IEEE P1003.2 Draft 11.2 – September 1991

Copyright © 1991 by the
Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017, USA
All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change. The publication, distribution, or copying of this draft, as well as all derivative works based on this draft, is expressly prohibited except as set forth below.

Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only, and subject to the restrictions contained herein.

Permission is hereby also granted for member bodies and technical committees of ISO and IEC to reproduce this document for purposes of developing a national position, subject to the restrictions contained herein.

Permission is hereby also granted to the preceding entities to make limited copies of this document in an electronic form only for the stated activities.

The following restrictions apply to reproducing or transmitting the document in any form: 1) all copies or portions thereof must identify the document's IEEE project number and draft number, and must be accompanied by this entire notice in a prominent location; 2) no portion of this document may be redistributed in any modified or abridged form without the prior approval of the IEEE Standards Department.

Other entities seeking permission to reproduce this document, or any portion thereof, for standardization or other activities, must contact the IEEE Standards Department for the appropriate license.

Use of information contained in this unapproved draft is at your own risk.

IEEE Standards Department
Copyright and Permissions
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA
+1 (908) 562-3800
+1 (908) 562-1571 [FAX]

Based on the questionable value in a shell script of getting an entire array of characters back, and the problem of returning a collation description that makes sense, short of a complete `localedef` source, the output from requests for categories LC_CTYPE and LC_COLLATE has been made implementation defined.

The `-m` option has been added to allow applications to query for the existence of charmaps. The output is a list of the charmaps (implementation-supplied and user-supplied, if any) on the system.

The `-c` option was included for readability when more than one category is 2
selected (e.g., via more than one keyword name or via a category name). It is 2
valid both with and without the `-k` option. 2

The `charmap` keyword, which returns the name of the charmap (if any) that was used when the current locale was created, was introduced to allow applications needing the information to retrieve it.

4.35 `localedef` — Define locale environment

4.35.1 Synopsis

`localedef [-c][-f` *charmap* `][-i` *sourcefile* `]` *name*

4.35.2 Description

The `localedef` utility shall convert source definitions for locale categories into a format usable by the functions and utilities whose operational behavior is determined by the setting of the locale environment variables defined in 2.5. It is implementation defined whether users shall have the capability to create new locales, in addition to those supplied by the implementation. If the symbolic constant {POSIX 2_LOCALEDEF } is defined, then the system supports the creation of new locales. In a system not supporting this capability, the `localedef` utility shall terminate with an exit code of 3.

The utility shall read source definitions for one or more locale categories belonging to the same locale from the file named in the `-i` option (if specified) or from standard input.

The *name* operand identifies the target locale. The utility shall support the creation of *public* , or generally accessible locales, as well as *private* , or restricted-access locales. Implementations may restrict the capability to create or modify public locales to users with the appropriate privileges.

Each category source definition shall be identified by the corresponding environment variable name and terminated by an `END` *category-name* statement. The following categories shall be supported. In addition, the input may contain source for implementation-defined categories.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

LC_CTYPE	Defines character classification and case conversion.
LC_COLLATE	Defines collation rules.
LC_MONETARY	Defines the format and symbols used in formatting of monetary information.
LC_NUMERIC	Defines the decimal delimiter, grouping, and grouping symbol for nonmonetary numeric editing.
LC_TIME	Defines the format and content of date and time information.
LC_MESSAGES	Defines the format and values of affirmative and negative responses.

4.35.3 Options

The `localedef` utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following options shall be supported by the implementation:

`−c` Create permanent output even if warning messages have been issued.

`−f` *charmap*
Specify the pathname of a file containing a mapping of character symbols and collating element symbols to actual character encodings. The format of the *charmap* is described under 2.4.1. This option shall be specified if symbolic names (other than collating symbols defined in a `collating-symbol` keyword) are used. If the `−f` option is not present, an implementation-defined default character mapping file shall be used. 2

`−i` *inputfile* The pathname of a file containing the source definitions. If this option is not present, source definitions shall be read from standard input. The format of the *inputfile* is described in 2.5.2.

