

# ISO character entities and their L<sup>A</sup>T<sub>E</sub>X equivalents

Vidar Bronken Gundersen, Rune Mathisen

Revision: 4.03, Date: 2001-01-14

## Abstract

This is an attempt to make an overview of SGML/XML character entities in ISO 8879:1986 and their L<sup>A</sup>T<sub>E</sub>X equivalents. It is also a handy reference to the ISO character entities for non-L<sup>A</sup>T<sub>E</sub>X users as it shows the visual appearance of the character entities.

The latest version of this documentation and the source material resides at <http://www.bitjungle.com/~isoent/>.

If you find errors or have comments/suggestions of any kind, please email us [isoent@bitjungle.com](mailto:isoent@bitjungle.com).

Permissions to use, copy, modify and distribute this document for any purpose and without fee is hereby granted in perpetuity that the copyright notice below and this paragraph appear in all copies. It is provided “as is” without expressed or implied warranty.

©2000 Vidar Bronken Gundersen, Rune Mathisen  
<http://www.bitjungle.com/~isoent/>

# Contents






































<b>1</b>	<b>General and publishing symbols</b>	<b>3</b>
1.1	Publishing (pub) . . . . .	3
1.2	Numeric and special graphic (num) . . . . .	5
1.3	Box and line drawing (box) . . . . .	7
<b>2</b>	<b>Mathematics and sciences</b>	<b>9</b>
2.1	General technical (tech) . . . . .	9
2.2	Greek symbols (grk3) . . . . .	11
2.3	Alternative greek symbols (grk4) . . . . .	12
2.4	Added Math Symbols: Ordinary (amso) . . . . .	13
2.5	Added Math Symbols: Binary operators (amsb) . . . . .	14
2.6	Added Math Symbols: Relations (amsr) . . . . .	15
2.7	Added Math Symbols: Negated relations (amsn) . . . . .	18
2.8	Added Math Symbols: Arrow Relations (amsa) . . . . .	20
2.9	Added Math Symbols: Delimiters (amsc) . . . . .	21
<b>3</b>	<b>Latin based alphabets</b>	<b>22</b>
3.1	Added Latin 1 (lat1) . . . . .	22
3.2	Added Latin 2 (lat2) . . . . .	24
3.3	Diacritical marks (dia) . . . . .	27
<b>4</b>	<b>Non-Latin based alphabets</b>	<b>28</b>
4.1	Russian Cyrillic (cyr1) . . . . .	28
4.2	Non-Russian Cyrillic (cyr2) . . . . .	30
4.3	Greek letters (grk1) . . . . .	30
4.4	Monotoniko Greek (grk2) . . . . .	32
<b>5</b>	<b>Notes</b>	<b>33</b>
5.1	Mapping data (ent.xml) . . . . .	33
5.2	The “Description” column . . . . .	33
5.3	Required L <sup>A</sup> T <sub>E</sub> X packages and fonts . . . . .	33
5.4	Characters unavailable to normal PDF output . . . . .	34
5.5	Displaying Cyrillic characters . . . . .	34
5.6	Latin 1 encoding . . . . .	34
5.7	Numeric and publishing symbols . . . . .	34
<b>6</b>	<b>References</b>	<b>35</b>

# 1 General and publishing symbols

## 1.1 Publishing (pub)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
emsp	<code>\hspace{1em}</code>			=em space
ensp	<code>\hspace{0.5em}</code>			=en space (1/2-em)
emsp13	<code>\hspace{0.33em}</code>			=1/3-em space
emsp14	<code>\hspace{0.25em}</code>			=1/4-em space
numsp	<code>\hphantom{0}</code>			=digit space (width of a number)
puncsp	<code>\hphantom{,}</code>			=punctuation space (width of comma)
thinsp	<code>\hspace{0.167em}</code>			=thin space (1/6-em)
hairsp	<code>\hspace{1pt}</code>			=hair space
mdash	---	—		=em dash
ndash	--	–		=en dash
dash	-	-		=hyphen (true graphic)
blank	<code>\textvisiblespace</code>	␣		=significant blank symbol
hellip	<code>\ldots</code>	...		=ellipsis (horizontal)
nldr	<code>\nldr</code>	..	local	=double baseline dot (en leader)
frac13	<code>\sfrac{1}{3}</code>	1/3	(math) local	=fraction one-third
frac23	<code>\sfrac{2}{3}</code>	2/3	(math) local	=fraction two-thirds
frac15	<code>\sfrac{1}{5}</code>	1/5	(math) local	=fraction one-fifth
frac25	<code>\sfrac{2}{5}</code>	2/5	(math) local	=fraction two-fifths
frac35	<code>\sfrac{3}{5}</code>	3/5	(math) local	=fraction three-fifths
frac45	<code>\sfrac{4}{5}</code>	4/5	(math) local	=fraction four-fifths
frac16	<code>\sfrac{1}{6}</code>	1/6	(math) local	=fraction one-sixth
frac56	<code>\sfrac{5}{6}</code>	5/6	(math) local	=fraction five-sixths
incare	<code>{^c\!/\/!_o}</code>	c/o	(math)	=in-care-of symbol
block	<code>\block</code>	■	local	=full block
uhblk	<code>\uhblk</code>	◼	local	=upper half block
lhblk	<code>\lhblk</code>	◻	local	=lower half block
blk14	<code>\textcolor[gray]{.75}{\block}</code>	■	color,local	=25% shaded block
blk12	<code>\textcolor[gray]{.5}{\block}</code>	■	color,local	=50% shaded block
blk34	<code>\textcolor[gray]{.25}{\block}</code>	■	color,local	=75% shaded block
marker	<code>\marker</code>	■	local	=histogram marker
cir	<code>\circ</code>	○	(math)	/circ B: =circle, open
squ	<code>\square</code>	□	(math) amssymb	=square, open

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
rect	<code>\fbox{~~}</code>			=rectangle, open
utri	<code>\vartriangle</code>		(math) amssymb	/triangle =up triangle, open
dtri	<code>\triangledown</code>		(math) amssymb	/triangledown =down triangle, open
star	<code>\ding{73}</code>		pifont	=star, open
bull	<code>\textbullet</code>			/bullet B: =round bullet, filled
sqf	<code>\blacksquare</code>		(math) amssymb	/blacksquare =sq bullet, filled
utrif	<code>\blacktriangle</code>		(math) amssymb	/blacktriangle =up tri, filled
dtrif	<code>\blacktriangledown</code>		(math) amssymb	/blacktriangledown =dn tri, filled
ltrif	<code>\blacktriangleleft</code>		(math) amssymb	/blacktriangleleft R: =l tri, filled
rtrif	<code>\blacktriangleright</code>		(math) amssymb	/blacktriangleright R: =r tri, filled
clubs	<code>\ding{168}</code>		pifont	/clubsuit =club suit symbol
diams	<code>\ding{169}</code>		pifont	/diamondsuit =diamond suit symbol
hearts	<code>\ding{170}</code>		pifont	/heartsuit =heart suit symbol
spades	<code>\ding{171}</code>		pifont	/spadesuit =spades suit symbol
malt	<code>\maltese</code>		amsfonts	/maltese =maltese cross
dagger	<code>\dag</code>			/dagger B: =dagger
Dagger	<code>\ddag</code>			/ddagger B: =double dagger
check	<code>\checkmark</code>		amsfonts	/checkmark =tick, check mark
cross	<code>\ding{55}</code>		pifont	=ballot cross
sharp	<code>\sharp</code>		(math)	/sharp =musical sharp
flat	<code>\flat</code>		(math)	/flat =musical flat
male	<code>\male</code>		wasysym	=male symbol
female	<code>\female</code>		wasysym	=female symbol
phone	<code>\phone</code>		wasysym	=telephone symbol
telrec	<code>\recorder</code>		wasysym	=telephone recorder symbol
copysr	<code>\textcircledP</code>		textcomp	=sound recording copyright sign
caret	<code>\mathchar"1356</code>		(math)	=caret (insertion mark)
lsquor	<code>,</code>			=rising single quote, left (low)
ldquor	<code>,,</code>			=rising dbl quote, left (low)
fflig	<code>ff</code>			small ff ligature
filig	<code>fi</code>			small fi ligature
fjlig	<code>fj</code>			small fj ligature
fflig	<code>ffi</code>			small ffi ligature
fflig	<code>ffl</code>			small ffl ligature
flig	<code>fl</code>			small fl ligature
mldr	<code>\mldr</code>		local	em leader
rdquor	<code>‘</code>			rising dbl quote, right (high)

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
rsquor	‘	‘		rising single quote, right (high)
vellip	\vdots	⋮	(math)	vertical ellipsis
hybull	\hybull	–	local	rectangle, filled (hyphen bullet)
loz	\lozenge	◇	(math) amssymb	/lozenge - lozenge or total mark
lozf	\blacklozenge	◆	(math) amssymb	/blacklozenge - lozenge, filled
ltr	\triangleleft	◁	(math)	/triangleleft B: l triangle, open
rtri	\triangleright	▷	(math)	/triangleright B: r triangle, open
starf	\bigstar	★	(math) amssymb	/bigstar - star, filled
natur	\natural	♮	(math)	/natural - music natural
rx	\textrecipe	℞	textcomp	pharmaceutical prescription (Rx)
sext	\ding{86}	✳	pifont	sextile (6-pointed star)
target	\mathchar"2208	⊕	(math)	register mark or target
dlcrop	\dlcrop	┘	local	downward left crop mark
drcrop	\drcrop	┘	local	downward right crop mark
ulcrop	\ulcrop	┘	local	upward left crop mark
urcrop	\urcrop	┘	local	upward right crop mark

## 1.2 Numeric and special graphic (num)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
half	\sfrac{1}{2}	1/2	(math) local	=fraction one-half
frac12	\sfrac{1}{2}	1/2	(math) local	=fraction one-half
frac14	\sfrac{1}{4}	1/4	(math) local	=fraction one-quarter
frac34	\sfrac{3}{4}	3/4	(math) local	=fraction three-quarters
frac18	\sfrac{1}{8}	1/8	(math) local	=fraction one-eighth
frac38	\sfrac{3}{8}	3/8	(math) local	=fraction three-eighths
frac58	\sfrac{5}{8}	5/8	(math) local	=fraction five-eighths
frac78	\sfrac{7}{8}	7/8	(math) local	=fraction seven-eighths
sup1	ˆ1	¹	(math)	=superscript one
sup2	ˆ2	²	(math)	=superscript two
sup3	ˆ3	³	(math)	=superscript three
plus	+	+	(math)	=plus sign
plusmn	\pm	±	(math)	/pm B: =plus-or-minus sign

