 변환 - IBM 및 Microsoft EBCDIC/PC 코드 페이지

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

표 A-3 UTF-8에 대한 사용 가능한 iconv 코드 변환 - IBM 및 Microsoft EBCDIC/PC 코드 페이지 (계속)

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

표 A-3 UTF-8에 대한 사용 가능한 iconv 코드 변환 - IBM 및 Microsoft EBCDIC/PC 코드 페이지 (계속)

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

색인

번호와 기호

16비트 유니코드 3.0 코드 세트, 170

A

API, 국제화, 41

B

Bengali 키보드, 82

C

C 로케일, 24

CDE

로케일 지원, 112

입력 방법, 114

code set, 문자 지원, 113

Complex Text Layout (CTL), 157-176

Motif, 159-168

Motif 라이브러리, 176

XOC 자원, 159

XOM, 158

구별 부호, 157-176

구조, 158

레이아웃 방향, 168-172

렌더 테이블 만들기, 172-173

마우스 선택, 174

수평 탭, 173-174

키보드 선택, 175

Complex Text Layout (CTL) (계속)

텍스트 방향, 157-176

텍스트 자원, 175

합자, 157-176

Complex Text Layout(CTL)

Rendition 편집, 171

렌더링 만들기, 170

Compose 키, 33

Compose 키 시퀀스

Latin-1, 118

Latin-2, 122

Latin-3, 123

Latin-4, 124

Latin-5, 125

Latin-9, 125

그?스어, 3개 키, 141

그?스어, 4개 키, 142

그리스어, 135

악센트 사용 불능 키, 126

.cshrc, STREAMS 모듈 설정, 150

ctype 매크로, 40

D

Devanagari 키보드, 82

DtMail, MIME 문자 세트, 151

dtterm, 149

E

en_US.UTF-8, FontSet 정의, 154

en_US.UTF-8, 지원, 112

F

FontSet 정의, 154

G

genmsg 유틸리티, 48

GMT 오프셋, 27

Gujarati 키보드, 82

Gurmukhi 키보드, 83

H

Hanzi, 32

I

iconv

일본어 문자 코드 변환, 80

코드 변환, 50

iconv 변환 모듈, EBCDIC/PC 코드 페이지, 219-221

iconv 변환 모드, 유니코드, 195-216

input 방법, 인도어 방법, 81

ISO 라틴-1, 23

ISO8859, 문자 지원, 113

K

Kannada 키보드, 83

Kedmanee 키보드, 108

향상, 109

L

LANG 환경 변수, 146

Latin-1, Compose 키 시퀀스, 118

Latin-2, Compose 키 시퀀스, 122

Latin-3, Compose 키 시퀀스, 123

Latin-4, Compose 키 시퀀스, 124

Latin-5, Compose 키 시퀀스, 125

Latin-9, Compose 키 시퀀스, 125

LC_ALL, 23

libc

API, 37-38

날짜 및 시간 형식, 44

넓은 문자 및 문자열 처리, 45

넓은 문자 입력 및 출력, 46, 47

넓은 문자 클래스, 42

넓은 문자열, 47

로케일 수정 및 질의, 42

로케일 질의, 42

멀티바이트 처리, 45

메시징 기능, 41

문자 분류 및 음역, 43

문자 조합, 44

응용 프로그램 연결, 35-51

정규 표현식, 42

코드 변환 기능, 41

통화 형식, 44

linking, 응용 프로그램, 39

M

Malayalam 키보드, 84

map, Kannada 문자, 96

mbtwoc, 41-47

Motif

TextField, 169

UIL 인자, 168

XmNalignment, 163, 169, 170

XmNeditPolicy, 163

XmNlayoutDirection, 160

XmNlayoutDirection, 169

XmNlayoutModifier, 169, 170

XmNrendition 태그, 163

XmRendition, 161, 169

XmStringDirection, 161

XmStringDirectionCreate, 167

XmText, 169

XmTextFieldGetLayoutModifier, 166

XmTextFieldSetLayoutModifier, 167

XmTextGetLayoutModifier, 166

XmTextSetLayoutModifier, 167

mp

Xprt 클라이언트, 179

인쇄 필터, 177, 178

mp (계속)
트루타입, 18
포스트스크립트 변수, 188
mp.conf 파일, 180

P

Pattajoti 키보드, 109
PinYin, 33
Portable Layout Services (PLS), 180
POSIX 로캘, 24

S

setlocale command, 146
SPARC 키보드, 66
strconf command, 149
STREAMS
TTY 환경, 147
로딩 모듈, 147
코드 변환, 147
stty, 유틸리티, 149
Sun Ray, USB 유형 6 키보드, 19

