Document number: SC22/WG14 N665
(J11 97-028)

Title: POSIX Alignment

Author: Keld Simonsen
Author affiliation: DKUUG
Postal address: Fruebjergvej 3, DK-2100 København Ø
Email address: keld@dkuug.dk
Telephone number: +45 3917-9944
Fax number: +45 3325-6543

Sponsor: DS
Date: 1997-02-14

Proposal category:
    __ Editorial change/non-normative contribution
    XX Correction
    XX New feature
    __ Addition to obsolescent feature list
    __ Other (please specify)

Area of standard affected:
    XX Environment
    XX Language
    __ Preprocessor
    XX Library
       XX Macro/typedef/tag name
       XX Function
       XX Header
    __ Other (please specify)

Prior art: ISO/IEC 9945 POSIX standards
Target audience: general

Related documents: N431 (Rationale and analysis), N507, N538, N586, N658
Proposal attached: proposal paper
Review committee: Keld Simonsen, Rex Jaeschke, Doug Gwyn, Frank Farance, John Nestor

Abstract:
The paper gives proposals for alignment of C9X with the POSIX standards wrt internationalization features.

Introduction

This paper gives proposals for changes to the C standard to align it with the POSIX standards POSIX System API (C language) (POSIX-1), and ISO/IEC 9945-2:1993 POSIX Shell and Utilities (POSIX-2). It does not cover newer proposals for POSIX or other related specifications, that are not yet international standards.

It builds on the document N431, which gave an overview of internationalisation in C and POSIX standards, a comparison of the functionality and features provided, and also mentioned other incompatibilities between C and POSIX standards. Thus N431 gave the background and rationale for the proposed changes, and it was decided in the Copenhagen meeting to do further work based on N431. The paper here descibes in detail what the changes should be.

Frank Farance has taken an action item on a proposal that addresses non-

internationalization oriented alignment with POSIX.1, such as described in clause 8 of ISO/IEC 9945-1.

Internationalization may be abbreviated as I18N in the following.

The following section numbers refer to the C9X Draft 8.

Changes to N586

Changes to the N586 document are:

LC_MESSAGES description with yesexpr and noexpr deleted
Reserving a "std/" namespace for setlocale() deleted
Deleted numerical thousands and decimal seperator being more than one character.

Changes from N658 (1997-02-10)

Changes to the N658 document are:
localedef MAXINT==-1 clause deleted
POSIX.1 caluse 8 alignment note deleted
some minor text added as discussed 1997-02-13
Class table added
description of lconv struct entries for time added

7.3.1 Character testing functions

POSIX-2 adds in its section 2.5.2 a class "blank", consisting initially of the characters <space> and <tab>. This character class should be added, possibly by adding a function isblank() that is similar to the isspace() function except that the test is for a standard blank character, and the characters covered initially only are space (' ') and horizontal tab ('\t'). Similary a function iswblank() should be added. Text for 7.3.1.12:

"7.3.1.12 the isblank function

Synopsis

[1]

```
#include <ctype.h>
int isblank(int c);
```

Description:

[2] The isblank function tests for any character that is a standard or locale-specific blank character, and for which isalnum is false. The purpose of the function is for testing for blank characters in a line. The standard blank characters are the following: space (' '), and horizontal tab ('\t').

Table 1: Valid Character Class Combinations

In Class	Can also belong to upper	lower	alpha	digit	space	cntrl	punct	graph	print	xdigit	blank
upper	–	–	A	x	x	x	x	A	A	–	x
lower	–	–	A	x	x	x	x	A	A	–	x
alpha	–	–	–	x	x	x	x	A	A	–	x
digit	x	x	x	–	x	x	x	A	A	A	x
space	x	x	x	x	–	–	*	*	*	x	–
cntrl	x	x	x	x	–	–	x	x	x	x	–
punct	x	x	x	x	–	x	–	A	A	x	–
graph	–	–	–	–	–	x	–	–	A	–	–
print	–	–	–	–	–	x	–	–	–	–	–
xdigit	–	–	–	–	x	x	x	A	A	–	x
blank	x	x	x	x	A	–	*	*	*	x	–

NOTES:
Note 1: Explanation of codes:
A Automatically included; see text
– Permitted
x Mutually exclusive
* See note 2

Note 2: The <space> character, which is part of the space and blank class, cannot belong to punct or graph, but automatically shall belong to the print class. Other space or blank characters can be classified as punct, graph, and/or print.

