

NAME

groff_char – groff glyph names

DESCRIPTION

This manual page lists the standard **groff** glyph names and the default input mapping, latin1. The glyphs in this document look different depending on which output device was chosen (with option **-T** for the [man\(1\)](#) program or the roff formatter). Glyphs not available for the device that is being used to print or view this manual page are marked with ‘(N/A)’; the device currently used is ‘pdf’.

In the actual version, **groff** provides only 8-bit characters for direct input and named entities for further glyphs. On ASCII platforms, input character codes in the range 0 to 127 (decimal) represent the usual 7-bit ASCII characters, while codes between 127 and 255 are interpreted as the corresponding characters in the *latin1* (ISO-8859-1) code set by default. This mapping is contained in the file *latin1.tmac* and can be changed by loading a different input encoding. Note that some of the input characters are reserved by **groff**, either for internal use or for special input purposes. On EBCDIC platforms, only code page *cp1047* is supported (which contains the same characters as latin1; the input encoding file is called *cp1047.tmac*). Again, some input characters are reserved for internal and special purposes.

All roff systems provide the concept of named glyphs. In traditional roff systems, only names of length 2 were used, while groff also provides support for longer names. It is strongly suggested that only named glyphs are used for all character representations outside of the printable 7-bit ASCII range.

Some of the predefined groff escape sequences (with names of length 1) also produce single glyphs; these exist for historical reasons or are printable versions of syntactical characters. They include ‘\’, ‘\’’, ‘\’’, ‘\’’, ‘\’’, ‘\’’, ‘\’’, and ‘\e’; see [groff\(7\)](#).

In groff, all of these different types of characters and glyphs can be tested positively with the ‘.if c’ conditional.

REFERENCE

In this section, the glyphs in groff are specified in tabular form. The meaning of the columns is as follows.

Output shows how the glyph is printed for the current device; although this can have quite a different shape on other devices, it always represents the same glyph.

Input name

specifies how the glyph is input either directly by a key on the keyboard, or by a groff escape sequence.

Input code

applies to glyphs which can be input with a single character, and gives the ISO latin1 decimal code of that input character. Note that this code is equivalent to the lowest 256 Unicode characters, including 7-bit ASCII in the range 0 to 127.

PostScript name

gives the usual PostScript name of the glyph.

Unicode decomposed

is the glyph name used in composite glyph names.

7-bit Character Codes 32-126

These are the basic glyphs having 7-bit ASCII code values assigned. They are identical to the printable characters of the character standards ISO-8859-1 (latin1) and Unicode (range *Basic Latin*). The glyph names used in composite glyph names are ‘u0020’ up to ‘u007E’.

Note that input characters in the range 0–31 and character 127 are *not* printable characters. Most of them are invalid input characters for **groff** anyway, and the valid ones have special meaning. For EBCDIC, the printable characters are in the range 66–255.

48–57 Decimal digits 0 to 9 (print as themselves).

65–90 Upper case letters A–Z (print as themselves).

97–122 Lower case letters a–z (print as themselves).

Most of the remaining characters not in the just described ranges print as themselves; the only exceptions are the following characters:

- ` the ISO latin1 ‘Grave Accent’ (code 96) prints as ‘, a left single quotation mark; the original character can be obtained with ‘\`’.
- ' the ISO latin1 ‘Apostrophe’ (code 39) prints as ', a right single quotation mark; the original character can be obtained with ‘\`a_q’.
- the ISO latin1 ‘Hyphen, Minus Sign’ (code 45) prints as a hyphen; a minus sign can be obtained with ‘\`-’.
- ~ the ISO latin1 ‘Tilde’ (code 126) is reduced in size to be usable as a diacritic; a larger glyph can be obtained with ‘\`t_i’.
- ^ the ISO latin1 ‘Circumflex Accent’ (code 94) is reduced in size to be usable as a diacritic; a larger glyph can be obtained with ‘\`h_a’.

<i>Output</i>	<i>Input name</i>	<i>Input code</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
!	!	33	exclam	u0021	
"	"	34	quotedbl	u0022	
#	#	35	numbersign	u0023	
\$	\$	36	dollar	u0024	
%	%	37	percent	u0025	
&	&	38	ampersand	u0026	
'	'	39	quoteright	u0027	
((40	parenleft	u0028	
))	41	parenright	u0029	
*	*	42	asterisk	u002A	
+	+	43	plus	u002B	
,	,	44	comma	u002C	
-	-	45	hyphen	u2010	
.	.	46	period	u002E	
/	/	47	slash	u002F	
:	:	58	colon	u003A	
;	;	59	semicolon	u003B	
<	<	60	less	u003C	
=	=	61	equal	u003D	
>	>	62	greater	u003E	
?	?	63	question	u003F	
@	@	64	at	u0040	
[[91	bracketleft	u005B	
\	\	92	backslash	u005C	
]]	93	bracketright	u005D	
^	^	94	circumflex	u005E	circumflex accent
_	_	95	underscore	u005F	
`	`	96	quoteleft	u0060	
{	{	123	braceleft	u007B	
		124	bar	u007C	
}	}	125	braceright	u007D	
~	~	126	tilde	u007E	tilde accent

8-bit Character Codes 160 to 255

They are interpreted as printable characters according to the *latin1* (ISO-8859-1) code set, being identical to the Unicode range *Latin-1 Supplement*.

