X Locale Database Specification

Yoshio Horiuchi, IBM Japan

X Locale Database Specification

by Yoshio Horiuchi

Copyright © 1994 IBM Corporation
Copyright © 1994 X Consortium

License to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of IBM not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

IBM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS, IN NO EVENT SHALL IBM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from X Consortium.

X Window System is a trademark of The Open Group.

Table of Contents

Chapter 1. LocaleDB

General

An X Locale Database contains the subset of a user's environment that depends on language, in X Window System. It is made up from one or more categories. Each category consists of some classes and sub-classes.

It is provided as a plain ASCII text file, so a user can change its contents easily. It allows a user to customize the behavior of internationalized portion of Xlib without changing Xlib itself.

This document describes;

- Database Format Definition
- Contents of Database in sample implementation

Since it is hard to define the set of required information for all platforms, only the flexible database format is defined. The available entries in database are implementation dependent.

Database Format Definition

The X Locale Database contains one or more category definitions. This section describes the format of each category definition.

The category definition consists of one or more class definitions. Each class definition has a pair of class name and class value, or has several subclasses which are enclosed by the left brace ({) and the right brace (}).

Comments can be placed by using the number sign character (#). Putting the number sign character on the top of the line indicates that the entire line is comment. Also, putting any whitespace character followed by the number sign character indicates that a part of the line (from the number sign to the end of the line) is comment. A line can be continued by placing backslash (\) character as the last character on the line; this continuation character will be discarded from the input. Comment lines cannot be continued on a subsequent line using an escaped new line character.

X Locale Database only accepts XPCS, the X Portable Character Set. The reserved symbols are; the quotation mark("), the number sign (#), the semicolon(;), the backslash(\), the left brace({) and the right brace(}).

The format of category definition is;

```
CategoryDefinition     ::=     CategoryHeader CategorySpec CategoryTrailer
CategoryHeader         ::=     CategoryName NL
CategorySpec           ::=     { ClassSpec }
CategoryTrailer        ::=     "END" Delimiter CategoryName NL
CategoryName           ::=     String
```

ClassSpec ::= ClassName Delimiter ClassValue NL
ClassName ::= String
ClassValue ::= ValueList | "{" NL { ClassSpec } "}"
ValueList ::= Value | Value ";" ValueList
Value ::= ValuePiece | ValuePiece Value
ValuePiece ::= String | QuotedString | NumericString
String ::= Char { Char }
QuotedString ::= """ QuotedChar { QuotedChar } """
NumericString ::= "\\o" OctDigit { OctDigit }
| "\\d" DecDigit { DecDigit }
| "\\x" HexDigit { HexDigit }
Char ::= <XPCS except NL, Space or unescaped reserved symbols>
QuotedChar ::= <XPCS except unescaped """>
OctDigit ::= <character in the range of "0" - "7">
DecDigit ::= <character in the range of "0" - "9">
HexDigit ::= <character in the range of "0" - "9", "a" - "f", "A" - "F">
Delimiter ::= Space { Space }
Space ::= <space> | <horizontal tab>
NL ::= <newline>

Elements separated by vertical bar (|) are alternatives. Curly braces ({...}) indicate zero or more repetitions of the enclosed elements. Square brackets ([...]) indicate that the enclosed element is optional. Quotes ("...") are used around literal characters.

The backslash, which is not the top character of the NumericString, is recognized as an escape character, so that the next one character is treated as a literal character. For example, the two-character sequence, ""\"""(the backslash followed by the quotation mark) is recognized and replaced with a quotation mark character. Any whitespace character, that is not the Delimiter, unquoted and unescaped, is ignored.

Contents of Database

The available categories and classes depend on implementation, because different platform will require different information set. For example, some platform have system locale but some platform don't. Furthermore, there might be a difference in functionality even if the platform has system locale.

In current sample implementation, categories listed below are available.

XLC_FONTSET:XFontSet relative information
XLC_XLOCALE:Character classification and conversion information

XLC_FONTSET Category

The XLC_FONTSET category defines the XFontSet relative information. It contains the CHARSET_REGISTRY-CHARSET_ENCODING name and character mapping side (GL, GR, etc), and is used in Output Method (OM).

class	super class	description
fsN		Nth fontset (N=0,1,2, ...)
charset	fsN	list of encoding name
font	fsN	list of font encoding name

fsN	Includes an encoding information for Nth charset, where N is the index number (0,1,2,...). If there are 4 charsets available in current locale, 4 fontsets, fs0, fs1, fs2 and fs3, should be defined. This class has two subclasses, 'charset' and 'font'.
charset	Specifies an encoding information to be used internally in Xlib for this fontset. The format of value is;

EncodingInfo	::=	EncodingName [":" EncodingSide]
EncodingName	::=	CHARSET_REGISTRY-CHARSET_ENCODING
EncodingSide	::=	"GL" | "GR"

For detail definition of CHARSET_REGISTRY-CHARSET_ENCODING, refer "X Logical Font Descriptions" document.

example:
    ISO8859-1:GL

font	Specifies a list of encoding information which is used for searching appropriate font for this fontset. The left most entry has highest priority.