7.3.2 Character case mapping functions

C has only an implicit statement on locale dependence for the case mapping functions, referring to isupper/islower. The locale dependence can be made explicit by adding: "as specified by the current locale" to both the toupper() and tolower() descriptions, so it reads (for tolower):

"If the argument is a character for which isupper is true and there is a corresponding character as specified by the current locale for which islower is true, the tolower function returns the corresponding character; otherwise, the argument is returned unchanged."

7.4 Localization

The POSIX-2 standard was approved after adoption of the C standard, and it contains a format for specifying locales and accompanying charmaps. This is a valuable and

standardized way of specifying locales, on the other hand many C compilers do not operate under a POSIX operating system. It is proposed to add in 7.4 after the macro (LC_ALL etc) section in a non-normative note:

"Footnote:
POSIX-2 specifies locale and charmap formats that may be used to specify locales for C."

A reference to the POSIX-2 standard should be added to the informative bibliography.

7.4.2.1 p_sign_posn and n_sign_posn

POSIX has added a 5th value and thus it is proposed to add:

"5 A space separates the symbol and the sign string, if adjacent."

A table giving the formats for the combinations of *_sign_posn needs to be present, this is not given here yet. ?????

7.5.2.1 int_curr_symbol different from currency_symbol

As there may be differences between the order of how local currency is written and how international currency is written, it is proposed to add the 6 following members of the lconv struct:

int_p_cs_precedes
int_p_sep_by_space
int_n_cs_precedes
int_n_sep_by_space
int_p_sign_pos
int_n_sign_pos

with equivalent wording as "p_cs_precedes" etc, where "currency_symbol" is replaced with "int_curr_symbol" in 7.5.2.1[3].

If any of these values are not present, the value is taken from the corresponding currency_symbol variable.

In section 7.5.2.1 the examples need to be enhanced.

7.5.2.1 time lconv variables

In addition to the defined entries, define the following:

abday Define the abbreviated weekday names, referenced by the %a field descriptor. The operand shall consist of an array of seven strings. The first string shall be the abbreviated name of the first day of the week (Sunday), the second the abbreviated name of the second day, and so on. The value for the C locale is:

["Sun","Mon","Tue","Wed","Thu","Fri","Sat"].

day Define the full weekday names, referenced by the %A field descriptor. The operand shall consist of an array of seven strings. The first string shall be the full name of the first day of the week (Sunday), the second the full name of the second day, and so on. The value for the C locale is:
["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"]

abmon Define the abbreviated month names, referenced by the %b field descriptor. The operand shall consist of an array of twelve strings. The first string shall be the abbreviated name of the first month of the year (January), the second the abbreviated name of the second month, and so on. The value for the C locale is:
["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"]

mon Define the full month names, referenced by the %B field descriptor. The operand shall consist of an array of twelve strings. The first string shall be the full name of the first month of the year (January), the second the full name of the second month, and so on. The value for the C locale is:
["January","February","March","April","May","June","July","August","Septembeer","October","November","December"]

d_t_fmt Define the appropriate date and time representation, referenced by the %c field descriptor. The operand shall consist of a string, and can contain any combination of characters and field descriptors. In addition, the string can contain escape sequences defined in Table 2. The value for the C locale is: "...".

d_fmt Define the appropriate date representation, referenced by the %x field descriptor. The operand shall consist of a string, and can contain any combination of characters and field descriptors. In addition, the string can contain escape sequences defined in Table 2. The value for the C locale is: "".

t_fmt Define the appropriate time representation, referenced by the %X field descriptor. The operand shall consist of a string, and can contain any combination of characters and field descriptors. In addition, the string can contain escape sequences defined in Table 2. The value for the C locale is "%T".

am_pm Define the appropriate representation of the ante meridiem and post meridiem strings, referenced by the %p field descriptor. The operand shall consist of an array of two strings. The first string shall represent the antemeridiem designation, the last string the postmeridiem designation. The value for the C locale is ["",""].

t_fmt_ampm Define the appropriate time representation in the 12-hour clock format with am_pm, referenced by the %r field descriptor. The operand shall consist of a string and can contain any combination of characters and field descriptors. If the string is empty, the 12-hour format is not supported in the FDCC-set. The value for the C locale is " %T".