Input characters in range 128-159 (on non-EBCDIC hosts) are not printable characters.

160 the ISO latin1 *no-break space* is mapped to ‘\~’, the stretchable space character.

173 the soft hyphen control character. **groff** never uses this character for output (thus it is omitted in the table below); the input character 173 is mapped onto ‘\%’.

The remaining ranges (161–172, 174–255) are printable characters that print as themselves. Although they can be specified directly with the keyboard on systems with a latin1 code page, it is better to use their glyph names; see next section.

<i>Output</i>	<i>Input name</i>	<i>Input code</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
¡	¡	161	exclamdown	u00A1	inverted exclamation mark
¢	¢	162	cent	u00A2	
£	£	163	sterling	u00A3	
¤	¤	164	currency	u00A4	
¥	¥	165	yen	u00A5	
		166	brokenbar	u00A6	
§	§	167	section	u00A7	
¨	¨	168	dieresis	u00A8	
©	©	169	copyright	u00A9	
ª	ª	170	ordfeminine	u00AA	
«	«	171	guillemotleft	u00AB	
¬	¬	172	logicalnot	u00AC	
®	®	174	registered	u00AE	
–	–	175	macron	u00AF	
°	°	176	degree	u00B0	
±	±	177	plusminus	u00B1	
²	²	178	twosuperior	u00B2	
³	³	179	threesuperior	u00B3	
´	´	180	acute	u00B4	acute accent
µ	µ	181	mu	u00B5	micro sign
¶	¶	182	paragraph	u00B6	
·	·	183	periodcentered	u00B7	
¸	¸	184	cedilla	u00B8	
¹	¹	185	onesuperior	u00B9	
º	º	186	ordmasculine	u00BA	
»	»	187	guillemotright	u00BB	
¼	¼	188	onequarter	u00BC	
½	½	189	onehalf	u00BD	
¾	¾	190	threequarters	u00BE	
¿	¿	191	questiondown	u00BF	
À	À	192	Agrave	u0041_0300	
Á	Á	193	Aacute	u0041_0301	
Â	Â	194	Acircumflex	u0041_0302	
Ã	Ã	195	Atilde	u0041_0303	
Ä	Ä	196	Adieresis	u0041_0308	
Å	Å	197	Aring	u0041_030A	
Æ	Æ	198	AE	u00C6	
Ç	Ç	199	Ccedilla	u0043_0327	
È	È	200	Egrave	u0045_0300	
É	É	201	Eacute	u0045_0301	
Ê	Ê	202	Ecircumflex	u0045_0302	
Ë	Ë	203	Edieresis	u0045_0308	

<i>Output</i>	<i>Input name</i>	<i>Input code</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
Ì	ì	204	Igrave	u0049_0300	
Í	í	205	Iacute	u0049_0301	
Î	î	206	Icircumflex	u0049_0302	
Ï	ï	207	Idieresis	u0049_0308	
Ð	ð	208	Eth	u00D0	
Ñ	ñ	209	Ntilde	u004E_0303	
Ò	ò	210	Ograve	u004F_0300	
Ó	ó	211	Oacute	u004F_0301	
Ô	ô	212	Ocircumflex	u004F_0302	
Õ	õ	213	Otilde	u004F_0303	
Ö	ö	214	Odieresis	u004F_0308	
×	×	215	multiply	u00D7	
Ø	ø	216	Oslash	u00D8	
Ù	ù	217	Ugrave	u0055_0300	
Ú	ú	218	Uacute	u0055_0301	
Û	û	219	Ucircumflex	u0055_0302	
Ü	ü	220	Udieresis	u0055_0308	
Ý	ý	221	Yacute	u0059_0301	
Þ	þ	222	Thorn	u00DE	
ß	ß	223	germandbls	u00DF	
à	à	224	agrave	u0061_0300	
á	á	225	aacute	u0061_0301	
â	â	226	acircumflex	u0061_0302	
ã	ã	227	atilde	u0061_0303	
ä	ä	228	adieresis	u0061_0308	
å	å	229	aring	u0061_030A	
æ	æ	230	ae	u00E6	
ç	ç	231	cedilla	u0063_0327	
è	è	232	egrave	u0065_0300	
é	é	233	eacute	u0065_0301	
ê	ê	234	ecircumflex	u0065_0302	
ë	ë	235	edieresis	u0065_0308	
ì	ì	236	igrave	u0069_0300	
í	í	237	iacute	u0069_0301	
î	î	238	icircumflex	u0069_0302	
ï	ï	239	idieresis	u0069_0308	
ð	ð	240	eth	u00F0	
ñ	ñ	241	ntilde	u006E_0303	
ò	ò	242	ograde	u006F_0300	
ó	ó	243	oacute	u006F_0301	
ô	ô	244	ocircumflex	u006F_0302	
õ	õ	245	otilde	u006F_0303	
ö	ö	246	odieresis	u006F_0308	
÷	÷	247	divide	u00F7	
ø	ø	248	oslash	u00F8	
ù	ù	249	ugrave	u0075_0300	
ú	ú	250	uacute	u0075_0301	
û	û	251	ucircumflex	u0075_0302	
ü	ü	252	udieresis	u0075_0308	
ý	ý	253	yacute	u0079_0301	
þ	þ	254	thorn	u00FE	