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
lt	<code>\textless</code>	<		=less-than sign R:
equals	=	=	(math)	=equals sign R:
gt	<code>\textgreater</code>	>		=greater-than sign R:
divide	<code>\div</code>	÷	(math)	/div B: =divide sign
times	<code>\times</code>	×	(math)	/times B: =multiply sign
curren	<code>\textcurrency</code>	¤	textcomp	=general currency sign
pound	<code>\pounds</code>	£		=pound sign
dollar	<code>\\$</code>	\$		=dollar sign
cent	<code>\textcent</code>	¢	textcomp	=cent sign
yen	<code>\textyen</code>	¥	textcomp	/yen =yen sign
num	<code>\#</code>	#		=number sign
percent	<code>\%</code>	%		=percent sign
amp	<code>\&amp;</code>	&		=ampersand
ast	<code>\ast</code>	*	(math)	/ast B: =asterisk
commat	@	@		=commercial at
lsqb	[	[		/lbrack O: =left square bracket
bsol	<code>\textbackslash</code>	\		/backslash =reverse solidus
rsqb	]	]		/rbrack C: =right square bracket
lcub	<code>\{</code>	{		/lbrace O: =left curly bracket
horbar	---	—		=horizontal bar
verbar	<code>\vert</code>		(math)	/vert =vertical bar
rcub	<code>\}</code>	}		/rbrace C: =right curly bracket
micro	<code>\textmu</code>	μ	textcomp	=micro sign
ohm	<code>\textohm</code>	Ω	textcomp	=ohm sign
deg	<code>\textdegree</code>	°		=degree sign
ordm	<code>\textordmasculine</code>	º		=ordinal indicator, masculine
ordf	<code>\textordfeminine</code>	ª		=ordinal indicator, feminine
sect	<code>\S</code>	§		=section sign
para	<code>\P</code>	¶		=pilcrow (paragraph sign)
middot	<code>\textperiodcentered</code>	·		/centerdot B: =middle dot
larr	<code>\leftarrow</code>	←	(math)	/leftarrow /gets A: =leftward arrow
rarr	<code>\rightarrow</code>	→	(math)	/rightarrow /to A: =rightward arrow
uarr	<code>\uparrow</code>	↑	(math)	/uparrow A: =upward arrow
darr	<code>\downarrow</code>	↓	(math)	/downarrow A: =downward arrow
copy	<code>\copyright</code>	©		=copyright sign
reg	<code>\textregistered</code>	®		/circledR =registered sign
trade	<code>\texttrademark</code>	™		=trade mark sign

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
brvbar	<code>\textbrokenbar</code>		textcomp	=broken (vertical) bar
not	<code>\lnot</code>	¬	(math)	/neg /lnot =not sign
sung	<code>\textmusicalnote</code>	♪	textcomp	=music note (sung text sign)
excl	<code>!</code>	!		=exclamation mark
iexcl	<code>!‘</code>	¡		=inverted exclamation mark
quot	<code>"</code>	”		=quotation mark
apos	<code>,</code>	,		=apostrophe
lpar	<code>(</code>	(		O: =left parenthesis
rpar	<code>)</code>	)		C: =right parenthesis
comma	<code>,</code>	,		P: =comma
lowbar	<code>\underline{\space\space}</code>	—		=low line
hyphen	<code>-</code>	-		=hyphen
period	<code>.</code>	.		=full stop, period
sol	<code>/</code>	/		=solidus
colon	<code>:</code>	:		/colon P:
semi	<code>;</code>	;		=semicolon P:
quest	<code>?</code>	?		=question mark
iquest	<code>?‘</code>	¿		=inverted question mark
laquo	<code>\guillemotleft</code>	«	fontenc:T1	=angle quotation mark, left
raquo	<code>\guillemotright</code>	»	fontenc:T1	=angle quotation mark, right
lsquo	<code>‘</code>	‘		=single quotation mark, left
rsquo	<code>’</code>	’		=single quotation mark, right
ldquo	<code>‘‘</code>	“		=double quotation mark, left
rdquo	<code>’’</code>	”		=double quotation mark, right
nbspc	<code>~</code>			=no break (required) space
shy	<code>\-</code>			=soft hyphen

### 1.3 Box and line drawing (box)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
boxh	<code>\boxh</code>	—	local	horizontal line
boxv	<code>\boxv</code>		local	vertical line
boxur	<code>\boxur</code>	┐	local	upper right quadrant
boxul	<code>\boxul</code>	┌	local	upper left quadrant
boxdl	<code>\boxdl</code>	└	local	lower left quadrant

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
boxdr	\boxdr		local	lower right quadrant
boxvr	\boxvr		local	upper and lower right quadrants
boxhu	\boxhu		local	upper left and right quadrants
boxvl	\boxvl		local	upper and lower left quadrants
boxhd	\boxhd		local	lower left and right quadrants
boxvh	\boxvh		local	all four quadrants
boxvR	\boxvR		local	upper and lower right quadrants
boxhU	\boxhU		local	upper left and right quadrants
boxvL	\boxvL		local	upper and lower left quadrants
boxhD	\boxhD		local	lower left and right quadrants
boxvH	\boxvH		local	all four quadrants
boxH	\boxH		local	horizontal line
boxV	\boxV		local	vertical line
boxUR	\boxUR		local	upper right quadrant
boxUL	\boxUL		local	upper left quadrant
boxDL	\boxDL		local	lower left quadrant
boxDR	\boxDR		local	lower right quadrant
boxVR	\boxVR		local	upper and lower right quadrants
boxHU	\boxHU		local	upper left and right quadrants
boxVL	\boxVL		local	upper and lower left quadrants
boxHD	\boxHD		local	lower left and right quadrants
boxVH	\boxVH		local	all four quadrants
boxVr	\boxVr		local	upper and lower right quadrants
boxHu	\boxHu		local	upper left and right quadrants
boxVl	\boxVl		local	upper and lower left quadrants
boxHd	\boxHd		local	lower left and right quadrants
boxVh	\boxVh		local	all four quadrants
boxuR	\boxuR		local	upper right quadrant
boxUl	\boxUl		local	upper left quadrant
boxdL	\boxdL		local	lower left quadrant

continued on next page...



ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
boxDr	<code>\boxDr</code>		local	lower right quadrant
boxUr	<code>\boxUr</code>		local	upper right quadrant
boxuL	<code>\boxuL</code>		local	upper left quadrant
boxDl	<code>\boxDl</code>		local	lower left quadrant
boxdR	<code>\boxdR</code>		local	lower right quadrant

## 2 Mathematics and sciences

### 2.1 General technical (tech)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
Dot	<code>\ddot{}</code>	¨	(math)	=dieresis or umlaut mark
fnoF	<code>f</code>	<i>f</i>	(math)	=function of (italic small f)
Verbar	<code>\Vert</code>		(math)	/Vert =dbl vertical bar
permil	<code>\textperthousand</code>	‰	textcomp	=per thousand
prime	<code>\prime</code>	'	(math)	/prime =prime or minute
Prime	<code>{''}</code>	''	(math)	=double prime or second
tprime	<code>{'''}</code>	'''	(math)	triple prime
tdot	<code>\ddot{ }\kern 3pt\dot{ }</code>	⋯	(math)	three dots above
DotDot	<code>\ddot{ }\kern 4.5pt\ddot{ }</code>	⋰	(math)	four dots above
hamilt	<code>H</code>	<i>ℋ</i>	(math) mathrsfs	Hamiltonian (script capital H)
lagran	<code>L</code>	<i>ℒ</i>	(math) mathrsfs	Lagrangian (script capital L)
angst	<code>\AA</code>	Å		Angstrom =capital A, ring
bernou	<code>B</code>	<i>ℬ</i>	(math) mathrsfs	Bernoulli function (script capital B)
phmmat	<code>M</code>	<i>ℳ</i>	(math) mathrsfs	physics M-matrix (script capital M)
order	<code>\mathit{o}</code>	<i>o</i>	(math)	order of (script small o)
aleph	<code>\aleph</code>	ℵ	(math)	/aleph =aleph, Hebrew
lArr	<code>\Leftarrow</code>	⇐	(math)	/Leftarrow A: =is implied by
rArr	<code>\Rightarrow</code>	⇒	(math)	/Rightarrow A: =implies
iff	<code>\iff</code>	⇔	(math)	/iff =if and only if
forall	<code>\forall</code>	∀	(math)	/forall =for all
part	<code>\partial</code>	∂	(math)	/partial =partial differential
exist	<code>\exists</code>	∃	(math)	/exists =at least one exists
nabla	<code>\nabla</code>	∇	(math)	/nabla =del, Hamilton operator
isin	<code>\in</code>	∈	(math)	/in R: =set membership

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
notin	<code>\not\in</code>	∉	(math)	N: negated set membership
ni	<code>\ni</code>	∋	(math)	/ni /owns R: =contains
minus	<code>-</code>	−	(math)	B: =minus sign
mnplus	<code>\mp</code>	∓	(math)	/mp B: =minus-or-plus sign
lowast	<code>\ast</code>	*	(math)	low asterisk
compfn	<code>\circ</code>	∘	(math)	B: composite function (small circle)
radic	<code>\surd</code>	√	(math)	/surd =radical
prop	<code>\propto</code>	∝	(math)	/propto R: =is proportional to
infin	<code>\infty</code>	∞	(math)	/infty =infinity
ang90	<code>\angle</code>	∟	(math) local	=right (90 degree) angle
angsph	<code>\sphericalangle</code>	∠	(math) amssymb	/sphericalangle =angle-spherical
par	<code>\parallel</code>	∥	(math)	/parallel R: =parallel
and	<code>\wedge</code>	∧	(math)	/wedge /land B: =logical and
or	<code>\vee</code>	∨	(math)	/vee /lor B: =logical or
cap	<code>\cap</code>	∩	(math)	/cap B: =intersection
cup	<code>\cup</code>	∪	(math)	/cup B: =union or logical sum
int	<code>\int</code>	∫	(math)	/int L: =integral operator
conint	<code>\oint</code>	∮	(math)	/oint L: =contour integral operator
there4	<code>\therefore</code>	∴	(math) amssymb	/therefore R: =therefore
becaus	<code>\because</code>	∵	(math) amssymb	/because R: =because
sim	<code>\sim</code>	≈	(math)	/sim R: =similar
sime	<code>\simeq</code>	≈	(math)	/simeq R: =similar, equals
cong	<code>\cong</code>	≅	(math)	/cong R: =congruent with
ap	<code>\approx</code>	≈	(math)	/approx R: =approximate
wedgeq	<code>\stackrel{\wedge}{=}</code>	≡	(math)	R: corresponds to (wedge, equals)
ne	<code>\neq</code>	≠	(math)	/ne /neq R: =not equal
equiv	<code>\equiv</code>	≡	(math)	/equiv R: =identical with
le	<code>\leq</code>	≤	(math)	/leq /le R: =less-than-or-equal
ge	<code>\geq</code>	≥	(math)	/geq /ge R: =greater-than-or-equal
sub	<code>\subset</code>	⊂	(math)	/subset R: =subset or is implied by
sup	<code>\supset</code>	⊃	(math)	/supset R: =superset or implies
sube	<code>\subseteq</code>	⊆	(math)	/subseteq R: =subset, equals
supe	<code>\supseteq</code>	⊇	(math)	/supseteq R: =superset, equals
bottom	<code>\perp</code>	⊥	(math)	/bot B: =perpendicular
perp	<code>\perp</code>	⊥	(math)	/perp R: =perpendicular
square	<code>\square</code>	□	(math) amssymb	/square B: =square

continued on next page...