It is implementation defined whether the following optional keywords shall be recognized. If they are not supported, but present in a description file, they shall be ignored.

era Shall be used to define alternate Eras, corresponding to the %E field descriptor modifier. The format of the operand is unspecified, but shall support the definition

of the %EC and %Ey field descriptors, and may also define the era_year format (%EY).

era_year Shall be used to define the format of the year in alternate Era format, corresponding to the %EY field descriptor.

era_d_fmt Shall be used to define the format of the date in alternate Era notation, corresponding to the %Ex field descriptor.

alt_digits Shall be used to define alternate symbols for digits, corresponding to the %O field descriptor modifier. The operand shall consist of semicolon-separated strings. The first string shall be the alternate symbol corresponding with zero, the second string the symbol corresponding with one, and so on. Up to 100 alternate symbol strings can be specified. The %O modifier indicates that the string corresponding to the value specified via the field descriptor shall be used instead of the value.

7.14.3.5 strftime

The date utility in POSIX-2 4.15 has all of the formats of C's strftime() plus more, all of which are proposed to be added to strftime:

%C is replaced by the century (a year divided by 100 and truncated to an integer) as a decimal number (00-99)

%D is replaced by the date in the format mm/dd/yy

%e is replaced by the day of the month as a decimal number (1-31 in a two-digit field with leading <space> fill)

%F is replaced by the date in the format YYYY-MM-DD (ISO 8601 format)

%h a synonym for %b

%n is replaced by a <newline> character

%r is replaced by the 12 h clock time (01-12) using the AM/PM notation; in the "C" locale, this shall be equivalent to "%I:%M:%S %p"

%t is replaced by a <tab> character

%T is replaced by the 24 h clock time (00-23) in the format HH:MM:SS.

%u is replaced by the week of the year (Sunday as the first day of the week) as a decimal number (00-53). All days in a new year preceding the first Sunday shall be considered to be in week 0.

%V is replaced by the week of the year (Monday as the first day of the week) as a decimal number (00-53). The method for determining the week number shall be as specified in ISO 8601.

A number of modified field descriptors %O<d> and %E<d> are also defined (4.15.4.2). Some field descriptors can be modified by the E and O modifier characters to indicate a different format or specification as specified in the LC_TIME locale description. If the corresponding data (see era, era_year, era_d_fmt, and alt_digits) is not specified or not supported for the current locale, the unmodified field descriptor value shall be used.

Kelds note: I cannot see where era, era_year era_d_fmt and alt_digits are defined, as well as the normal date and time locale stuff. Should these not be added to 7.5[2] and 7.5.2.1[3]?

%Ec Locale's alternate date and time representation.
%EC The name of the base year (period) in the locale's alternate representation.
%Ex Locale's alternate date representation.
%Ey Offset from %EC (year only) in the locale's alternate representation.
%EY Full alternate year representation.
%Od Day of month using the locale's alternate numeric symbols.
%Oe Day of month using the locale's alternate numeric symbols.
%OH Hour (24-hour clock) using the locale's alternate numeric symbols.
%OI Hour (12-hour clock) using the locale's alternate numeric symbols.
%Om Month using the locale's alternate numeric symbols.
%OM Minutes using the locale's alternate numeric symbols.
%OS Seconds using the locale's alternate numeric symbols.
%OU Week number of the year (Sunday as the first day of the week) using the locale's alternate numeric symbols.
%Ow Weekday as number in the locale's alternate representation (Sunday=0).
%OW Week number of the year (Monday as the first day of the week) using the locale's alternate numeric symbols.
%Oy Year (offset from %C) in alternate representation.

Differences from POSIX

This proposal introduces the following changes from POSIX (all additions):

Adds to lconv struct:

int_p_cs_precedes
int_p-sep_by_space
int_n_cs_precedes
int_n_sep_by_space

Adds %F for ISO 8601 format in strftime()

Other specifications

An amendment (.2b) to POSIX-2 is currently underway (currently out for CD registration ballot) providing further specifications in the i18n area. This may be relevant to C functionality.

WG20 is working on an specification standards for cultural conventions, which has POSIX and C downwards compatible locale and charmap functionality. The extensions should be relevant for the C standard. This is currently WD stage, expected to go to CD registration Oct 1996.

WG21 is specifying a number of i18n and character functionality in their C++ standard. This is currently at CD stage.

It is proposed to watch these activities closely and align where possible.

<END OF DOCUMENT>