ÿ ŷ 255 ydieresis u0079_0308

Named Glyphs

Glyph names can be embedded into the document text by using escape sequences. [groff\(7\)](#) describes how these escape sequences look. Glyph names can consist of quite arbitrary characters from the ASCII or latin1 code set, not only alphanumeric characters. Here some examples:

`\(ch` A glyph having the 2-character name *ch*.

`\[char_name]`

A glyph having the name *char_name* (having length 1, 2, 3, ...). Note that ‘c’ is not the same as ‘\[c]’ (c a single character): The latter is internally mapped to glyph name ‘\c’. By default, groff defines a single glyph name starting with a backslash, namely ‘\’, which can be either accessed as ‘\–’ or ‘\[–]’.

`\[base_glyph composite_1 composite_2 ...]`

A composite glyph; see below for a more detailed description.

In groff, each 8-bit input character can also be referred to by the construct ‘\[char*n*]’ where *n* is the decimal code of the character, a number between 0 and 255 without leading zeros (those entities are *not* glyph names). They are normally mapped onto glyphs using the `.trin` request. Another special convention is the handling of glyphs with names directly derived from a Unicode code point; this is discussed below. Moreover, new glyph names can be created by the `.char` request; see [groff\(7\)](#).

In the following, a plus sign in the ‘Notes’ column indicates that this particular glyph name appears in the PS version of the original troff documentation, CSTR 54.

Entries marked with ‘***’ denote glyphs for mathematical purposes (mainly used for DVI output). Normally, such glyphs have metrics which make them unusable in normal text.

<i>Output</i>	<i>Input name</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
Ð	<code>\[-D]</code>	Eth	u00D0	uppercase eth
ð	<code>\[Sd]</code>	eth	u00F0	lowercase eth
Þ	<code>\[TP]</code>	Thorn	u00DE	uppercase thorn
þ	<code>\[Tp]</code>	thorn	u00FE	lowercase thorn
ß	<code>\[ss]</code>	germandbls	u00DF	German sharp s

Ligatures and Other Latin Glyphs

ff	<code>\[ff]</code>	ff	u0066_0066	ff ligature +
fi	<code>\[fi]</code>	fi	u0066_0069	fi ligature +
fl	<code>\[fl]</code>	fl	u0066_006C	fl ligature +
ffi	<code>\[Fi]</code>	ffi	u0066_0066_0069	ffi ligature +
ffl	<code>\[Fl]</code>	ffl	u0066_0066_006C	ffl ligature +
Ł	<code>\[/L]</code>	Lslash	u0141	(Polish)
ł	<code>\[/l]</code>	lslash	u0142	(Polish)
Ø	<code>\[/O]</code>	Oslash	u00D8	(Scandinavian)
ø	<code>\[/o]</code>	oslash	u00F8	(Scandinavian)
Æ	<code>\[AE]</code>	AE	u00C6	
æ	<code>\[ae]</code>	ae	u00E6	
Œ	<code>\[OE]</code>	OE	u0152	
œ	<code>\[oe]</code>	oe	u0153	
IJ	<code>\[IJ]</code>	IJ	u0132	(Dutch)
ij	<code>\[ij]</code>	ij	u0133	(Dutch)
ı	<code>\[.i]</code>	dotlessi	u0131	(Turkish)
(N/A)	<code>\[.j]</code>	dotlessj	---	j without a dot