4.35.4 Operands

The following operand shall be supported by the implementation:

name Identifies the locale. See 2.5 for a description of the use of this name. If the name contains one or more slash characters, *name* shall be interpreted as a pathname where the created locale definition(s) shall be stored. If *name* does not contain any slash characters, the interpretation of the name is implementation defined and the locale shall be public. This capability may be restricted to users with appropriate privileges.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

4.35.5 External Influences

4.35.5.1 Standard Input

Unless the `-i` option is specified, the standard input shall be a text file containing
one or more locale category source definitions, as described in 2.5.2. When lines 1
are continued using the escape character mechanism, there is no limit to the 1
length of the accumulated continued line. 1

4.35.5.2 Input Files

The character set mapping file specified as the *charmap* option-argument is described under 2.4.1. If a locale category source definition contains a `copy` statement, as defined in 2.5.2, and the `copy` statement names a valid, existing locale, then `localedef` shall behave as if the source definition had contained a valid category source definition for the named locale.

4.35.5.3 Environment Variables

The following environment variables shall affect the execution of `localedef`:

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

LC_ALL This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**. and **LC_**∗ variables as described in 2.6.

LC_COLLATE (This variable shall have no affect on `localedef`; the POSIX Locale shall be used for this category.)

LC_CTYPE This variable shall determine the locale for the interpretation of sequences of bytes of argument data as characters (e.g., single- versus multibyte characters). This variable shall have no affect on the processing of `localedef` input data; the POSIX Locale shall be used for this purpose, regardless of the value of this variable.

LC_MESSAGES This variable shall determine the language in which messages should be written.

4.35.5.4 Asynchronous Events

Default.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

4.35.6 External Effects

4.35.6.1 Standard Output

The utility shall report all categories successfully processed, in an unspecified format.

4.35.6.2 Standard Error

Used only for diagnostic messages.

4.35.6.3 Output Files

The format of the created output is unspecified. If the *name* operand does not contain a slash, the existence of an output file for the locale is unspecified.

4.35.7 Extended Description

None.

4.35.8 Exit Status

The `localedef` utility shall exit with one of the following values:

- 0 No errors occurred and the locale(s) were successfully created.
- 1 Warnings occurred and the locale(s) were successfully created.
- 2 The locale specification exceeded implementation limits or the coded character set or sets used were not supported by the implementation, and no locale was created.
- 3 The capability to create new locales is not supported by the implementation.
- >3 Warnings or errors occurred and no output was created.

4.35.9 Consequences of Errors

If an error is detected, no permanent output shall be created.

If warnings occur, permanent output shall be created if the `-c` option was specified. The following conditions shall cause warning messages to be issued:

— If a symbolic name not found in the *charmap* file is used for the descriptions of the LC_CTYPE or LC_COLLATE categories (for other categories, this shall be an error conditions).

— If the number of operands to the `order` keyword exceeds the {COLL_WEIGHTS_MAX} limit.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

— If optional keywords not supported by the implementation are present in the source. 1

Other implementation-defined conditions may also cause warnings.

4.35.10 Rationale. *(This subclause is not a part of P1003.2)*

Usage, Examples

The output produced by the `localedef` utility is implementation defined. The *name* operand is used to identify the specific locale. (As a consequence, although several categories can be processed in one execution, only categories belonging to the same locale can be processed.)

The *charmap* definition is optional, and is contained outside the locale definition. This allows both completely ''self-defined'' source files, and ''generic'' sources (applicable to more than one code set). To aid portability, all *charmap* definitions shall use the same symbolic names for the portable character set. As explained in 2.4.1, it is implementation defined whether or not users or applications can provide additional character set description files. Therefore, the `−f` option might be operable only when an implementation-provided *charmap* is named.

History of Decisions Made

This description is based on work performed in the UniForum Technical Committee Subcommittee on Internationalization.

The `localedef` utility is provided as a standard, portable interface for implementations that allow users to create new locales, in addition to implementation-supplied ones.

The ability to create new locales and categories, already available on many commercially available implementations of POSIX compliant systems, provides the means by which application providers can develop portable applications which use standard interfaces to adjust the behavior of the application to language and culture differences.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

4.36 `logger` — Log messages

4.36.1 Synopsis

`logger` *string* . . .

4.36.2 Description

The `logger` utility saves a message, in an unspecified manner and format, containing the *string* operands provided by the user. The messages are expected to be evaluated later by personnel performing system administration tasks.

4.36.3 Options

None.

4.36.4 Operands

The following operands shall be supported by the implementation:

string One of the string arguments whose contents are concatenated together, in the order specified, separated by single `<space>`s.

4.36.5 External Influences

4.36.5.1 Standard Input

None.

4.36.5.2 Input Files

None.

4.36.5.3 Environment Variables

The following environment variables shall affect the execution of `logger`:

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

LC_ALL This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.