ISO entity	$\LaTeX$ equivalent	Glyph	$\LaTeX$ requirements	Description
lang	<code>\langle</code>	$\langle$	(math)	/langle O: =left angle bracket
rang	<code>\rangle</code>	$\rangle$	(math)	/rangle C: =right angle bracket

## 2.2 Greek symbols (grk3)

ISO entity	$\LaTeX$ equivalent	Glyph	$\LaTeX$ requirements	Description
alpha	<code>\alpha</code>	$\alpha$	(math)	=small alpha, Greek
beta	<code>\beta</code>	$\beta$	(math)	=small beta, Greek
gamma	<code>\gamma</code>	$\gamma$	(math)	=small gamma, Greek
Gamma	<code>\Gamma</code>	$\Gamma$	(math)	=capital Gamma, Greek
gammad	<code>\digamma</code>	$\digamma$	(math) amssymb	/digamma
delta	<code>\delta</code>	$\delta$	(math)	=small delta, Greek
Delta	<code>\Delta</code>	$\Delta$	(math)	=capital Delta, Greek
epsi	<code>\epsilon</code>	$\epsilon$	(math)	=small epsilon, Greek
epsiv	<code>\varepsilon</code>	$\varepsilon$	(math)	/varepsilon
epsis	<code>\epsilon</code>	$\epsilon$	(math)	/straightepsilon
zeta	<code>\zeta</code>	$\zeta$	(math)	=small zeta, Greek
eta	<code>\eta</code>	$\eta$	(math)	=small eta, Greek
thetas	<code>\theta</code>	$\theta$	(math)	straight theta
Theta	<code>\Theta</code>	$\Theta$	(math)	=capital Theta, Greek
thetav	<code>\vartheta</code>	$\vartheta$	(math)	/vartheta - curly or open theta
iota	<code>\iota</code>	$\iota$	(math)	=small iota, Greek
kappa	<code>\kappa</code>	$\kappa$	(math)	=small kappa, Greek
kappav	<code>\varkappa</code>	$\varkappa$	(math) amssymb	/varkappa
lambda	<code>\lambda</code>	$\lambda$	(math)	=small lambda, Greek
Lambda	<code>\Lambda</code>	$\Lambda$	(math)	=capital Lambda, Greek
mu	<code>\mu</code>	$\mu$	(math)	=small mu, Greek
nu	<code>\nu</code>	$\nu$	(math)	=small nu, Greek
xi	<code>\xi</code>	$\xi$	(math)	=small xi, Greek
Xi	<code>\Xi</code>	$\Xi$	(math)	=capital Xi, Greek
pi	<code>\pi</code>	$\pi$	(math)	=small pi, Greek
piv	<code>\varpi</code>	$\varpi$	(math)	/varpi
Pi	<code>\Pi</code>	$\Pi$	(math)	=capital Pi, Greek
rho	<code>\rho</code>	$\rho$	(math)	=small rho, Greek
rhov	<code>\varrho</code>	$\varrho$	(math)	/varrho

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
sigma	<code>\sigma</code>	$\sigma$	(math)	=small sigma, Greek
Sigma	<code>\Sigma</code>	$\Sigma$	(math)	=capital Sigma, Greek
sigmav	<code>\varsigma</code>	$\varsigma$	(math)	/varsigma
tau	<code>\tau</code>	$\tau$	(math)	=small tau, Greek
upsi	<code>\upsilon</code>	$\upsilon$	(math)	=small upsilon, Greek
Upsi	<code>\Upsilon</code>	$\Upsilon$	(math)	=capital Upsilon, Greek
phis	<code>\phi</code>	$\phi$	(math)	/straightphi - straight phi
Phi	<code>\Phi</code>	$\Phi$	(math)	=capital Phi, Greek
phiv	<code>\varphi</code>	$\varphi$	(math)	/varphi - curly or open phi
chi	<code>\chi</code>	$\chi$	(math)	=small chi, Greek
psi	<code>\psi</code>	$\psi$	(math)	=small psi, Greek
Psi	<code>\Psi</code>	$\Psi$	(math)	=capital Psi, Greek
omega	<code>\omega</code>	$\omega$	(math)	=small omega, Greek
Omega	<code>\Omega</code>	$\Omega$	(math)	=capital Omega, Greek

### 2.3 Alternative greek symbols (grk4)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
b.alpha	<code>\alpha</code>	$\alpha$	(math) amsbsy	=small alpha, Greek
b.beta	<code>\beta</code>	$\beta$	(math) amsbsy	=small beta, Greek
b.gamma	<code>\gamma</code>	$\gamma$	(math) amsbsy	=small gamma, Greek
b.Gamma	<code>\Gamma</code>	$\Gamma$	(math) amsbsy	=capital Gamma, Greek
b.gammad	<code>\digamma</code>	$\digamma$	(math) amsbsy,amssymb	/digamma
b.delta	<code>\delta</code>	$\delta$	(math) amsbsy	=small delta, Greek
b.Delta	<code>\Delta</code>	$\Delta$	(math) amsbsy	=capital Delta, Greek
b.epsi	<code>\epsilon</code>	$\epsilon$	(math) amsbsy	=small epsilon, Greek
b.epsiv	<code>\varepsilon</code>	$\varepsilon$	(math) amsbsy	/varepsilon
b.opsis	<code>\epsilon</code>	$\epsilon$	(math) amsbsy	/straightepsilon
b.zeta	<code>\zeta</code>	$\zeta$	(math) amsbsy	=small zeta, Greek
b.eta	<code>\eta</code>	$\eta$	(math) amsbsy	=small eta, Greek
b.thetas	<code>\theta</code>	$\theta$	(math) amsbsy	straight theta
b.Theta	<code>\Theta</code>	$\Theta$	(math) amsbsy	=capital Theta, Greek
b.thetav	<code>\vartheta</code>	$\vartheta$	(math) amsbsy	/vartheta - curly or open theta
b.iota	<code>\iota</code>	$\iota$	(math) amsbsy	=small iota, Greek
b.kappa	<code>\kappa</code>	$\kappa$	(math) amsbsy	=small kappa, Greek

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
b.kappav	<code>\varkappa</code>	ϰ	(math) amsbsy,amssymb	/varkappa
b.lambda	<code>\lambda</code>	λ	(math) amsbsy	=small lambda, Greek
b.Lambda	<code>\Lambda</code>	Λ	(math) amsbsy	=capital Lambda, Greek
b.mu	<code>\mu</code>	μ	(math) amsbsy	=small mu, Greek
b.nu	<code>\nu</code>	ν	(math) amsbsy	=small nu, Greek
b.xi	<code>\xi</code>	ξ	(math) amsbsy	=small xi, Greek
b.Xi	<code>\Xi</code>	Ξ	(math) amsbsy	=capital Xi, Greek
b.pi	<code>\pi</code>	π	(math) amsbsy	=small pi, Greek
b.Pi	<code>\Pi</code>	Π	(math) amsbsy	=capital Pi, Greek
b.piv	<code>\varpi</code>	ϖ	(math) amsbsy	/varpi
b.rho	<code>\rho</code>	ρ	(math) amsbsy	=small rho, Greek
b.rhov	<code>\varrho</code>	ϱ	(math) amsbsy	/varrho
b.sigma	<code>\sigma</code>	σ	(math) amsbsy	=small sigma, Greek
b.Sigma	<code>\Sigma</code>	Σ	(math) amsbsy	=capital Sigma, Greek
b.sigmav	<code>\varsigma</code>	ς	(math) amsbsy	/varsigma
b.tau	<code>\tau</code>	τ	(math) amsbsy	=small tau, Greek
b.upsilon	<code>\upsilon</code>	υ	(math) amsbsy	=small upsilon, Greek
b.Upsilon	<code>\Upsilon</code>	Υ	(math) amsbsy	=capital Upsilon, Greek
b.phis	<code>\phi</code>	φ	(math) amsbsy	/straightphi - straight phi
b.Phi	<code>\Phi</code>	Φ	(math) amsbsy	=capital Phi, Greek
b.phiv	<code>\varphi</code>	φ	(math) amsbsy	/varphi - curly or open phi
b.chi	<code>\chi</code>	χ	(math) amsbsy	=small chi, Greek
b.psi	<code>\psi</code>	ψ	(math) amsbsy	=small psi, Greek
b.Psi	<code>\Psi</code>	Ψ	(math) amsbsy	=capital Psi, Greek
b.omega	<code>\omega</code>	ω	(math) amsbsy	=small omega, Greek
b.Omega	<code>\Omega</code>	Ω	(math) amsbsy	=capital Omega, Greek

## 2.4 Added Math Symbols: Ordinary (amso)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
ang	<code>\angle</code>	∠	(math)	/angle - angle
angmsd	<code>\measuredangle</code>	∠	(math) amssymb	/measuredangle - angle-measured
beth	<code>\beth</code>	beth	(math) amssymb	/beth - beth, Hebrew
bprime	<code>\backprime</code>	∕	(math) amssymb	/backprime - reverse prime
comp	<code>\complement</code>	∕	(math) amssymb	/complement - complement sign

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
daleth	<code>\daleth</code>	ד	(math) amssymb	/daleth - daleth, Hebrew
ell	<code>\ell</code>	ℓ	(math)	/ell - cursive small l
empty	<code>\emptyset</code>	∅	(math)	/emptyset /varnothing =small o, slash
gimel	<code>\gimel</code>	ג	(math) amssymb	/gimel - gimel, Hebrew
image	<code>\Im</code>	ℑ	(math)	/Im - imaginary
inodot	<code>\imath</code>	ι	(math)	/imath =small i, no dot
jnodot	<code>\jmath</code>	Ƶ	(math)	/jmath - small j, no dot
nexist	<code>\nexists</code>	∄	(math) amssymb	/nexists - negated exists
oS	<code>\circledS</code>	Ⓢ	(math) amssymb	/circledS - capital S in circle
planck	<code>\hbar</code>	ℏ	(math)	/hbar /hslash - Planck's over 2pi
real	<code>\Re</code>	ℜ	(math)	/Re - real
sbsol	<code>\smallsetminus</code>	∖	(math) amssymb	/sbs - short reverse solidus
vprime	<code>{'}</code>	'	(math)	/varprime - prime, variant
weierp	<code>\wp</code>	℘	(math)	/wp - Weierstrass p

## 2.5 Added Math Symbols: Binary operators (amsb)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
amalg	<code>\amalg</code>	∐	(math)	/amalg B: amalgamation or coproduct
Barwed	<code>\doublebarwedge</code>	⌘	(math) amssymb	/doublebarwedge B: log and, dbl bar
barwed	<code>\barwedge</code>	⌘	(math) amssymb	/barwedge B: logical and, bar above
Cap	<code>\Cap</code>	⋈	(math) amssymb	/Cap /doublecap B: dbl intersection
Cup	<code>\Cup</code>	⋈	(math) amssymb	/Cup /doublecup B: dbl union
cuvee	<code>\curlyvee</code>	⋈	(math) amssymb	/curlyvee B: curly logical or
cuwed	<code>\curlywedge</code>	⋈	(math) amssymb	/curlywedge B: curly logical and
diam	<code>\diamond</code>	◇	(math)	/diamond B: open diamond
divonx	<code>\divideontimes</code>	⋈	(math) amssymb	/divideontimes B: division on times
intcal	<code>\intercal</code>	⋈	(math) amssymb	/intercal B: intercal
lthree	<code>\leftthreetimes</code>	⋈	(math) amssymb	/leftthreetimes B:
ltimes	<code>\ltimes</code>	⋈	(math) amssymb	/ltimes B: times sign, left closed
minusb	<code>\boxminus</code>	⊞	(math) amssymb	/boxminus B: minus sign in box
oast	<code>\circledast</code>	⊛	(math) amssymb	/circledast B: asterisk in circle
ocir	<code>\circledcirc</code>	⊙	(math) amssymb	/circledcirc B: open dot in circle
odash	<code>\circleddash</code>	⊖	(math) amssymb	/circleddash B: hyphen in circle
odot	<code>\odot</code>	⊙	(math)	/odot B: middle dot in circle