Accented Characters

Á	\['A]	Aacute	u0041_0301
Ć	\['C]	Cacute	u0043_0301
É	\['E]	Eacute	u0045_0301
Í	\['I]	Iacute	u0049_0301
Ó	\['O]	Oacute	u004F_0301
Ú	\['U]	Uacute	u0055_0301
Ý	\['Y]	Yacute	u0059_0301
á	\['a]	aacute	u0061_0301
ć	\['c]	cacute	u0063_0301
é	\['e]	eacute	u0065_0301
í	\['i]	iacute	u0069_0301
ó	\['o]	oacute	u006F_0301
ú	\['u]	uacute	u0075_0301
ý	\['y]	yacute	u0079_0301
Ä	\[:A]	Adieresis	u0041_0308
Ë	\[:E]	Edieresis	u0045_0308
Ï	\[:I]	Idieresis	u0049_0308
Ö	\[:O]	Odieresis	u004F_0308
Ü	\[:U]	Udieresis	u0055_0308
Ÿ	\[:Y]	Ydieresis	u0059_0308
ä	\[:a]	adieresis	u0061_0308
ë	\[:e]	edieresis	u0065_0308
ï	\[:i]	idieresis	u0069_0308
ö	\[:o]	odieresis	u006F_0308
ü	\[:u]	udieresis	u0075_0308
ÿ	\[:y]	ydieresis	u0079_0308
Â	\[^A]	Acircumflex	u0041_0302
Ê	\[^E]	Ecircumflex	u0045_0302
Î	\[^I]	Icircumflex	u0049_0302
Ô	\[^O]	Ocircumflex	u004F_0302
Û	\[^U]	Ucircumflex	u0055_0302
â	\[^a]	acircumflex	u0061_0302
ê	\[^e]	ecircumflex	u0065_0302
î	\[^i]	icircumflex	u0069_0302
ô	\[^o]	ocircumflex	u006F_0302
û	\[^u]	ucircumflex	u0075_0302
À	\[`A]	Agrave	u0041_0300
È	\[`E]	Egrave	u0045_0300
Ì	\[`I]	Igrave	u0049_0300
Ò	\[`O]	Ograve	u004F_0300
Ù	\[`U]	Ugrave	u0055_0300
à	\[`a]	agrave	u0061_0300
è	\[`e]	egrave	u0065_0300
ì	\[`i]	igrave	u0069_0300
ò	\[`o]	ograve	u006F_0300
ù	\[`u]	ugrave	u0075_0300
Ã	\[~A]	Atilde	u0041_0303
Ñ	\[~N]	Ntilde	u004E_0303
Õ	\[~O]	Otilde	u004F_0303
ã	\[~a]	atilde	u0061_0303
ñ	\[~n]	ntilde	u006E_0303

A with umlaut

<i>Output</i>	<i>Input name</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
õ	\[~o]	otilde	u006F_0303	
Š	\[vS]	Scaron	u0053_030C	
š	\[vs]	scaron	u0073_030C	
Ž	\[vZ]	Zcaron	u005A_030C	
ž	\[vz]	zcaron	u007A_030C	
Ç	\[,C]	Ccedilla	u0043_0327	
ç	\[,c]	ccedilla	u0063_0327	
Å	\[oA]	Aring	u0041_030A	
å	\[oa]	aring	u0061_030A	

Accents

The **composite** request is used to map most of the accents to non-spacing glyph names; the values given in parentheses are the original (spacing) ones.

<i>Output</i>	<i>Input name</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
˘	\[a"]	hungarumlaut	u030B (u02DD)	(Hungarian)
ˉ	\[a-]	macron	u0304 (u00AF)	
˙	\[a.]	dotaccent	u0307 (u02D9)	
ˆ	\[a^]	circumflex	u0302 (u005E)	
ˊ	\[aa]	acute	u0301 (u00B4)	+
ˋ	\[ga]	grave	u0300 (u0060)	+
˘	\[ab]	breve	u0306 (u02D8)	
¸	\[ac]	cedilla	u0327 (u00B8)	
¨	\[ad]	dieresis	u0308 (u00A8)	umlaut
ˇ	\[ah]	caron	u030C (u02C7)	háček
°	\[ao]	ring	u030A (u02DA)	circle
˜	\[a~]	tilde	u0303 (u007E)	
ˆ	\[ho]	ogonek	u0328 (u02DB)	hook
^	\[ha]	asciicircum	u005E	(spacing)
~	\[ti]	asciitilde	u007E	(spacing)

Quotes

„	\[Bq]	quotedblbase	u201E	low double comma quote
,	\[bq]	quotesinglbase	u201A	low single comma quote
“	\[lq]	quotedblleft	u201C	
”	\[rq]	quotedblright	u201D	
‘	\[oq]	quoteleft	u2018	single open quote
’	\[cq]	quoteright	u2019	single closing quote
'	\[aq]	quotesingle	u0027	apostrophe quote (ASCII 39)
"	\[dq]	quotedbl	u0022	double quote (ASCII 34)
«	\[Fo]	guillemotleft	u00AB	
»	\[Fc]	guillemotright	u00BB	
<	\[fo]	guilsinglleft	u2039	
>	\[fc]	guilsinglright	u203A	

Punctuation

¡	\[r!]	exclamdown	u00A1	
¿	\[r?]	questiondown	u00BF	
—	\[em]	emdash	u2014	+
–	\[en]	endash	u2013	
-	\[hy]	hyphen	u2010	+

Brackets

The extensible bracket pieces are font-invariant glyphs. In classical troff only one glyph was available to vertically extend brackets, braces, and parentheses: ‘bv’. We map it rather arbitrarily to u23AA.