T

Tamil 키보드, 84
Teluga 키보드, 84
TTY 환경, 설정, 147
TypeOfText, 162

U

UIL, 168
UTC, 27
UTF-8, 지원, 112

X

X Logical Fonts Description (XLFD), 182
X 인쇄 서버(Xprt), mp, 179
XmText
>left-character(extend), 164

XmText (계속)
backward-cell(extend), 166
delete-left-character(), 165
delete-right-character(extend)
, 165
forward-cell(extend), 166
right-character(extend), 164
right-word(extend), 165
XPG4 응용프로그램, 40
.xpr 파일, 186
xterm, 149

가

가타카나, 31

간

간지, 31
간체 한자
문자 지원, 113
입력 모드, 144

국

국제화
ISO 라틴-1, 23
정의, 21-23
국제화 API, 41
국제화된 도메인 이름(IDN), 49

그

그리스 UNIX, 키보드, 134
그리스 유럽, 키보드, 134
그리스어
문자 지원, 113
입력 모드, 134, 135

글

글꼴
매핑, 180

클짚 (계속)
별명 짓기, 180
일본어 비트맵, 78
일본어 트루 타입, 79

날
날짜 형식, 27

넓
넓은 문자, 지원, 41-47

네
네델란드(네델란드어) 키보드, 69

노
노르웨이어 키보드, 69

단
단말기
설정 옵션, 149
지원, 149

덴
덴마크어 키보드, 67

독
독일어, 68
문자 지원, 113

라
라이브러리, (CSI), 37

러
러시아어, 문자 지원, 113

레
레이아웃 동작, 162

로
로컬 환경 변수, 146
로컬

C, 24
Compose 키, 60
POSIX, 24
Solaris, 77-110
날짜 형식, 27
남미, 59
남유럽, 59
단어 구분자, 30
동유럽, 56
문자 세트, 31
문화적 규약, 25
범주, 25
부분, 24, 53
북미, 58
북부 유럽, 58
북아프리카, 57
서유럽, 60
숫자 형식, 27
시간 형식, 26
아시아, 54
오스트랄라시아, 55
일본어, 77
전체, 24, 53
정렬 순서, 31
정의, 23
중동, 57
중앙 아메리카, 56
키보드 차이, 33
통화, 21
통화 형식, 28
페이지 크기, 34
환경 변수, 146

매

매핑, 영어와 인도어 스크립트용 음성 해당자, 85

맵

Bengali 문자, 86
Gujarati 문자, 88
Gurmukhi 문자, 91
Malayalam 문자, 99
Telugu 문자, 105
타밀 문자, 102
힌두 문자, 94

멀

멀티바이트, 변환, 41-47
멀티바이트 문자, 38

메

메시지 카탈로그, 48

명

명령, CSI, 36-37

문

문자
멀티바이트, 38
변환, 41-47
지원, 113
형성, 157-176
문자 코드화
UTF-16, 112
UTF-32, 112
UTF-8, 112
유니코드, 111

미

미국/UNIX 키보드, 73
미국 키보드, 73

발

발트어, 문자 지원, 113

벨

벨기에 키보드, 66

변

변환
사용자 정의 코드 세트, 49
유니코드 iconv 모듈, 195-216

복

복합 텍스트 레이아웃(CTL), 14

서

서부 유럽, 문자 지원, 113
서유럽, 문자 지원, 113

스

스웨덴어 키보드, 70
스위스(독일어) 키보드, 71
스위스(프랑스어) 키보드, 71
스칸디나비아어, 문자 지원, 113
스크립트 선택, 115
스페인어, 문자 지원, 113
스페인어 키보드, 70

시

시간 형식, 26
시간대, 27

시스템 라이브러리, 에 대한 링크 만들기 응용 프로그램, 39

아

아랍어
문자 지원, 113
입력 모드, 132
아랍어 키보드, 66

알

알파벳, 31

연

연속 음성 입력 방법, 인도어 스크립트, 107

영

영국 키보드, 72
영어
문자 지원, 113
입력 모드, 117

유

유니코드
16진법 입력 모드, 145
개요, 111
변환 모듈, 195-216
유틸리티
genmsg, 48
genmsg, 48
iconv, 50
stty, 149
로캘, 146

응

응용 프로그램
FontSet/XmFontList 정의, 154

응용 프로그램 (계속)
시스템 라이브러리에 대한 링크 만들기, 39
응용프로그램, XPG4, 40

이

이탈리아어 키보드, 68

인

인도어, input 방법, 81
인쇄 필터, 177

일

일본어
iconv 모듈, 80
가타카나, 31
간지, 31
글꼴, 78-79
로캘, 77
문자 세트, 77-78
문자 지원, 113
입력 모드, 143
입력 방법, 79
전체 지역화 패키지, 80
지역화, 77-80
히라가나, 31
일본어 키보드, 68
일조 절약 시간(DST), 27