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
ominus	<code>\ominus</code>	⊖	(math)	/ominus B: minus sign in circle
oplus	<code>\oplus</code>	⊕	(math)	/oplus B: plus sign in circle
osol	<code>\oslash</code>	⊘	(math)	/oslash B: solidus in circle
otimes	<code>\otimes</code>	⊗	(math)	/otimes B: multiply sign in circle
plusb	<code>\boxplus</code>	⊞	(math) amssymb	/boxplus B: plus sign in box
plusdo	<code>\dotplus</code>	⊕̇	(math) amssymb	/dotplus B: plus sign, dot above
rthree	<code>\rightthreetimes</code>	⋈	(math) amssymb	/rightthreetimes B:
rtimes	<code>\rtimes</code>	⋉	(math) amssymb	/rtimes B: times sign, right closed
sdot	<code>\cdot</code>	⋅	(math)	/cdot B: small middle dot
sdotb	<code>\boxdot</code>	⊠	(math) amssymb	/dotsquare /boxdot B: small dot in box
setmn	<code>\setminus</code>	∖	(math)	/setminus B: reverse solidus
sqcap	<code>\sqcap</code>	⊓	(math)	/sqcap B: square intersection
sqcup	<code>\sqcup</code>	⊔	(math)	/sqcup B: square union
ssetmn	<code>\smallsetminus</code>	∖	(math) amssymb	/smallsetminus B: sm reverse solidus
sstarf	<code>\star</code>	★	(math)	/star B: small star, filled
timesb	<code>\boxtimes</code>	⊠	(math) amssymb	/boxtimes B: multiply sign in box
top	<code>\top</code>	⊤	(math)	/top B: inverted perpendicular
uplus	<code>\uplus</code>	⊕	(math)	/uplus B: plus sign in union
wreath	<code>\wr</code>	⌘	(math)	/wr B: wreath product
xcirc	<code>\bigcirc</code>	◯	(math)	/bigcirc B: large circle
xdtri	<code>\bigtriangledown</code>	∇	(math)	/bigtriangledown B: big dn tri, open
xutri	<code>\bigtriangleup</code>	△	(math)	/bigtriangleup B: big up tri, open
coprod	<code>\coprod</code>	∏	(math)	/coprod L: coproduct operator
prod	<code>\prod</code>	∏	(math)	/prod L: product operator
sum	<code>\sum</code>	∑	(math)	/sum L: summation operator

## 2.6 Added Math Symbols: Relations (amstr)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
ape	<code>\approxeq</code>	≈	(math) amssymb	/approxeq R: approximate, equals
asympt	<code>\asymp</code>	≈	(math)	/asympt R: asymptotically equal to
bcong	<code>\backcong</code>	≍	(math) amssymb,local	/backcong R: reverse congruent
bepsi	<code>\backepsilon</code>	⋈	(math) amssymb	/backepsilon R: such that
bowtie	<code>\bowtie</code>	⋈	(math)	/bowtie R:
bsim	<code>\backsim</code>	≈	(math) amssymb	/backsim R: reverse similar

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
bsime	<code>\backsimeq</code>		(math) amssymb	/backsimeq R: reverse similar, eq
bump	<code>\Bumpeq</code>		(math) amssymb	/Bumpeq R: bumpy equals
bumpe	<code>\bumpeq</code>		(math) amssymb	/bumpeq R: bumpy equals, equals
cire	<code>\circeq</code>		(math) amssymb	/circeq R: circle, equals
colone	<code>:=</code>		(math)	/coloneq R: colon, equals
cuepr	<code>\curlyeqprec</code>		(math) amssymb	/curlyeqprec R: curly eq, precedes
cuesc	<code>\curlyeqsucc</code>		(math) amssymb	/curlyeqsucc R: curly eq, succeeds
cupre	<code>\preccurlyeq</code>		(math) amssymb	/curlypreceq R: curly precedes, eq
dashv	<code>\dashv</code>		(math)	/dashv R: dash, vertical
ecir	<code>\eqcirc</code>		(math) amssymb	/eqcirc R: circle on equals sign
ecolon	<code>=:</code>		(math)	/eqcolon R: equals, colon
eDot	<code>\doteqdot</code>		(math) amssymb	/doteqdot /Doteq R: eq, even dots
esdot	<code>\doteq</code>		(math)	/doteq R: equals, single dot above
efDot	<code>\fallingdotseq</code>		(math) amssymb	/fallingdotseq R: eq, falling dots
egs	<code>\eqslantgtr</code>		(math) amssymb	/eqslantgtr R: equal-or-gtr, slanted
els	<code>\eqslantless</code>		(math) amssymb	/eqslantless R: eq-or-less, slanted
erDot	<code>\risingdotseq</code>		(math) amssymb	/risingdotseq R: eq, rising dots
fork	<code>\pitchfork</code>		(math) amssymb	/pitchfork R: pitchfork
frown	<code>\frown</code>		(math)	/frown R: down curve
gap	<code>\gtrapprox</code>		(math) amssymb	/gtrapprox R: greater, approximate
gsdot	<code>\gtrdot</code>		(math) amssymb	/gtrdot R: greater than, single dot
gE	<code>\geqq</code>		(math) amssymb	/geqq R: greater, double equals
gel	<code>\gtreqless</code>		(math) amssymb	/gtreqless R: greater, equals, less
gEl	<code>\gtreqqless</code>		(math) amssymb	/gtreqqless R: gt, dbl equals, less
ges	<code>\geqslant</code>		(math) amssymb	/geqslant R: gt-or-equal, slanted
Gg	<code>\ggg</code>		(math) amssymb	/ggg /Gg /gggtr R: triple gtr-than
gl	<code>\gtrless</code>		(math) amssymb	/gtrless R: greater, less
gsim	<code>\gtrsim</code>		(math) amssymb	/gtrsim R: greater, similar
Gt	<code>\gg</code>		(math)	/gg R: dbl greater-than sign
lap	<code>\lessapprox</code>		(math) amssymb	/lessapprox R: less, approximate
ldot	<code>\lessdot</code>		(math) amssymb	/lessdot R: less than, with dot
lE	<code>\leqq</code>		(math) amssymb	/leqq R: less, double equals
lEg	<code>\lesseqgtr</code>		(math) amssymb	/lesseqgtr R: less, dbl eq, greater
leg	<code>\lesseqgtr</code>		(math) amssymb	/lesseqgtr R: less, eq, greater
les	<code>\leqslant</code>		(math) amssymb	/leqslant R: less-than-or-eq, slant
lg	<code>\lessgtr</code>		(math) amssymb	/lessgtr R: less, greater

continued on next page...



ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
Ll	<code>\lll</code>	≪≪	(math) amssymb	/Ll /lll /llless R: triple less-than
lsim	<code>\lesssim</code>	∩	(math) amssymb	/lesssim R: less, similar
Lt	<code>\ll</code>	≪	(math)	/ll R: double less-than sign
ltrie	<code>\trianglelefteq</code>	∇	(math) amssymb	/trianglelefteq R: left triangle, eq
mid	<code>\mid</code>	—	(math)	/mid R:
models	<code>\models</code>	⊨	(math)	/models R:
pr	<code>\prec</code>	⋪	(math)	/prec R: precedes
prap	<code>\precapprox</code>	⋪≈	(math) amssymb	/precapprox R: precedes, approximate
pre	<code>\preceq</code>	⋪=	(math)	/preceq R: precedes, equals
prsim	<code>\precsim</code>	⋪∩	(math) amssymb	/precsim R: precedes, similar
rtrie	<code>\trianglerighteq</code>	∇	(math) amssymb	/trianglerighteq R: right tri, eq
samalg	<code>\amalg</code>	∐	(math)	/smallamalg R: small amalg
sc	<code>\succ</code>	⋫	(math)	/succ R: succeeds
scap	<code>\succapprox</code>	⋫≈	(math) amssymb	/succapprox R: succeeds, approximate
scue	<code>\succcurlyeq</code>	⋫=	(math) amssymb	/succurlyeq R: succeeds, curly eq
sce	<code>\succeq</code>	⋫=	(math)	/succeq R: succeeds, equals
scsim	<code>\succsim</code>	⋫∩	(math) amssymb	/succsim R: succeeds, similar
sfrown	<code>\smallfrown</code>	∩	(math) amssymb	/smallfrown R: small down curve
smid	<code>\shortmid</code>	—	(math) amssymb	/shortmid R:
smile	<code>\smile</code>	∩	(math)	/smile R: up curve
spar	<code>\shortparallel</code>	∥	(math) amssymb	/shortparallel R: short parallel
sqsub	<code>\sqsubset</code>	⊏	(math) amssymb	/sqsubset R: square subset
sqsube	<code>\sqsubseteq</code>	⊏=	(math)	/sqsubseteq R: square subset, equals
sqsup	<code>\sqsupset</code>	⊐	(math) amssymb	/sqsupset R: square superset
sqsupe	<code>\sqsupseteq</code>	⊐=	(math)	/sqsupseteq R: square superset, eq
ssmile	<code>\smallsmile</code>	∩	(math) amssymb	/smallsmile R: small up curve
Sub	<code>\Subset</code>	⊆	(math) amssymb	/Subset R: double subset
subE	<code>\subseteqq</code>	⊆=	(math) amssymb	/subseteqq R: subset, dbl equals
Sup	<code>\Supset</code>	⊇	(math) amssymb	/Supset R: dbl superset
supE	<code>\supseteqq</code>	⊇=	(math) amssymb	/supseteqq R: superset, dbl equals
thkap	<code>\thickapprox</code>	≈	(math) amssymb	/thickapprox R: thick approximate
thksim	<code>\thicksim</code>	≈	(math) amssymb	/thicksim R: thick similar
trie	<code>\triangleq</code>	≐	(math) amssymb	/triangleq R: triangle, equals
twixt	<code>\between</code>	⋈	(math) amssymb	/between R: between
vdash	<code>\vdash</code>	⊥	(math)	/vdash R: vertical, dash
Vdash	<code>\Vdash</code>	⊥	(math) amssymb	/Vdash R: dbl vertical, dash
vDash	<code>\vDash</code>	⊥	(math) amssymb	/vDash R: vertical, dbl dash

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
veebar	<code>\veebar</code>	$\veebar$	(math) amssymb	/veebar R: logical or, bar below
vltri	<code>\vartriangleleft</code>	$\vartriangleleft$	(math) amssymb	/vartriangleleft R: l tri, open, var
vprop	<code>\varpropto</code>	$\varpropto$	(math) amssymb	/varpropto R: proportional, variant
vrtri	<code>\vartriangleright</code>	$\vartriangleright$	(math) amssymb	/vartriangleright R: r tri, open, var
Vvdash	<code>\Vvdash</code>	$\Vvdash$	(math) amssymb	/Vvdash R: triple vertical, dash



































## 2.7 Added Math Symbols: Negated relations (amsn)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
gnap	<code>\gnapprox</code>	$\gtrapprox$	(math) amssymb	/gnapprox N: greater, not approximate
gne	<code>\gneq</code>	$\gtrneq$	(math) amssymb	/gneq N: greater, not equals
gnE	<code>\gneqq</code>	$\gtrneqq$	(math) amssymb	/gneqq N: greater, not dbl equals
gnsim	<code>\gnsim</code>	$\gtrsim$	(math) amssymb	/gnsim N: greater, not similar
gvnE	<code>\gvertneqq</code>	$\gtrvertneqq$	(math) amssymb	/gvertneqq N: gt, vert, not dbl eq
lnap	<code>\lnapprox</code>	$\lesssim$	(math) amssymb	/lnapprox N: less, not approximate
lnE	<code>\lneqq</code>	$\lesseqgtr$	(math) amssymb	/lneqq N: less, not double equals
lne	<code>\lneq</code>	$\lesseqgtr$	(math) amssymb	/lneq N: less, not equals
lnsim	<code>\lnsim</code>	$\lesssim$	(math) amssymb	/lnsim N: less, not similar
lvnE	<code>\lvertneqq</code>	$\lesvertneqq$	(math) amssymb	/lvertneqq N: less, vert, not dbl eq
nap	<code>\not\approx</code>	$\not\approx$	(math)	/napprox N: not approximate
ncong	<code>\not\cong</code>	$\not\cong$	(math)	/ncong N: not congruent with
nequiv	<code>\not\equiv</code>	$\not\equiv$	(math)	/nequiv N: not identical with
ngE	<code>\ngeqq</code>	$\ngtrneqq$	(math) amssymb	/ngeqq N: not greater, dbl equals
nge	<code>\not\geq</code>	$\ngtr$	(math)	/ngeq N: not greater-than-or-equal
nges	<code>\ngeqslant</code>	$\ngtrslant$	(math) amssymb	/ngeqslant N: not gt-or-eq, slanted
ngt	<code>\not&gt;</code>	$\ngtr$	(math)	/ngtr N: not greater-than
nle	<code>\not\leq</code>	$\lesseqgtr$	(math)	/nleq N: not less-than-or-equal
nlE	<code>\nleqq</code>	$\lesseqgtr$	(math) amssymb	/nleqq N: not less, dbl equals
nles	<code>\nleqslant</code>	$\lesseqgtr$	(math) amssymb	/nleqslant N: not less-or-eq, slant
nlt	<code>\not&lt;</code>	$\lesseqgtr$	(math)	/nless N: not less-than
nltri	<code>\ntriangleleft</code>	$\ntriangleleft$	(math) amssymb	/ntriangleleft N: not left triangle
nltrie	<code>\ntrianglelefteq</code>	$\ntrianglelefteq$	(math) amssymb	/ntrianglelefteq N: not l tri, eq
nmid	<code>\nmid</code>	$\nmid$	(math) amssymb	/nmid
npar	<code>\nparallel</code>	$\nparallel$	(math) amssymb	/nparallel N: not parallel
npr	<code>\not\prec</code>	$\nprec$	(math)	/nprec N: not precedes