Note that not all devices contain extensible bracket pieces which can be piled up with ‘\b’ due to the restrictions of the escape’s piling algorithm. A general solution to build brackets out of pieces is the following macro:

```
.\" Make a pile centered vertically 0.5em
.\" above the baseline.
.\" The first argument is placed at the top.
.\" The pile is returned in string 'pile'
.eo
.de pile-make
.  nr pile-wd 0
.  nr pile-ht 0
.  ds pile-args
.
.  nr pile-# \n[.]$
.  while \n[pile-#] \{\
.    nr pile-wd (\n[pile-wd] >? \w'\$[\n[pile-#]]')
.    nr pile-ht +(\n[rst] - \n[rsb])
.    as pile-args \v'\n[rsb]u'\
.    as pile-args \Z'\$[\n[pile-#]]'\
.    as pile-args \v'-\n[rst]u'\
.    nr pile-# -1
.  \}
.
.  ds pile \v'(-0.5m + (\n[pile-ht]u / 2u))'\
.  as pile \*[pile-args]\
.  as pile \v'((\n[pile-ht]u / 2u) + 0.5m)'\
.  as pile \h'\n[pile-wd]u'\
..
.ec
```

Another complication is the fact that some glyphs which represent bracket pieces in original troff can be used for other mathematical symbols also, for example ‘lf’ and ‘rf’ which provide the ‘floor’ operator. Other devices (most notably for DVI output) don’t unify such glyphs. For this reason, the four glyphs ‘lf’, ‘rf’, ‘lc’, and ‘rc’ are not unified with similarly looking bracket pieces. In **groff**, only glyphs with long names are guaranteed to pile up correctly for all devices (provided those glyphs exist).

<i>Output</i>	<i>Input name</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
[\[lB]	bracketleft	u005B	
]	\[rB]	bracketright	u005D	
{	\[lC]	braceleft	u007B	
}	\[rC]	braceright	u007D	
<	\[lA]	angleleft	u27E8	left angle bracket
>	\[rA]	angleright	u27E9	right angle bracket
	\[bv]	braceex	u23AA	vertical extension *** +
	\[braceex]	braceex	u23AA	
[\[bracketlefttp]	bracketlefttp	u23A1	
[\[bracketleftbt]	bracketleftbt	u23A3	
	\[bracketleftex]	bracketleftex	u23A2	
]	\[bracketrighttp]	bracketrighttp	u23A4	

<i>Output</i>	<i>Input name</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
]	\[bracketrightbt]	bracketrightbt	u23A6	
	\[bracketrighttex]	bracketrighttex	u23A5	
[\[lt]	bracelefttp	u23A7	+
[\[braceleftttp]	bracelefttp	u23A7	
{	\[lk]	braceleftmid	u23A8	+
{	\[bracelefttmid]	braceleftmid	u23A8	
[\[lb]	braceleftbt	u23A9	+
[\[braceleftbt]	braceleftbt	u23A9	
	\[bracelefttex]	bracelefttex	u23AA	
]	\[rt]	bracerighttp	u23AB	+
]	\[bracerightttp]	bracerighttp	u23AB	
}	\[rk]	bracerightmid	u23AC	+
}	\[bracerighttmid]	bracerightmid	u23AC	
]	\[rb]	bracerightbt	u23AD	+
]	\[bracerightbt]	bracerightbt	u23AD	
	\[bracerighttex]	bracerighttex	u23AA	
(\[parenlefttp]	parenlefttp	u239B	
(\[parenleftbt]	parenleftbt	u239D	
	\[parenlefttex]	parenlefttex	u239C	
)	\[parenrighttp]	parenrighttp	u239E	
)	\[parenrightbt]	parenrightbt	u23A0	
	\[parenrighttex]	parenrighttex	u239F	
<i>Arrows</i>				
←	\[<-]	arrowleft	u2190	+
→	\[->]	arrowright	u2192	+
↔	\[<>]	arrowboth	u2194	(horizontal)
↓	\[da]	arrowdown	u2193	+
↑	\[ua]	arrowup	u2191	+
↕	\[va]	arrowupdn	u2195	
⇐	\[lA]	arrowdblleft	u21D0	
⇒	\[rA]	arrowdblright	u21D2	
⇔	\[hA]	arrowdblboth	u21D4	(horizontal)
⇓	\[dA]	arrowdbldown	u21D3	
⇑	\[uA]	arrowdblup	u21D1	
(N/A)	\[vA]	uni21D5	u21D5	vertical double-headed double arrow
—	\[an]	arrowhorizex	u23AF	horizontal arrow extension

Lines

The font-invariant glyphs ‘br’, ‘ul’, and ‘rn’ form corners; they can be used to build boxes. Note that both the PostScript and the Unicode-derived names of these three glyphs are just rough approximations.

