NAME

locale – get locale-specific information

SYNOPSIS

locale [–**a** | –**m**]
locale [–**ck**] *name* . . .

AVAILABILITY

SUNWloc

DESCRIPTION

The **locale** utility writes information about the current locale environment, or all public locales, to the standard output. For the purposes of this section, a *public locale* is one provided by the implementation that is accessible to the application.

When **locale** is invoked without any arguments, it summarizes the current locale environment for each locale category as determined by the settings of the environment variables.

When invoked with operands, it writes values that have been assigned to the keywords in the locale categories, as follows:

- Specifying a keyword name selects the named keyword and the category containing that keyword.
- Specifying a category name selects the named category and all keywords in that category.

OPTIONS

The following options are supported:

–**a** Write information about all available public locales. The available locales include **POSIX**, representing the POSIX locale.

–**c** Write the names of selected locale categories. The –**c** option increases readability when more than one category is selected (for example, via more than one keyword name or via a category name). It is valid both with and without the –**k** option.

–**k** Write the names and values of selected keywords. The implementation may omit values for some keywords; see **OPERANDS**.

–**m** Write names of available charmaps; see **localedef**(1).

OPERANDS

The following operand is supported:

name The name of a locale category, the name of a keyword in a locale category, or the reserved name **charmap**. The named category or keyword will be selected for output. If a single *name* represents both a locale category name and a keyword name in the current locale, the results are unspecified; otherwise, both category and keyword names can be specified as *name* operands, in any sequence.

EXAMPLES

In the following examples, the assumption is that locale environment variables are set as follows:

LANG=locale_x LC_COLLATE=locale_y

The command:

```
locale
```

would result in the following output:

```
LANG=locale_x
LC_CTYPE="locale_x"
LC_NUMERIC="locale_x"
LC_TIME="locale_x"
LC_COLLATE=locale_y
LC_MONETARY="locale_x"
LC_MESSAGES="locale_x"
LC_ALL=
```

The command:

```
LC_ALL=POSIX locale -ck decimal_point
```

would produce:

```
LC_NUMERIC
decimal_point="."
```

The following command shows an application of **locale** to determine whether a user-supplied response is affirmative:

```
if printf "%s\n" "$response" | grep -Eq "$(locale yesexpr)"
then
        affirmative processing goes here
else
        non-affirmative processing goes here
fi
```

ENVIRONMENT

See **environ**(5) for the descriptions of **LANG**, **LC_ALL**, **LC_TYPE**, **LC_MESSAGES**, and **NLSPATH**.

The **LANG**, **LC_**∗, and **NLSPATH** environment variables must specify the current locale environment to be written out; they will be used if the −**a** option is not specified.

EXIT STATUS

The following exit values are returned:

0 All the requested information was found and output successfully.

>0 An error occurred.

SEE ALSO

localedef(1), **charmap**(5), **locale**(5)

NOTES

If **LC_CTYPE** or keywords in the category **LC_CTYPE** are specified, only the values in the codeset 0 are written out.

If **LC_COLLATE** or keywords in the category **LC_COLLATE** are specified, no actual values are written out.