입

입력 모드
en_US.UTF-8 로캘, 114
간체 한자, 144
그리스어, 134, 135
아랍어, 132
영어, 117
유니코드 16진법, 145
일본어, 143
정체 한자, 144
키릴 문자, 133
테이블 조회, 146

입력 모드 (계속)

- 한국어, 144
- 히브리어, 142

입력 방법

- ATOK 방법, 79
- 인도어, 85
- 일본어, 79
- 태국어, 110

전

- 전체 지역화 패키지, 일본어, 80

정

- 정적 링크 만들기, 39
- 정체 한자
 - 문자 지원, 113
 - 입력 모드, 144
- 정체 한자 키보드, 71

중

- 중국어
 - bopomofo, 33
 - Hanzi, 32
 - pinyin, 33
 - zhuyin, 33
 - 대만, 32
 - 입력 방법, 20
 - 중화인민공화국, 32
 - 홍콩, 32
- 중화인민공화국, 33

지

- 지역화, 77-110
 - 구성 파일, 180
 - 정의, 21-23

체

- 체코어, 문자 지원, 113

코

- 코드 독립(CSI), 프로세스 코드 형식, 38
- 코드 변환 모듈, 195-216, 217-219
- 코드 세트, 23
 - 변환, 150
- 코드 세트 독립(CSI)
 - ASCII 슬래시, 36
 - Java 국제화, 36
 - Shift-JIS 코드 세트, 35-37
 - 널 바이트, 36
 - 동적으로 링크된 응용 프로그램, 39
 - 라이브러리, 37
 - 로컬 데이터베이스, 37-38
 - 멀티바이트 문자, 38
 - 명령, 36-37
 - 파일 코드 코드화, 36
 - 확장 UNIX 코드(EUC), 35-37

키

- 키릴 문자, 입력 모드, 133
- 키릴어 키보드, 66
- 키보드
 - Bengali, 82
 - CTL 선택, 175
 - Devanagari, 82
 - Gujarati, 82
 - Gurmukhi, 83
 - Kannada, 83
 - Kedmanee, 108
 - Kedmanee 향상, 109
 - Malayalam, 84
 - Pattajoti, 109
 - SPARC, 66
 - Tamil, 84
 - Teluga, 84
 - type 4, 5 및 5c, 63
 - 그리스 UNIX, 134
 - 그리스 유럽, 134
 - 네델란드(네델란드어), 69
 - 노르웨이어, 69
 - 덴마크어, 67
 - 독일어, 68
 - 레이아웃, 66, 82, 108
 - 레이아웃 변경, 63
 - 미국, 73
 - 미국/UNIX, 73

키보드 (계속)

- 벨기에, 66
- 스웨덴어, 70
- 스위스(독일어), 71
- 스위스(프랑스어), 71
- 스페인어, 70
- 아랍어, 133
- 아어, 66
- 영국, 72
- 유형 6, 62
- 이탈리아어, 68
- 일본어, 68, 143
- 정체 한자, 71
- 지역, 61
- 체코로 레이아웃 변경, 65
- 키릴 문자(러시아), 133
- 키릴어, 66
- 터키어 F, 72
- 터키어 Q, 72
- 포르투갈어, 70
- 프랑스어, 67
- 핀란드어, 67
- 한국어, 69, 144
- 히브리어, 143

태

- 태국어, 32
 - 문자 시퀀스 검사, 108-110
 - 문자 지원, 113
 - 입력 방법, 108, 110
 - 키보드 레이아웃, 20

터

- 터키어, 문자 지원, 113
- 터키어 F 키보드, 72
- 터키어 Q 키보드, 72

테

- 테이블 조회, 입력 모드, 146

페

- 페이지 크기, 일반 크기, 34

포

- 포르투갈어 키보드, 70
- 포스트스크립트
 - 런타임 변수, 188
 - 프롤로그 파일, 186

폴

- 폴란드어, 문자 지원, 113

프

- 프랑스어 키보드, 67
- 프롤로그 파일, 186

핀

- 핀란드어 키보드, 67

한

- 한국어
 - 문자 지원, 113
 - 입력 모드, 144
 - 한글, 32
 - 한자, 32
- 한국어 키보드, 69
- 한글, 32
- 한자, 32

형

- 헝가리어, 문자 지원, 113

히

- 히라가나, 31

히브리, 문자 지원, 113

히브리어

Yiddish, 33

입력 모드, 142

힌

힌디어

Devanagari, 33

문자 지원, 113