‘rn’ also serves in classical troff as the horizontal extension of the square root sign.

‘ru’ is a font-invariant glyph, namely a rule of length 0.5m.

<i>Output</i>	<i>Input name</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
	\[ba]	bar	u007C	
	\[br]	SF110000	u2502	box rule +
—	\[ul]	underscore	u005F	+

<i>Output</i>	<i>Input name</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
—	<code>\[rn]</code>	overline	u203E	use ‘ <code>\[radicallex]</code> ’ for continuation of square root +
—	<code>\[ru]</code>	---	---	baseline rule +
⋈	<code>\[bb]</code>	brokenbar	u00A6	
/	<code>\[sl]</code>	slash	u002F	+
\	<code>\[rs]</code>	backslash	u005C	reverse solidus

Text markers

○	<code>\[ci]</code>	circle	u25CB	+
•	<code>\[bu]</code>	bullet	u2022	+
‡	<code>\[dd]</code>	daggerdbl	u2021	double dagger sign +
†	<code>\[dg]</code>	dagger	u2020	+
◇	<code>\[lz]</code>	lozenge	u25CA	
◻	<code>\[sq]</code>	uni25A1	u25A1	white square +
¶	<code>\[ps]</code>	paragraph	u00B6	
§	<code>\[sc]</code>	section	u00A7	+
☞	<code>\[lh]</code>	uni261C	u261C	hand pointing left +
☞	<code>\[rh]</code>	a14	u261E	hand pointing right +
@	<code>\[at]</code>	at	u0040	
#	<code>\[sh]</code>	numbersign	u0023	
↵	<code>\[CR]</code>	carriagereturn	u21B5	
✓	<code>\[OK]</code>	a19	u2713	check mark, tick

Legal Symbols

©	<code>\[co]</code>	copyright	u00A9	+
®	<code>\[rg]</code>	registered	u00AE	+
™	<code>\[tm]</code>	trademark	u2122	
(N/A)	<code>\[bs]</code>	---	---	AT&T Bell Labs logo (not used in groff) +

Currency symbols

\$	<code>\[Do]</code>	dollar	u0024	
¢	<code>\[ct]</code>	cent	u00A2	+
€	<code>\[eu]</code>	---	u20AC	official Euro symbol
€	<code>\[Eu]</code>	Euro	u20AC	font-specific Euro glyph variant
¥	<code>\[Ye]</code>	yen	u00A5	
£	<code>\[Po]</code>	sterling	u00A3	British currency sign
¤	<code>\[Cs]</code>	currency	u00A4	Scandinavian currency sign
f	<code>\[Fn]</code>	florin	u0192	Dutch currency sign

Units

°	<code>\[de]</code>	degree	u00B0	+
‰	<code>\[%0]</code>	perthousand	u2030	per thousand, per mille sign
′	<code>\[fm]</code>	minute	u2032	footmark, prime +
″	<code>\[sd]</code>	second	u2033	
μ	<code>\[mc]</code>	mu	u00B5	micro sign
ª	<code>\[Of]</code>	ordfeminine	u00AA	
º	<code>\[Om]</code>	ordmasculine	u00BA	

Logical Symbols

^	<code>\[AN]</code>	logicaland	u2227	
∨	<code>\[OR]</code>	logicalor	u2228	
¬	<code>\[no]</code>	logicalnot	u00AC	+

<i>Output</i>	<i>Input name</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
¬	\[tno]	logicalnot	u00AC	text variant of ‘no’
∃	\[te]	existential	u2203	there exists, existential quantifier
∀	\[fa]	universal	u2200	for all, universal quantifier
∋	\[st]	suchthat	u220B	
∴	\[3d]	therefore	u2234	
∴	\[tf]	therefore	u2234	
	\[or]	bar	u007C	bitwise OR operator (as used in C) +

