NAME

charmap – character set description file

DESCRIPTION

A character set description file or *charmap* defines characteristics for a coded character set. Other information about the coded character set may also be in the file. Coded character set character values are defined using symbolic character names followed by character encoding values.

The character set description file provides:

- The capability to describe character set attributes (such as collation order or character classes) independent of character set encoding, and using only the characters in the portable character set. This makes it possible to create generic **localedef**(1) source files for all codesets that share the portable character set.
- Standardized symbolic names for all characters in the portable character set, making it possible to refer to any such character regardless of encoding.

Symbolic Names

Each symbolic name is included in the file and is mapped to a unique encoding value (except for those symbolic names that are shown with identical glyphs). If the control characters commonly associated with the symbolic names in the following table are supported by the implementation, the symbolic names and their corresponding encoding values are included in the file. Some of the encodings associated with the symbolic names in this table may be the same as characters in the portable character set table.

<ACK>	<DC2>	<ENQ>	<FS>	<IS4>	<SOH>
<BEL>	<DC3>	<EOT>	<GS>	<LF>	<STX>
<BS>	<DC4>	<ESC>	<HT>	<NAK>	<SUB>
<CAN>	<DEL>	<ETB>	<IS1>	<RS>	<SYN>
<CR>	<DLE>	<ETX>	<IS2>	<SI>	<US>
<DC1>	<EM>	<FF>	<IS3>	<SO>	<VT>

Declarations

The following declarations can precede the character definitions. Each must consist of the symbol shown in the following list, starting in column 1, including the surrounding brackets, followed by one or more blank characters, followed by the value to be assigned to the symbol.

<*code_set_name*> The name of the coded character set for which the character set description file is defined.

<*mb_cur_max*> The maximum number of bytes in a multi-byte character. This defaults to 1.

<*mb_cur_min*> An unsigned positive integer value that defines the minimum number of bytes in a character for the encoded character set.

<*escape_char*> The escape character used to indicate that the characters following will be interpreted in a special way, as defined later in this section. This defaults to backslash (\ ), which is the character glyph used

in all the following text and examples, unless otherwise noted.

<*comment_char*> The character that when placed in column 1 of a charmap line, is used to indicate that the line is to be ignored. The default character is the number sign (#).

Format

The character set mapping definitions will be all the lines immediately following an identifier line containing the string **CHARMAP** starting in column 1, and preceding a trailer line containing the string **END CHARMAP** starting in column 1. Empty lines and lines containing a <*comment_char*> in the first column will be ignored. Each non-comment line of the character set mapping definition (that is, between the **CHARMAP** and **END CHARMAP** lines of the file) must be in either of two forms:

"%s %s %s\n",<*symbolic-name*>,<*encoding*>,<*comments*>

or

"%s...%s %s %s\n",<*symbolic-name*>,<*symbolic-name*>, <*encoding*>,<*comments*>

In the first format, the line in the character set mapping definition defines a single symbolic name and a corresponding encoding. A character following an escape character is interpreted as itself; for example, the sequence <\\\>> represents the symbolic name \> enclosed between angle brackets.

In the second format, the line in the character set mapping definition defines a range of one or more symbolic names. In this form, the symbolic names must consist of zero or more non-numeric characters, followed by an integer formed by one or more decimal digits. The characters preceding the integer must be identical in the two symbolic names, and the integer formed by the digits in the second symbolic name must be equal to or greater than the integer formed by the digits in the first name. This is interpreted as a series of symbolic names formed from the common part and each of the integers between the first and the second integer, inclusive. As an example, **<j0101>...<j0104>** is interpreted as the symbolic names **<j0101>**, **<j0102>**, **<j0103>**, and **<j0104>**, in that order.

A character set mapping definition line must exist for all symbolic names and must define the coded character value that corresponds to the character glyph indicated in the table, or the coded character value that corresponds with the control character symbolic name. If the control characters commonly associated with the symbolic names are supported by the implementation, the symbolic name and the corresponding encoding value must be included in the file. Additional unique symbolic names may be included. A coded character value can be represented by more than one symbolic name.

The encoding part is expressed as one (for single-byte character values) or more concatenated decimal, octal or hexadecimal constants in the following formats:

"%cd%d",<*escape_char*>,<*decimal byte value*>
"%cx%x",<*escape_char*>,<*hexadecimal byte value*>
"%c%o",<*escape_char*>,<*octal byte value*>

Decimal Constants

Decimal constants must be represented by two or three decimal digits, preceded by the escape character and the lower-case letter **d**; for example, **\d05**, **\d97**, or **\d143**. Hexadecimal constants must be represented by two hexadecimal digits, preceded by the escape character and the lower-case letter **x**; for example, **\x05**, **\x61**, or **\x8f**. Octal constants must be represented by two or three octal digits, preceded by the escape character; for example, **\05**, **\141**, or **\217**. In a portable charmap file, each constant must represent an 8-bit byte. Implementations supporting other byte sizes may allow constants to represent values larger than those that can be represented in 8-bit bytes, and to allow additional digits in constants. When constants are concatenated for multi-byte character values, they must be of the same type, and interpreted in byte order from first to last with the least significant byte of the multi-byte character specified by the last constant.

Ranges of Symbolic Names

In lines defining ranges of symbolic names, the encoded value is the value for the first symbolic name in the range (the symbolic name preceding the ellipsis). Subsequent symbolic names defined by the range will have encoding values in increasing order. For example, the line

```
<j0101>...<j0104>         \d129\d254
```

will be interpreted as:

```
<j0101>         \d129\d254
<j0102>         \d129\d255
<j0103>         \d130\d0
<j0104>         \d130\d1
```

Note that this line will be interpreted as the example even on systems with bytes larger than 8 bits. The comment is optional.

SEE ALSO

locale(1), **localedef**(1), **nl_langinfo**(3C), **locale**(5)