XLC_XLOCALE Category

The XLC_XLOCALE category defines character classification, conversion and other character attributes.

class	super class	description
encoding_name		codeset name
mb_cur_max		MB_CUR_MAX
state_depend_encoding		state dependent or not
wc_encoding_mask		for parsing wc string
wc_shift_bits		for conversion between wc and mb
csN		Nth charset (N=0,1,2,...)

class	super class	description
side	csN	mapping side (GL, etc)
length	csN	length of a character
mb_encoding	csN	for parsing mb string
wc_encoding	csN	for parsing wc string
ct_encoding	csN	list of encoding name for ct

encoding_name	Specifies a codeset name of current locale.
mb_cur_max	Specifies a maximum allowable number of bytes in a multi-byte character. It is corresponding to MB_CUR_MAX of "ISO/IEC 9899:1990 C Language Standard".
state_depend_encoding	Indicates a current locale is state dependent. The value should be specified "True" or "False".
wc_encoding_mask	Specifies a bit-mask for parsing wide-char string. Each wide character is applied bit-and operation with this bit-mask, then is classified into the unique charset, by using 'wc_encoding'.
wc_shift_bits	Specifies a number of bit to be shifted for converting from a multi-byte character to a wide character, and vice-versa.
csN	Includes a character set information for Nth charset, where N is the index number (0,1,2,...). If there are 4 charsets available in current locale, cs0, cs1, cs2 and cs3 should be defined. This class has five subclasses, 'side', 'length', 'mb_encoding' 'wc_encoding' and 'ct_encoding'.
side	Specifies a mapping side of this charset. The format of this value is; Side ::= EncodingSide[":Default"] The suffix ":Default" can be specified. It indicates that a character belongs to the specified side is mapped to this charset in initial state.
length	Specifies a number of bytes of a multi-byte character of this charset. It should not contain the length of any single-shift sequence.
mb_encoding	Specifies a list of shift sequence for parsing multi-byte string. The format of this value is;

```
MBEncoding    ::=  ShiftType ShiftSequence
               |   ShiftType ShiftSequence ";"
                   MBEncoding
```

ShiftType ::= "<SS>"|"<LSL>"|"<LSR>"
ShiftSequence ::= SequenceValue|SequenceValue ShiftSequence
SequenceValue ::= NumericString

example:
<LSL> \x1b \x28 \x4a; <LSL> \x1b \x28 \x42

wc_encoding	Specifies an integer value for parsing wide-char string. It is used to determine the charset for each wide character, after applying bit-and operation using 'wc_encoding_mask'. This value should be unique in all csN classes.
ct_encoding	Specifies a list of encoding information that can be used for Compound Text.

Sample of X Locale Database

The following is sample X Locale Database file.

```
#  XLocale Database Sample for ja_JP.euc
#

#
#      XLC_FONTSET category
#
XLC_FONTSET
#      fs0 class (7 bit ASCII)
fs0     {
        charset             ISO8859-1:GL
        font                ISO8859-1:GL; JISX0201.1976-0:GL
}
#      fs1 class (Kanji)
fs1     {
        charset             JISX0208.1983-0:GL
        font                JISX0208.1983-0:GL
}
#      fs2 class (Half Kana)
fs2     {
        charset             JISX0201.1976-0:GR
        font                JISX0201.1976-0:GR
}
#      fs3 class (User Defined Character)
# fs3     {
#        charset            JISX0212.1990-0:GL
#        font               JISX0212.1990-0:GL
# }
END XLC_FONTSET
```

```
#
#       XLC_XLOCALE category
#
XLC_XLOCALE

encoding_name              ja.euc
mb_cur_max                 3
state_depend_encoding      False

wc_encoding_mask           \x00008080
wc_shift_bits              8

#       cs0 class
cs0      {
        side                   GL:Default
        length                 1
        wc_encoding            \x00000000
        ct_encoding            ISO8859-1:GL; JISX0201.1976-0:GL
}
#       cs1 class
cs1      {
        side                   GR:Default
        length                 2

        wc_encoding            \x00008080

        ct_encoding            JISX0208.1983-0:GL; JISX0208.1983-0:GR;\
                               JISX0208.1983-1:GL; JISX0208.1983-1:GR
}

#       cs2 class
cs2      {
        side                   GR
        length                 1
        mb_encoding            <SS> \x8e

        wc_encoding            \x00000080

        ct_encoding            JISX0201.1976-0:GR
}

#       cs3 class
# cs3      {
#         side                 GL
#         length               2
#         mb_encoding          <SS> \x8f
# #if HasWChar32
#         wc_encoding          \x20000000
# #else
#         wc_encoding          \x00008000
# #endif
#         ct_encoding          JISX0212.1990-0:GL; JISX0212.1990-0:GR
# }
```

```
END XLC_XLOCALE
```

Reference

[1] *ISO/IEC 9899:1990 C Language Standard*

[2] *X Logical Font Descriptions*