Mathematical Symbols

½	\[12]	onehalf	u00BD	+
¼	\[14]	onequarter	u00BC	+
¾	\[34]	threequarters	u00BE	+
⅛	\[18]	oneeighth	u215B	
⅜	\[38]	threeeighths	u215C	
⅝	\[58]	fiveeighths	u215D	
⅞	\[78]	seveneighths	u215E	
¹	\[S1]	onesuperior	u00B9	
²	\[S2]	twosuperior	u00B2	
³	\[S3]	threesuperior	u00B3	
+	\[pl]	plus	u002B	plus sign in special font +
−	\[mi]	minus	u2212	minus sign in special font +
(N/A)	\[+-]	uni2213	u2213	
±	\[+-]	plusminus	u00B1	+
±	\[t+-]	plusminus	u00B1	text variant of ‘+-’
·	\[pc]	periodcentered	u00B7	
·	\[md]	dotmath	u22C5	multiplication dot
×	\[mu]	multiply	u00D7	+
×	\[tmu]	multiply	u00D7	text variant of ‘mu’
⊗	\[c*]	circlemultiply	u2297	multiply sign in a circle
⊕	\[c+]	circleplus	u2295	plus sign in a circle
÷	\[di]	divide	u00F7	division sign +
÷	\[tdi]	divide	u00F7	text variant of ‘di’
/	\[f/]	fraction	u2044	bar for fractions
*	\[**]	asteriskmath	u2217	+
≤	\[<=]	lessequal	u2264	+
≥	\[>=]	greaterequal	u2265	+
≪	\[<<]	uni226A	u226A	much less
≫	\[>>]	uni226B	u226B	much greater
=	\[eq]	equal	u003D	equals sign in special font +
≠	\[!=]	notequal	u003D_0338	+
≡	\[==]	equivalence	u2261	+
≢	\[ne]	uni2262	u2261_0338	
≐	\[=~]	congruent	u2245	approx. equal
≈	\[=]	uni2243	u2243	asymptot. equal to +
~	\[ap]	similar	u223C	+
≈	\[~~]	approxequal	u2248	almost equal to
≈	\[~=]	approxequal	u2248	
∝	\[pt]	proportional	u221D	+
∅	\[es]	emptyset	u2205	+
∈	\[mo]	element	u2208	+
∉	\[nm]	notelement	u2208_0338	

<i>Output</i>	<i>Input name</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
\subset	<code>\[sb]</code>	probersubset	u2282	+
$\not\subset$	<code>\[nb]</code>	notsubset	u2282_0338	
\supset	<code>\[sp]</code>	probersuperset	u2283	+
$\not\supset$	<code>\[nc]</code>	uni2285	u2283_0338	not superset
\subseteq	<code>\[ib]</code>	reflexsubset	u2286	+
\supseteq	<code>\[ip]</code>	reflexsuperset	u2287	+
\cap	<code>\[ca]</code>	intersection	u2229	intersection, cap +
\cup	<code>\[cu]</code>	union	u222A	union, cup +
\angle	<code>\[/_]</code>	angle	u2220	
\perp	<code>\[pp]</code>	perpendicular	u22A5	
\int	<code>\[is]</code>	integral	u222B	+
\int	<code>\[integral]</code>	integral	u222B	***
Σ	<code>\[sum]</code>	summation	u2211	***
Π	<code>\[product]</code>	product	u220F	***
(N/A)	<code>\[coproduct]</code>	uni2210	u2210	***
∇	<code>\[gr]</code>	gradient	u2207	+
$\sqrt{\quad}$	<code>\[sr]</code>	radical	u221A	square root +
$\sqrt{\quad}$	<code>\[sqrt]</code>	radical	u221A	***
$\sqrt{\quad}$	<code>\[radicalex]</code>	radicalex	---	continuation of square root
$\sqrt{\quad}$	<code>\[sqrtex]</code>	radicalex	---	***
\lceil	<code>\[lc]</code>	uni2308	u2308	left ceiling +
\rceil	<code>\[rc]</code>	uni2309	u2309	right ceiling +
\lfloor	<code>\[lf]</code>	uni230A	u230A	left floor +
\rfloor	<code>\[rf]</code>	uni230B	u230B	right floor +
∞	<code>\[if]</code>	infinity	u221E	+
\aleph	<code>\[Ah]</code>	aleph	u2135	
\Im	<code>\[Im]</code>	Ifraktur	u2111	Gothic I, imaginary
\Re	<code>\[Re]</code>	Rfraktur	u211C	Gothic R, real
\wp	<code>\[wp]</code>	weierstrass	u2118	Weierstrass p
∂	<code>\[pd]</code>	partialdiff	u2202	partial differentiation sign +
\hbar	<code>\[-h]</code>	uni210F	u210F	Planck constant over two pi
\hbar	<code>\[hbar]</code>	uni210F	u210F	

Greek glyphs

These glyphs are intended for technical use, not for real Greek; normally, the uppercase letters have upright shape, and the lowercase ones are slanted. There is a problem with the mapping of letter phi to Unicode. Prior to Unicode version 3.0, the difference between U+03C6, GREEK SMALL LETTER PHI, and U+03D5, GREEK PHI SYMBOL, was not clearly described; only the glyph shapes in the Unicode book could be used as a reference. Starting with Unicode 3.0, the reference glyphs have been exchanged and described verbally also: In mathematical context, U+03D5 is the stroked variant and U+03C6 the curly glyph. Unfortunately, most font vendors didn't update their fonts to this (incompatible) change in Unicode. At the time of this writing (January 2006), it is not clear yet whether the Adobe Glyph Names 'phi' and 'phi1' also change its meaning if used for mathematics, thus compatibility problems are likely to happen – being conservative, groff currently assumes that 'phi' in a PostScript symbol font is the stroked version.