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
npre	<code>\not\preceq</code>		(math)	/npreceq N: not precedes, equals
nrtri	<code>\ntriangleright</code>		(math) amssymb	/ntriangleright N: not rt triangle
nrtrie	<code>\ntrianglerighteq</code>		(math) amssymb	/ntrianglerighteq N: not r tri, eq
nsc	<code>\not\succ</code>		(math)	/nsucc N: not succeeds
nsce	<code>\not\succeq</code>		(math)	/nsuceq N: not succeeds, equals
nsim	<code>\not\sim</code>		(math)	/nsim N: not similar
nsime	<code>\not\simeq</code>		(math)	/nsimeq N: not similar, equals
nsmid	<code>\nshortmid</code>		(math) amssymb	/nshortmid
nspar	<code>\nshortparallel</code>		(math) amssymb	/nshortparallel N: not short par
nsub	<code>\not\subset</code>		(math)	/nsubset N: not subset
nsube	<code>\not\subseteq</code>		(math)	/nsubseteq N: not subset, equals
nsubE	<code>\nsubseteqq</code>		(math) amssymb	/nsubseteqq N: not subset, dbl eq
nsup	<code>\not\supset</code>		(math)	/nsupset N: not superset
nsupE	<code>\nsupseteqq</code>		(math) amssymb	/nsupseteqq N: not superset, dbl eq
nsupe	<code>\not\supseteq</code>		(math)	/nsupseteq N: not superset, equals
nvdash	<code>\nvdash</code>		(math) amssymb	/nvdash N: not vertical, dash
nvDash	<code>\nvDash</code>		(math) amssymb	/nvDash N: not vertical, dbl dash
nVDash	<code>\nVDash</code>		(math) amssymb	/nVDash N: not dbl vert, dbl dash
nVdash	<code>\nVdash</code>		(math) amssymb	/nVdash N: not dbl vertical, dash
prnap	<code>\precnapprox</code>		(math) amssymb	/precnapprox N: precedes, not approx
prnE	<code>\precneqq</code>		(math) amssymb	/precneqq N: precedes, not dbl eq
prnsim	<code>\precnsim</code>		(math) amssymb	/precnsim N: precedes, not similar
scnap	<code>\succnapprox</code>		(math) amssymb	/succnapprox N: succeeds, not approx
scnE	<code>\succneqq</code>		(math) amssymb	/succneqq N: succeeds, not dbl eq
sensim	<code>\succnsim</code>		(math) amssymb	/succnsim N: succeeds, not similar
subne	<code>\subsetneq</code>		(math) amssymb	/subsetneq N: subset, not equals
subnE	<code>\subsetneqq</code>		(math) amssymb	/subsetneqq N: subset, not dbl eq
supne	<code>\supsetneq</code>		(math) amssymb	/supsetneq N: superset, not equals
supnE	<code>\supsetneqq</code>		(math) amssymb	/supsetneqq N: superset, not dbl eq
vsubnE	<code>\varsubsetneqq</code>		(math) amssymb	/varsubsetneqq N: subset not dbl eq, var
vsubne	<code>\varsubsetneq</code>		(math) amssymb	/varsubsetneq N: subset, not eq, var
vsupne	<code>\varsupsetneq</code>		(math) amssymb	/varsupsetneq N: superset, not eq, var
vsupnE	<code>\varsupsetneqq</code>		(math) amssymb	/varsupsetneqq N: super not dbl eq, var

## 2.8 Added Math Symbols: Arrow Relations (amsa)

ISO entity	$\LaTeX$ equivalent	Glyph	$\LaTeX$ requirements	Description
cularr	<code>\curvearrowleft</code>		(math) amssymb	/curvearrowleft A: left curved arrow
curarr	<code>\curvearrowright</code>		(math) amssymb	/curvearrowright A: rt curved arrow
dArr	<code>\Downarrow</code>		(math)	/Downarrow A: down dbl arrow
darr2	<code>\downdownarrows</code>		(math) amssymb	/downdownarrows A: two down arrows
dharl	<code>\downharpoonleft</code>		(math) amssymb	/downleftharpoon A: dn harpoon-left
dharr	<code>\downharpoonright</code>		(math) amssymb	/downrightharpoon A: down harpoon-rt
lAarr	<code>\Lleftarrow</code>		(math) amssymb	/Lleftarrow A: left triple arrow
Larr	<code>\twoheadleftarrow</code>		(math) amssymb	/twoheadleftarrow A:
larr2	<code>\leftleftarrows</code>		(math) amssymb	/leftleftarrows A: two left arrows
larrhk	<code>\hookleftarrow</code>		(math)	/hookleftarrow A: left arrow-hooked
larrlp	<code>\looparrowleft</code>		(math) amssymb	/looparrowleft A: left arrow-looped
larrtl	<code>\leftarrowtail</code>		(math) amssymb	/leftarrowtail A: left arrow-tailed
lhard	<code>\leftharpoondown</code>		(math)	/leftharpoondown A: l harpoon-down
lharu	<code>\leftharpoonup</code>		(math)	/leftharpoonup A: left harpoon-up
hArr	<code>\Leftrightarrow</code>		(math)	/Leftrightarrow A: l&r dbl arrow
harr	<code>\leftrightharpoon</code>		(math)	/leftrightharpoon A: l&r arrow
lrarr2	<code>\leftrightharpoons</code>		(math) amssymb	/leftrightharpoons A: l arr over r arr
rlarr2	<code>\rightleftarrows</code>		(math) amssymb	/rightleftarrows A: r arr over l arr
harrw	<code>\leftrightsquigarrow</code>		(math) amssymb	/leftrightsquigarrow A: l&r arr-wavy
rlhar2	<code>\rightleftharpoons</code>		(math)	/rightleftharpoons A: r harp over l
lrhar2	<code>\leftrightharpoons</code>		(math) amssymb	/leftrightharpoons A: l harp over r
lsh	<code>\Lsh</code>		(math) amssymb	/Lsh A:
map	<code>\mapsto</code>		(math)	/mapsto A:
mumap	<code>\multimap</code>		(math) amssymb	/multimap A:
nearr	<code>\nearrow</code>		(math)	/nearrow A: NE pointing arrow
nlArr	<code>\nLeftarrow</code>		(math) amssymb	/nLeftarrow A: not implied by
nlarr	<code>\nleftarrow</code>		(math) amssymb	/nleftarrow A: not left arrow
nhArr	<code>\nLeftrightarrow</code>		(math) amssymb	/nLeftrightarrow A: not l&r dbl arr
nharr	<code>\nleftrightharpoon</code>		(math) amssymb	/nleftrightharpoon A: not l&r arrow
nrarr	<code>\nrightarrow</code>		(math) amssymb	/nrightarrow A: not right arrow
nrArr	<code>\nRightarrow</code>		(math) amssymb	/nRightarrow A: not implies
nwarr	<code>\nwarrow</code>		(math)	/nwarrow A: NW pointing arrow
olarr	<code>\circlearrowleft</code>		(math) amssymb	/circlearrowleft A: l arr in circle
orarr	<code>\circlearrowright</code>		(math) amssymb	/circlearrowright A: r arr in circle

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
rAarr	\Rrightarrow	⇒	(math) amssymb	/Rrightarrow A: right triple arrow
Rarr	\twoheadrightarrow	→	(math) amssymb	/twoheadrightarrow A:
rarr2	\rightrightarrows	⇔	(math) amssymb	/rightrightarrows A: two rt arrows
rarrhk	\hookrightarrow	↪	(math)	/hookrightarrow A: rt arrow-hooked
rarrlp	\looparrowright	↻	(math) amssymb	/looparrowright A: rt arrow-looped
rarrtl	\rightarrowtail	↗	(math) amssymb	/rightarrowtail A: rt arrow-tailed
rarrw	\rightsquigarrow	↗	(math) amssymb	/squigarrowright A: rt arrow-wavy
rhard	\rightharpoondown	↘	(math)	/rightharpoondown A: rt harpoon-down
rharu	\rightharpoonup	↗	(math)	/rightharpoonup A: rt harpoon-up
rsh	\Rsh	↗	(math) amssymb	/Rsh A:
drarr	\searrow	↘	(math)	/searrow A: downward rt arrow
dlarr	\swarrow	↙	(math)	/swarrow A: downward l arrow
uArr	\Uparrow	↑	(math)	/Uparrow A: up dbl arrow
uarr2	\uparrows	↑↑	(math) amssymb	/uparrows A: two up arrows
vArr	\Updownarrow	↕	(math)	/Updownarrow A: up&down dbl arrow
varr	\updownarrow	↕	(math)	/updownarrow A: up&down arrow
uharl	\upharpoonleft	↖	(math) amssymb	/upharpoonleft A: up harpoon-left
uharr	\upharpoonright	↗	(math) amssymb	/upharpoonright A: up harp-r
xlArr	\Longleftarrow	⇐	(math)	/Longleftarrow A: long l dbl arrow
xhArr	\Leftrightarrow	⇔	(math)	/Leftrightarrow A: long l&r dbl arr
xharr	\longleftarrow	⇐	(math)	/longleftarrow A: long l&r arr
xrArr	\longrightarrow	⇒	(math)	/longrightarrow A: long rt dbl arr

## 2.9 Added Math Symbols: Delimiters (amsc)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
rceil	\rceil	⌋	(math)	/rceil C: right ceiling
rfloor	\rfloor	⌋	(math)	/rfloor C: right floor
rpargt	\rlap{>}\,)	⌋	(math)	/rightparengtr C: right paren, gt
urcorn	\urcorner	⌋	(math) amsfonts	/urcorner C: upper right corner
drcorn	\lrcorner	⌋	(math) amsfonts	/lrcorner C: downward right corner
lceil	\lceil	⌈	(math)	/lceil O: left ceiling
lfloor	\lfloor	⌈	(math)	/lfloor O: left floor
lpargt	\rlap{>}\,)	⌋	(math)	/leftparengtr O: left parenthesis, gt
ulcorn	\ulcorner	⌈	(math) amsfonts	/ulcorner O: upper left corner
dlcorn	\llcorner	⌈	(math) amsfonts	/llcorner O: downward left corner

## 3 Latin based alphabets

### 3.1 Added Latin 1 (lat1)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
aacute	\' {a}	á		=small a, acute accent
Aacute	\' {A}	Á		=capital A, acute accent
acirc	\^ {a}	â		=small a, circumflex accent
Acirc	\^ {A}	Â		=capital A, circumflex accent
agrave	\' {a}	à		=small a, grave accent
Agrave	\' {A}	À		=capital A, grave accent
aring	\aa	å		=small a, ring
Aring	\AA	Å		=capital A, ring
atilde	\~ {a}	ã		=small a, tilde
Atilde	\~ {A}	Ã		=capital A, tilde
auml	\" {a}	ä		=small a, dieresis or umlaut mark
Auml	\" {A}	Ä		=capital A, dieresis or umlaut mark
aelig	\ae	æ		=small ae diphthong (ligature)
AElig	\AE	Æ		=capital AE diphthong (ligature)
ccedil	\c {c}	ç		=small c, cedilla
Ccedil	\c {C}	Ç		=capital C, cedilla
eth	\dh	ð	fontenc:T1	=small eth, Icelandic
ETH	\DH	Ð	fontenc:T1	=capital Eth, Icelandic
eacute	\' {e}	é		=small e, acute accent
Eacute	\' {E}	É		=capital E, acute accent
ecirc	\^ {e}	ê		=small e, circumflex accent
Ecirc	\^ {E}	Ê		=capital E, circumflex accent
egrave	\' {e}	è		=small e, grave accent
Egrave	\' {E}	È		=capital E, grave accent
euml	\" {e}	ë		=small e, dieresis or umlaut mark
Euml	\" {E}	Ë		=capital E, dieresis or umlaut mark
iacute	\' {\i}	í		=small i, acute accent
Iacute	\' {I}	Í		=capital I, acute accent
icirc	\^ {\i}	î		=small i, circumflex accent
Icirc	\^ {I}	Î		=capital I, circumflex accent
igrave	\' {\i}	ì		=small i, grave accent

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
Igrave	<code>\' {I}</code>	Ì		=capital I, grave accent
iuml	<code>\" {i}</code>	ï		=small i, dieresis or umlaut mark
Iuml	<code>\" {I}</code>	Ï		=capital I, dieresis or umlaut mark
ntilde	<code>\~ {n}</code>	ñ		=small n, tilde
Ntilde	<code>\~ {N}</code>	Ñ		=capital N, tilde
oacute	<code>\' {o}</code>	ó		=small o, acute accent
Oacute	<code>\' {O}</code>	Ó		=capital O, acute accent
ocirc	<code>\^ {o}</code>	ô		=small o, circumflex accent
Ocirc	<code>\^ {O}</code>	Ô		=capital O, circumflex accent
ograve	<code>\' {o}</code>	ò		=small o, grave accent
Ograve	<code>\' {O}</code>	Ò		=capital O, grave accent
oslash	<code>\o</code>	ø		=small o, slash
Oslash	<code>\O</code>	Ø		=capital O, slash
otilde	<code>\~ {o}</code>	õ		=small o, tilde
Otilde	<code>\~ {O}</code>	Õ		=capital O, tilde
ouml	<code>\" {o}</code>	ö		=small o, dieresis or umlaut mark
Ouml	<code>\" {O}</code>	Ö		=capital O, dieresis or umlaut mark
szlig	<code>\ss</code>	ß		=small sharp s, German (sz ligature)
thorn	<code>\th</code>	þ	fontenc:T1	=small thorn, Icelandic
THORN	<code>\TH</code>	Þ	fontenc:T1	=capital THORN, Icelandic
uacute	<code>\' {u}</code>	ú		=small u, acute accent
Uacute	<code>\' {U}</code>	Ú		=capital U, acute accent
ucirc	<code>\^ {u}</code>	û		=small u, circumflex accent
Ucirc	<code>\^ {U}</code>	Û		=capital U, circumflex accent
ugrave	<code>\' {u}</code>	ù		=small u, grave accent
Ugrave	<code>\' {U}</code>	Ù		=capital U, grave accent
uuml	<code>\" {u}</code>	ü		=small u, dieresis or umlaut mark
Uuml	<code>\" {U}</code>	Ü		=capital U, dieresis or umlaut mark
yacute	<code>\' {y}</code>	ý		=small y, acute accent
Yacute	<code>\' {Y}</code>	Ý		=capital Y, acute accent
yuml	<code>\" {y}</code>	ÿ		=small y, dieresis or umlaut mark

### 3.2 Added Latin 2 (lat2)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
abreve	<code>\u{a}</code>	ă		=small a, breve
Abreve	<code>\u{A}</code>	Ă		=capital A, breve
amacr	<code>\={a}</code>	ā		=small a, macron
Amacr	<code>\={A}</code>	Ā		=capital A, macron
aogon	<code>\k{a}</code>	ą	fontenc:T1	=small a, ogonek
Aogon	<code>\k{A}</code>	Ą	fontenc:T1	=capital A, ogonek
acute	<code>\' {c}</code>	ć		=small c, acute accent
Cacute	<code>\' {C}</code>	Ć		=capital C, acute accent
ccaron	<code>\v{c}</code>	č		=small c, caron
Ccaron	<code>\v{C}</code>	Č		=capital C, caron
ccirc	<code>\^ {c}</code>	ĉ		=small c, circumflex accent
Ccirc	<code>\^ {C}</code>	Ĉ		=capital C, circumflex accent
cdot	<code>\. {c}</code>	ċ		=small c, dot above
Cdot	<code>\. {C}</code>	Ĉ		=capital C, dot above
dcaron	<code>\v{d}</code>	ď		=small d, caron
Dcaron	<code>\v{D}</code>	Ď		=capital D, caron
dstrok	<code>\dj</code>	đ	fontenc:T1	=small d, stroke
Dstrok	<code>\DJ</code>	Đ	fontenc:T1	=capital D, stroke
ecaron	<code>\v{e}</code>	ě		=small e, caron
Ecaron	<code>\v{E}</code>	Ě		=capital E, caron
edot	<code>\. {e}</code>	ė		=small e, dot above
Edot	<code>\. {E}</code>	Ė		=capital E, dot above
emacr	<code>\={e}</code>	ē		=small e, macron
Emacr	<code>\={E}</code>	Ē		=capital E, macron
eogon	<code>\k{e}</code>	ę	fontenc:T1	=small e, ogonek
Eogon	<code>\k{E}</code>	Ę	fontenc:T1	=capital E, ogonek
gacute	<code>\' {g}</code>	g'		=small g, acute accent
gbreve	<code>\u{g}</code>	ğ		=small g, breve
Gbreve	<code>\u{G}</code>	Ğ		=capital G, breve
Gcedil	<code>\c{G}</code>	Ġ		=capital G, cedilla
gcirc	<code>\^ {g}</code>	ġ		=small g, circumflex accent
Gcirc	<code>\^ {G}</code>	Ĝ		=capital G, circumflex accent
gdot	<code>\. {g}</code>	ġ		=small g, dot above

continued on next page...



ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
Gdot	\.{G}	Ġ		=capital G, dot above
hcirc	\^{h}	ĥ		=small h, circumflex accent
Hcirc	\^{H}	Ĥ		=capital H, circumflex accent
hstrok	\hstrok	ħ	local	=small h, stroke
Hstrok	\Hstrok	Ħ	local	=capital H, stroke
Idot	\.{I}	İ		=capital I, dot above
Imacr	\={I}	Ī		=capital I, macron
imacr	\={\i}	ī		=small i, macron
ijlig	i\kern -.15em j	ij		=small ij ligature
IJlig	I\kern -.15em J	IJ		=capital IJ ligature
inodot	\i	ı		=small i without dot
iogon	\k{i}	ı̇	fontenc:T1	=small i, ogonek
Iogon	\k{I}	Ĭ	fontenc:T1	=capital I, ogonek
itilde	\~{\i}	ĩ		=small i, tilde
Itilde	\~{I}	Ĩ		=capital I, tilde
jcirc	\^{j}	ĵ		=small j, circumflex accent
Jcirc	\^{J}	Ĵ		=capital J, circumflex accent
kcedil	\c{k}	ķ		=small k, cedilla
Kcedil	\c{K}	Ķ		=capital K, cedilla
kgreen	\textsc{k}	ƙ		=small k, Greenlandic
lacute	\' {l}	ĺ		=small l, acute accent
Lacute	\' {L}	Ł		=capital L, acute accent
lcaron	\v{l}	ľ		=small l, caron
Lcaron	\v{L}	Ľ		=capital L, caron
lcedil	\c{l}	ļ		=small l, cedilla
Lcedil	\c{L}	Ļ		=capital L, cedilla
lmidot	\lmidot	ł	local	=small l, middle dot
Lmidot	\Lmidot	Ł	local	=capital L, middle dot
lstrok	\l	ł̣		=small l, stroke
Lstrok	\L	Ł̣		=capital L, stroke
nacute	\' {n}	ń		=small n, acute accent
Nacute	\' {N}	Ń		=capital N, acute accent
eng	\ng	ŋ	fontenc:T1	=small eng, Lapp
ENG	\NG	Ŋ	fontenc:T1	=capital ENG, Lapp
napos	n\kern-.2em\textsf{'}	ñ		=small n, apostrophe

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
ncaron	<code>\v{n}</code>	ñ		=small n, caron
Ncaron	<code>\v{N}</code>	Ñ		=capital N, caron
ncedil	<code>\c{n}</code>	ñ		=small n, cedilla
Ncedil	<code>\c{N}</code>	Ñ		=capital N, cedilla
odblac	<code>\H{o}</code>	ö		=small o, double acute accent
Odblac	<code>\H{O}</code>	Ö		=capital O, double acute accent
Omacr	<code>\={O}</code>	Ō		=capital O, macron
omacr	<code>\={o}</code>	ō		=small o, macron
oelig	<code>\oe</code>	œ		=small oe ligature
OElig	<code>\OE</code>	Œ		=capital OE ligature
racute	<code>\' {r}</code>	í		=small r, acute accent
Racute	<code>\' {R}</code>	Ŕ		=capital R, acute accent
rcaron	<code>\v{r}</code>	ř		=small r, caron
Rcaron	<code>\v{R}</code>	Ř		=capital R, caron
rcedil	<code>\c{r}</code>	ɾ		=small r, cedilla
Rcedil	<code>\c{R}</code>	Ŗ		=capital R, cedilla
sacute	<code>\' {s}</code>	ś		=small s, acute accent
Sacute	<code>\' {S}</code>	Ś		=capital S, acute accent
scaron	<code>\v{s}</code>	š		=small s, caron
Scaron	<code>\v{S}</code>	Š		=capital S, caron
scedil	<code>\c{s}</code>	ş		=small s, cedilla
Scedil	<code>\c{S}</code>	Ș		=capital S, cedilla
scirc	<code>\~{s}</code>	š		=small s, circumflex accent
Scirc	<code>\~{S}</code>	Ŝ		=capital S, circumflex accent
tcaron	<code>\v{t}</code>	ť		=small t, caron
Tcaron	<code>\v{T}</code>	Ť		=capital T, caron
tcedil	<code>\c{t}</code>	ţ		=small t, cedilla
Tcedil	<code>\c{T}</code>	Ț		=capital T, cedilla
tstrok	<code>\tstrok</code>	‡	local	=small t, stroke
Tstrok	<code>\Tstrok</code>	‡	local	=capital T, stroke
ubreve	<code>\u{u}</code>	ü		=small u, breve
Ubreve	<code>\u{U}</code>	Û		=capital U, breve
udblac	<code>\H{u}</code>	ü		=small u, double acute accent
Udblac	<code>\H{U}</code>	Û		=capital U, double acute accent
umacr	<code>\={u}</code>	ū		=small u, macron
Umacr	<code>\={U}</code>	Ū		=capital U, macron