In groff, symbol '`\[*f]`' always denotes the stroked version of phi, and '`\[+f]`' the curly variant.

A	<code>\[*A]</code>	Alpha	u0391	+
B	<code>\[*B]</code>	Beta	u0392	+
Γ	<code>\[*G]</code>	Gamma	u0393	+
Δ	<code>\[*D]</code>	Delta	u0394	+
E	<code>\[*E]</code>	Epsilon	u0395	+

<i>Output</i>	<i>Input name</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
Z	\[*Z]	Zeta	u0396	+
H	\[*Y]	Eta	u0397	+
Θ	\[*H]	Theta	u0398	+
I	\[*I]	Iota	u0399	+
K	\[*K]	Kappa	u039A	+
Λ	\[*L]	Lambda	u039B	+
M	\[*M]	Mu	u039C	+
N	\[*N]	Nu	u039D	+
Ξ	\[*C]	Xi	u039E	+
O	\[*O]	Omicron	u039F	+
Π	\[*P]	Pi	u03A0	+
P	\[*R]	Rho	u03A1	+
Σ	\[*S]	Sigma	u03A3	+
T	\[*T]	Tau	u03A4	+
Υ	\[*U]	Upsilon	u03A5	+
Φ	\[*F]	Phi	u03A6	+
X	\[*X]	Chi	u03A7	+
Ψ	\[*Q]	Psi	u03A8	+
Ω	\[*W]	Omega	u03A9	+
α	\[*a]	alpha	u03B1	+
β	\[*b]	beta	u03B2	+
γ	\[*g]	gamma	u03B3	+
δ	\[*d]	delta	u03B4	+
ε	\[*e]	epsilon	u03B5	+
ζ	\[*z]	zeta	u03B6	+
η	\[*y]	eta	u03B7	+
θ	\[*h]	theta	u03B8	+
ι	\[*i]	iota	u03B9	+
κ	\[*k]	kappa	u03BA	+
λ	\[*l]	lambda	u03BB	+
μ	\[*m]	mu	u03BC	+
ν	\[*n]	nu	u03BD	+
ξ	\[*c]	xi	u03BE	+
ο	\[*o]	omicron	u03BF	+
π	\[*p]	pi	u03C0	+
ρ	\[*r]	rho	u03C1	+
ς	\[ts]	sigma1	u03C2	terminal sigma +
σ	\[*s]	sigma	u03C3	+
τ	\[*t]	tau	u03C4	+
υ	\[*u]	upsilon	u03C5	+
φ	\[*f]	phi	u03D5	(stroked glyph)+
χ	\[*x]	chi	u03C7	+
ψ	\[*q]	psi	u03C8	+
ω	\[*w]	omega	u03C9	+
ϑ	\[+h]	theta1	u03D1	variant theta
φ	\[+f]	phi1	u03C6	variant phi (curly shape)
ϖ	\[+p]	omega1	u03D6	variant pi, looking like omega
(N/A)	\[+e]	uni03F5	u03F5	variant epsilon
<i>Card symbols</i>				
♣	\[CL]	club	u2663	black club suit

<i>Output</i>	<i>Input name</i>	<i>PostScript name</i>	<i>Unicode decomposed</i>	<i>Notes</i>
♠	\[SP]	spade	u2660	black spade suit
♥	\[HE]	heart	u2665	black heart suit
(N/A)	\[u2661]	uni2661	u2661	white heart suit
♦	\[DI]	diamond	u2666	black diamond suit
(N/A)	\[u2662]	uni2662	u2662	white diamond suit

AUTHOR

Copyright © 1989-2000, 2001, 2002, 2003, 2004, 2006, 2008, 2009 Free Software Foundation, Inc.

This document is distributed under the terms of the FDL (GNU Free Documentation License) version 1.3 or later. You should have received a copy of the FDL on your system, it is also available on-line at the [GNU copyleft site](#).

This document is part of *groff*, the GNU roff distribution. It was written by [James Clark](#) with additions by [Werner Lemberg](#) and [Bernd Warken](#).

SEE ALSO

[groff\(1\)](#) the GNU roff formatter

[groff\(7\)](#) a short reference of the groff formatting language

An extension to the troff character set for Europe, E.G. Keizer, K.J. Simonsen, J. Akkerhuis; EUUG Newsletter, Volume 9, No. 2, Summer 1989

[The Unicode Standard](#)