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
uogon	<code>\k{u}</code>	ϋ	fontenc:T1	=small u, ogonek
Uogon	<code>\k{U}</code>	Ū	fontenc:T1	=capital U, ogonek
uring	<code>\r{u}</code>	ũ		=small u, ring
Uring	<code>\r{U}</code>	Ū		=capital U, ring
utilde	<code>\~{u}</code>	ũ		=small u, tilde
Utilde	<code>\~{U}</code>	Ū		=capital U, tilde
wcirc	<code>\^{w}</code>	ŵ		=small w, circumflex accent
Wcirc	<code>\^{W}</code>	Ŵ		=capital W, circumflex accent
ycirc	<code>\^{y}</code>	ÿ		=small y, circumflex accent
Ycirc	<code>\^{Y}</code>	ÿ		=capital Y, circumflex accent
Yuml	<code>\"Y}</code>	ÿ		=capital Y, dieresis or umlaut mark
zacute	<code>\'z}</code>	ẏ		=small z, acute accent
Zacute	<code>\'Z}</code>	Ẑ		=capital Z, acute accent
zcaron	<code>\v{z}</code>	ž		=small z, caron
Zcaron	<code>\v{Z}</code>	Ž		=capital Z, caron
zdot	<code>\.{z}</code>	ẏ		=small z, dot above
Zdot	<code>\.{Z}</code>	Ẑ		=capital Z, dot above

### 3.3 Diacritical marks (dia)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
acute	<code>\'{} </code>	´		=acute accent
breve	<code>\u{} </code>	˘		=breve
caron	<code>\v{} </code>	ˇ		=caron
cedil	<code>\c{} </code>	¸		=cedilla
circ	<code>\^{} </code>	ˆ		=circumflex accent
dblac	<code>\H{} </code>	ˆˆ		=double acute accent
die	<code>\"{} </code>	¨		=diereis
uml	<code>\"{} </code>	¨		=umlaut mark
dot	<code>\.{} </code>	˙		=dot above
grave	<code>\'{} </code>	˘		=grave accent
macr	<code>\={}</code>	¯		=macron
ogon	<code>\k{} </code>	˛	fontenc:T1	=ogonek
ring	<code>\r{} </code>	˚		=ring
tilde	<code>\~{} </code>	˜		=tilde

## 4 Non-Latin based alphabets

### 4.1 Russian Cyrillic (cyr1)

ISO entity	$\LaTeX$ equivalent	Glyph	$\LaTeX$ requirements	Description
acy	a	а	cyracc,amsfonts	=small a, Cyrillic
Acy	A	А	cyracc,amsfonts	=capital A, Cyrillic
bcy	b	б	cyracc,amsfonts	=small be, Cyrillic
Bcy	B	Б	cyracc,amsfonts	=capital BE, Cyrillic
vcy	v	в	cyracc,amsfonts	=small ve, Cyrillic
Vcy	V	В	cyracc,amsfonts	=capital VE, Cyrillic
gcy	g	г	cyracc,amsfonts	=small ghe, Cyrillic
Gcy	G	Г	cyracc,amsfonts	=capital GHE, Cyrillic
dcy	d	д	cyracc,amsfonts	=small de, Cyrillic
Dcy	D	Д	cyracc,amsfonts	=capital DE, Cyrillic
iecy	e	е	cyracc,amsfonts	=small ie, Cyrillic
IEcy	E	Е	cyracc,amsfonts	=capital IE, Cyrillic
iocy	\{"e}	ё	cyracc,amsfonts	=small io, Russian
IOcy	\{"E}	Ё	cyracc,amsfonts	=capital IO, Russian
zhcy	zh	ж	cyracc,amsfonts	=small zhe, Cyrillic
ZHcy	Zh	Ж	cyracc,amsfonts	=capital ZHE, Cyrillic
zey	z	з	cyracc,amsfonts	=small ze, Cyrillic
Zcy	Z	З	cyracc,amsfonts	=capital ZE, Cyrillic
icy	i	и	cyracc,amsfonts	=small i, Cyrillic
Icy	I	И	cyracc,amsfonts	=capital I, Cyrillic
jcy	\u{i}	й	cyracc,amsfonts	=small short i, Cyrillic
Jcy	\u{I}	Й	cyracc,amsfonts	=capital short I, Cyrillic
key	k	к	cyracc,amsfonts	=small ka, Cyrillic
Kcy	K	К	cyracc,amsfonts	=capital KA, Cyrillic
lcy	l	л	cyracc,amsfonts	=small el, Cyrillic
Lcy	L	Л	cyracc,amsfonts	=capital EL, Cyrillic
mcy	m	м	cyracc,amsfonts	=small em, Cyrillic
Mcy	M	М	cyracc,amsfonts	=capital EM, Cyrillic
ncy	n	н	cyracc,amsfonts	=small en, Cyrillic
Ncy	N	Н	cyracc,amsfonts	=capital EN, Cyrillic
ocy	o	о	cyracc,amsfonts	=small o, Cyrillic
Ocy	O	О	cyracc,amsfonts	=capital O, Cyrillic

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
pcy	p	п	cyracc,amsfonts	=small pe, Cyrillic
Pcy	P	П	cyracc,amsfonts	=capital PE, Cyrillic
rcy	r	р	cyracc,amsfonts	=small er, Cyrillic
Rcy	R	Р	cyracc,amsfonts	=capital ER, Cyrillic
scy	s	с	cyracc,amsfonts	=small es, Cyrillic
Scy	S	С	cyracc,amsfonts	=capital ES, Cyrillic
tcy	t	т	cyracc,amsfonts	=small te, Cyrillic
Tcy	T	Т	cyracc,amsfonts	=capital TE, Cyrillic
ucy	u	у	cyracc,amsfonts	=small u, Cyrillic
Ucy	U	У	cyracc,amsfonts	=capital U, Cyrillic
fcy	f	ф	cyracc,amsfonts	=small ef, Cyrillic
Fcy	F	Ф	cyracc,amsfonts	=capital EF, Cyrillic
hkey	kh	х	cyracc,amsfonts	=small ha, Cyrillic
KHcy	Kh	Х	cyracc,amsfonts	=capital HA, Cyrillic
tscy	ts	ц	cyracc,amsfonts	=small tse, Cyrillic
TScy	Ts	Ц	cyracc,amsfonts	=capital TSE, Cyrillic
chcy	ch	ч	cyracc,amsfonts	=small che, Cyrillic
CHcy	Ch	Ч	cyracc,amsfonts	=capital CHE, Cyrillic
shcy	sh	ш	cyracc,amsfonts	=small sha, Cyrillic
SHcy	Sh	Ш	cyracc,amsfonts	=capital SHA, Cyrillic
shchcy	shch	щ	cyracc,amsfonts	=small shcha, Cyrillic
SHCHcy	Shch	Щ	cyracc,amsfonts	=capital SHCHA, Cyrillic
hardcy	\cdprime	ъ	cyracc,amsfonts	=small hard sign, Cyrillic
HARDcy	\Cdprime	Ъ	cyracc,amsfonts	=capital HARD sign, Cyrillic
ycy	y	ы	cyracc,amsfonts	=small yeru, Cyrillic
Ycy	Y	Ы	cyracc,amsfonts	=capital YERU, Cyrillic
softcy	\cprime	ь	cyracc,amsfonts	=small soft sign, Cyrillic
SOFTcy	\Cprime	Ь	cyracc,amsfonts	=capital SOFT sign, Cyrillic
ecy	\‘{e}	э	cyracc,amsfonts	=small e, Cyrillic
Ecy	\‘{E}	Э	cyracc,amsfonts	=capital E, Cyrillic
yucy	yu	ю	cyracc,amsfonts	=small yu, Cyrillic
YUcy	Yu	Ю	cyracc,amsfonts	=capital YU, Cyrillic
yacy	ya	я	cyracc,amsfonts	=small ya, Cyrillic
YAcy	Ya	Я	cyracc,amsfonts	=capital YA, Cyrillic
numero	NO	№	cyracc,amsfonts	=numero sign

## 4.2 Non-Russian Cyrillic (cyr2)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
djcy	dj	ђ	cyracc,amsfonts	=small dje, Serbian
DJcy	Dj	Ђ	cyracc,amsfonts	=capital DJE, Serbian
gjcy	\'g	ѓ	cyracc,amsfonts	=small gje, Macedonian
GJcy	\'G	Ѓ	cyracc,amsfonts	=capital GJE Macedonian
jukey	\=e	є	cyracc,amsfonts	=small je, Ukrainian
Jukey	\=E	Є	cyracc,amsfonts	=capital JE, Ukrainian
dscy	\dz	ѕ	cyracc,amsfonts	=small dse, Macedonian
DScy	\Dz	Ѕ	cyracc,amsfonts	=capital DSE, Macedonian
iukcy	\=\i	і	cyracc,amsfonts	=small i, Ukrainian
Iukcy	\=I	І	cyracc,amsfonts	=capital I, Ukrainian
yicy	\"\i	ї	cyracc,amsfonts	=small yi, Ukrainian
YIcy	\"I	Ї	cyracc,amsfonts	=capital YI, Ukrainian
jsercy	j	ј	cyracc,amsfonts	=small je, Serbian
Jsercy	J	Ј	cyracc,amsfonts	=capital JE, Serbian
ljcy	lj	љ	cyracc,amsfonts	=small lje, Serbian
LJcy	Lj	Љ	cyracc,amsfonts	=capital LJE, Serbian
njcy	nj	њ	cyracc,amsfonts	=small nje, Serbian
NJcy	Nj	Њ	cyracc,amsfonts	=capital NJE, Serbian
tshcy	\'c	ћ	cyracc,amsfonts	=small tshe, Serbian
TSHcy	\'C	Ћ	cyracc,amsfonts	=capital TSHE, Serbian
kjcy	\'k	ќ	cyracc,amsfonts	=small kje Macedonian
KJcy	\'K	Ќ	cyracc,amsfonts	=capital KJE, Macedonian
ubrcy	\u{u}	ў	cyracc,amsfonts	=small u, Byelorussian
Ubrcy	\u{U}	Ў	cyracc,amsfonts	=capital U, Byelorussian
dzcy	\dzh	џ	cyracc,amsfonts	=small dze, Serbian
DZcy	\Dzh	Џ	cyracc,amsfonts	=capital dze, Serbian

## 4.3 Greek letters (grk1)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
agr	a	α	grg	=small alpha, Greek
Agr	A	Α		=capital Alpha, Greek

**continued on next page. . .**

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
bgr	b	β	rgr	=small beta, Greek
Bgr	B	Β		=capital Beta, Greek
ggr	g	γ	rgr	=small gamma, Greek
Ggr	G	Γ	rgr	=capital Gamma, Greek
dgr	d	δ	rgr	=small delta, Greek
Dgr	D	Δ	rgr	=capital Delta, Greek
egr	e	ε	rgr	=small epsilon, Greek
Egr	E	Ε		=capital Epsilon, Greek
zgr	z	ζ	rgr	=small zeta, Greek
Zgr	Z	Ζ		=capital Zeta, Greek
eegr	h	η	rgr	=small eta, Greek
EEgr	H	Η		=capital Eta, Greek
thgr	j	θ	rgr	=small theta, Greek
THgr	J	Θ	rgr	=capital Theta, Greek
igr	i	ι	rgr	=small iota, Greek
Igr	I	Ι		=capital Iota, Greek
kgr	k	κ	rgr	=small kappa, Greek
Kgr	K	Κ		=capital Kappa, Greek
lgr	l	λ	rgr	=small lambda, Greek
Lgr	L	Λ	rgr	=capital Lambda, Greek
mgr	m	μ	rgr	=small mu, Greek
Mgr	M	Μ		=capital Mu, Greek
ngr	n	ν	rgr	=small nu, Greek
Ngr	N	Ν		=capital Nu, Greek
xgr	x	ξ	rgr	=small xi, Greek
Xgr	X	Ξ	rgr	=capital Xi, Greek
ogr	o	ο		=small omicron, Greek
Ogr	O	Ο		=capital Omicron, Greek
pgr	p	π	rgr	=small pi, Greek
Pgr	P	Π	rgr	=capital Pi, Greek
rgr	r	ρ	rgr	=small rho, Greek
Rgr	P	Ρ		=capital Rho, Greek
sgr	s	σ	rgr	=small sigma, Greek
Sgr	S	Σ	rgr	=capital Sigma, Greek
sfgr	c	ς	rgr	=final small sigma, Greek
tgr	t	τ	rgr	=small tau, Greek
Tgr	T	Τ		=capital Tau, Greek

continued on next page...

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
ugr	u	υ	rgr	=small upsilon, Greek
Ugr	Y	Υ		=capital Upsilon, Greek
phgr	f	φ	rgr	=small phi, Greek
PHgr	F	Φ	rgr	=capital Phi, Greek
khgr	q	χ	rgr	=small chi, Greek
KHgr	X	Χ		=capital Chi, Greek
psgr	y	ψ	rgr	=small psi, Greek
PSgr	Y	Ψ	rgr	=capital Psi, Greek
ohgr	w	ω	rgr	=small omega, Greek
OHgr	W	Ω	rgr	=capital Omega, Greek

#### 4.4 Monotoniko Greek (grk2)

ISO entity	L <sup>A</sup> T <sub>E</sub> X equivalent	Glyph	L <sup>A</sup> T <sub>E</sub> X requirements	Description
aacgr	'a	ά	rgr	=small alpha, accent, Greek
Aacgr	'A	Α	rgr	=capital Alpha, accent, Greek
eacgr	'e	έ	rgr	=small epsilon, accent, Greek
Eacgr	'E	Ε	rgr	=capital Epsilon, accent, Greek
eeacgr	'h	ή	rgr	=small eta, accent, Greek
EEacgr	'H	Η	rgr	=capital Eta, accent, Greek
idigr	"i	ϊ	rgr	=small iota, dieresis, Greek
Idigr	"I	Ι	rgr	=capital Iota, dieresis, Greek
iacgr	'i	ί	rgr	=small iota, accent, Greek
Iacgr	'I	Ι	rgr	=capital Iota, accent, Greek
idiagr	"'i	ΐ	rgr	=small iota, dieresis, accent, Greek
oacgr	'o	ό	rgr	=small omicron, accent, Greek
Oacgr	'O	Ο	rgr	=capital Omicron, accent, Greek
udigr	"u	ϋ	rgr	=small upsilon, dieresis, Greek
Udigr	"U	Υ	rgr	=capital Upsilon, dieresis, Greek
uacgr	'u	ύ	rgr	=small upsilon, accent, Greek
Uacgr	'U	Υ	rgr	=capital Upsilon, accent, Greek
udiagr	"'u	ΰ	rgr	=small upsilon, dieresis, accent, Greek
ohacgr	'w	ώ	rgr	=small omega, accent, Greek
OHacgr	'W	Ω	rgr	=capital Omega, accent, Greek



## 5 Notes

### 5.1 Mapping data (ent.xml)

The table is produced from a data file tagged with XML using XSL scripts. This approach makes it easy to do different queries.

The table comprises equivalent character encodings, positions, and descriptive texts.

Character sets covered so far are: L<sup>A</sup>T<sub>E</sub>X, known SGML/XML entity sets (ISO 8879, ISO 9573-13, HTML-4, MathML aliases/long names that are adapted from AMS and Mathematica from Wolfram Research), Unicode, and ISO 8859-1.

Multiply defined entities in ISO 8879: The inodot is defined both in amso and lat2. The glyphs are different!

### 5.2 The “Description” column

The *Description* column may contain three information fields, e.g:

```
/nsimeq N: not similar, equals
 1      2      3
```

The first field (a name preceded by a slash) is an identifier of a visual depiction of the character in MathSci, an expansion of mathfile, appendix D, 1/90, published by the American Mathematical Society.

The second field (a single uppercase letter, followed by a colon) indicates that the character belongs to a class whose glyphs are given special treatment in conventional mathematical typesetting. The classes are:

```
A relation (arrow)
B binary operator
C closing delimiter
L large operator
N relation (negate)
O opening delimiter
P punctuation
R relation
```

### 5.3 Required L<sup>A</sup>T<sub>E</sub>X packages and fonts

We have tried to use packages that come with “common” L<sup>A</sup>T<sub>E</sub>X distributions. Unfortunately, not all character entities defined in ISO 8879:1986 are covered by these. To produce these character entities, we have assembled some of the macros written by Taco Hoekwater, thereby avoiding long and nasty L<sup>A</sup>T<sub>E</sub>X control sequences (marked **local**).

The comma separated entries in the *L<sup>A</sup>T<sub>E</sub>X requirement* column refer to font packages that provide the character. **math** indicates that the symbol only are available in a math environment.

The following font packages are required:

Package	Version	Count	Distr. <sup>1</sup>	PS-Type1 <sup>2</sup>
amssymb	1996-11-03	189	.	.
amsfonts <sup>3</sup>	1997-09-17	99	.	.
amsbsy	1996-10-28	43	.	.
pifont <sup>4</sup>	1999-03-29	7	.	.
color <sup>5</sup>	1995-12-01	3	.	.
textcomp	1999-06-12	10	.	no
fontenc:T1	1994-06-01	19	.	no
cyracc <sup>6</sup>	1995-01-04	93	no	.
wasysym	1997-11-01	4	no	.
mathrsfs	1996-01-01	4	no	.
rgr <sup>7</sup>	1990-05-30	54	no	no
stmaryrd <sup>8</sup>	1994-03-03	.	no	.

All packages and fonts can be found and downloaded from the “Comprehensive T<sub>E</sub>X Archive Network” <http://www.ctan.org/find.html>.

Packages used for typesetting/preparing this document: **hyperref**, **graphicx**, **inputenc**, **longtable**, **fancyhdr**, **geometry**, and **sectsty**.

<sup>1</sup>Part of the standard L<sup>A</sup>T<sub>E</sub>X distribution? (This is MikTeX 1.20e.)

<sup>2</sup>Adobe PS-Type1 fonts available? (Needed for pretty PDF output.)

<sup>3</sup>AMSFonTS is loaded by AMS symbols.

<sup>4</sup>pifont (ZapfDingbats) is part of the PSNFSS package.

<sup>5</sup>Part of the graphics bundle

<sup>6</sup>cyracc.def is included in the AMSFonTS collection (plain T<sub>E</sub>X).

<sup>7</sup>Modern Greek fonts by Yannis Haralambous.

<sup>8</sup>St Mary’s Road symbol font is needed for ISO 9573-13 entities.

## 5.4 Characters unavailable to normal PDF output

These are produced or manipulated separately and inserted as graphics.

**Box drawings** Not displayed at all in a tabular environment.

**T1 font encoding** eth, ETH, thorn, THORN (lat1), dstrok, Dstrok, eng, ENG, aogon, Aogon, eogon, Eogon, iogon, Iogon, uogon, Uogon (lat2), ogon (dia), laquo, raquo (num).

When loaded, the T1 encoding causes dcaron, lcaron, tcaron (lat2) to be displayed with an apostrophe instead of the caron above.

Missing PS-Type1 font(s) for T1 encoding causes the document output (PDF) to use Type3 (bitmap) fonts.

**rgr package** idigr, Idigr, idiagr, udigr, Udigr, udiagr (grk2).

The diacritical marks should be above the character. Font bug?

**Lowercase script font** order (tech) is displayed in italic because we are short of a font with a small script character o.

## 5.5 Displaying Cyrillic characters

There are two possibilities for displaying Cyrillic characters:

Use the T2A font encoding part of the standard  $\LaTeX$  “Cyrillic bundle”. But there are no PS-Type1 fonts available. See `cyrguide.dvi`.

Or use the cyrillic alphabet (wncyr fonts) made available from the AMSFonts package using plain  $\TeX$ , by defining a new environment called `cyr`:

```
\input cyracc.def
\font\tencyr=wncyr10
\def\cyr{\tencyr\cyracc}
```

For details on using this cyrillic environment, see the AMSFonts User’s Guide (`amsfndoc.dvi`).

## 5.6 Latin 1 encoding

$\LaTeX 2_{\epsilon}$  has the capability of processing 8-bit characters. Put the following line in the preamble of your  $\LaTeX$  file:

```
\usepackage[latin1]{inputenc}
```

and  $\LaTeX$  will accept the complete ISO 8859-1 (Latin 1) character set. This will make typing  $\LaTeX$  much easier (i.e you don’t have to use the codes shown in section 3).

The Icelandic characters also need the T1 font encoding.

## 5.7 Numeric and publishing symbols

The “standard” way of writing fractions in  $\LaTeX$  is  $\frac{a}{b}$ . However, this does not look good in running text. We have therefore defined a “slashed” way of displaying fractions:  $\sfrac{a}{b}$  (defined in `isoent.sty`). So, instead of  $\frac{a}{b}$ , we prefer  $\sfrac{a}{b}$ . This is also consistent with the character overview in “Developing SGML DTDs” by Eve Maler.

$\LaTeX$  formats the `ff`, `fi`, `fj`, `fl`, `ffi` and `ffl` as ligature characters by default. `\textcompwordmark` is used to separate the characters.

Alternative glyphs for the suit symbols are:

<code>\clubsuit</code>	♣	club
<code>\diamondsuit</code>	◇	diamond
<code>\heartsuit</code>	♥	heart
<code>\spadesuit</code>	♠	spades

They all require a math environment.

## 6 References

Maler, Eve and Jeanne El Andaloussi. *Developing SGML DTDs : from text to model to markup*. Upper Saddle River, N.J.: Prentice Hall, 1996.

Lamport, Leslie. *L<sup>A</sup>T<sub>E</sub>X users guide and reference manual*. Addison-Wesley Publishing Company, 1994.

Goossens, Mittelbach and Samarin. *The L<sup>A</sup>T<sub>E</sub>X Companion*. Addison-Wesley Publishing Company, 1994.

Wilkins, David R. *Getting Started with L<sup>A</sup>T<sub>E</sub>X*. <http://www.maths.tcd.ie/~dwilkins/LaTeXprimer/>

*L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> for authors*. L<sup>A</sup>T<sub>E</sub>X 3 Project Team, 1997, pp. 20–22,27.

*AMSF<sub>onts</sub> User's Guide*  
<http://www.ams.org/tex/>

### Relevant standards and specifications

**ISO 8879:1986** (see Appendix D)  
Information processing — Text and office systems — Standard Generalized Markup Language (SGML)

**ISO/IEC 9573-13:1991**  
Information processing — SGML support facilities — Techniques for using SGML; Part 13: Public entity sets for mathematics and science

**HTML 4.01** (see Chapter 24. Character entity references in HTML 4)  
<http://www.w3.org/TR/html4/>

**MathML** (see Chapter 6. Entities, Characters and Fonts)  
<http://www.w3.org/TR/REC-MathML/>

**STIX project**  
<http://www.ams.org/STIX/>

**Unicode**  
<http://www.unicode.org/>

## Acknowledgements

We would like to thank:

Eli Toftøy-Andersen [elita@infotek.no](mailto:elita@infotek.no) and Steve Pepper [pepper@infotek.no](mailto:pepper@infotek.no) for proof-reading and suggestions.

Taco Hoekwater [taco.hoekwater@wkap.nl](mailto:taco.hoekwater@wkap.nl) for providing us with the `sgmlent` package.

Sebastian Rahtz [s.rahtz@elsevier.co.uk](mailto:s.rahtz@elsevier.co.uk) and Barbara Beeton [bnb@ams.org](mailto:bnb@ams.org) has provided us with mapping tables that we have used for proof reading.

Stefan Ulrich [ulrich@cis.uni-muenchen.de](mailto:ulrich@cis.uni-muenchen.de) for informing us about the rsfs fonts.