`teubner.sty`* An extension of the greek option of the *babel* package

Claudio Beccari

2008/02/10

1 Introduction

This package `teubner.sty` is an extension of the greek option of the *babel* package intended to typeset classical Greek with a philological approach. This version 2.1 cannot yet typeset the critical apparatus as the philologists are used to, but may be this work will continue and include also that facility.

This project is being carried on with the help of Mr. Paolo Ciacchi, who got a"master" degree at the University of Trieste, when he was writing his "master thesis" where he has to deal with ancient Greek philology.

This package is supposed to work with my CB fonts available on the Comprehensive TEX Archive Network (CTAN); one of the actions of this package consists in replacing the default "italic" Greek shape with the one called "Lispiakos" in Greece; this name derives from the high quality of the fonts used in the printers' shops in the city of Lipsia in the past 100 years or so; one of the printer shops that continues printing books for philologists (since 1849) is the B.G. Teubner Verlagsgesellschaft, that publishes the collection called "Bibliotheca Scriptorum Graecorum et Latinorum Teubneriana". The name given to this extension package is in homage to that printing company and to its high quality tradition in printing Greek texts.

The reader is supposed to know how to install a package with all its components in his/her TEX installation; here we do not spend a word on this topic, since the various implementations of the TEX systems are so idiosyncratic that the explanations good for one implementation are not good for another; the various operating systems also do not help in such matters. Since I have access only to a Windows 2k computer with MiKTeX and a Debian Linux one with the TEX-Live implementation of `teTeX`, I realize that

*This paper documents `teubner.sty` version 2.2d of 2008/02/10.

the explanations would be so different that they would result too confusing. Let's not speak of other operating systems and TEX distributions… Nevertheless do not forget to read carefully the accompanying file `teubner.txt`.

Another warning: while I am upgrading this documentation, the TEX-Live Team is studying how to reduce the amount of material to distribute on the TEX-Live disk, since the contributions are growing at such a rapid rate that it becomes urgent to chose what to include and what to leave outside, although always downloadable form the CTAN archives. One of the candidates to reduction is the collection of my CB Greek fonts; it is true that the whole collection of files, that includes the METAFONT source and driver files, the metric `tfm` files, and the PostScript `pfb` files, amounts to nearly 2000 files and the volume exceeds 100 MB. One of the proposals is to distribute a reduced zipped set called `cbsmall.zip` on the TEX-Live disk while retaining three other collections `cbtiny.zip`, `cbmedium.zip` and `cbhuge.zip` so that the user can proportionate the amount of downloaded fonts to his/her real needs. For using `teubner.sty` you need the full set of Lipsian fonts, so you should download it from CTAN should it be missing from your customized-size set of Greek fonts. Since the operation of the TEX-Live Team is under way, I cannot be more precise for the moment; should that Team decide as outlined, maybe I might modify this package so as to conform to the specific font collection of your installation.

This short documentation will start with briefly recalling some peculiarities of the CB fonts and their mapping to the Latin keyboard; afterwards it will list the new commands and their syntax.

2 The Greek CB fonts

The CB fonts come in all shapes, sizes and series as the extended European fonts that conform with the T1 encoding introduced after the Cork Conference of the TEX Users Group Society in 1991[1]. The CB fonts conform to the encoding that is still being called LGR, since up to now there is no established encoding name for the Greek alphabet among the TEX users, not yet, at least.

The regular shape has capital letters with serifs that are in the same style as the roman capital ones, while the lower case letters derive from the design by Didot and are very common in all texts. This shape comes also in boldface, together with the two corresponding oblique (or slanted) versions. The CB fonts contain also the upright and slanted, medium and boldface small caps alphabets.

The "italic" shape was designed in order to imitate the Olga font designed

[1]If the TEX-Live Team takes the decisions outlined in the previous section the smaller sets will have the fonts scaled up or down from a subset of the whole collection.

so as to have a contrasting style compared with the slanted Didot shape, in order to play the same role as the italic letters play with the latin roman alphabets. The Olga alphabets come in medium and boldface series, and in oblique and upright shapes.

The CB fonts are completed with the sans serif fonts, the monospaced typewriter fonts and the fonts for slides, besides an outline family that shows the regular shapes and series just with their contours.

The "cbleipzig" fonts imitate the beautiful shapes used in Lipsia; they come in medium bold and extra-bold series, without an upright version, and they are meant to replace the corresponding Olga shapes (the bold series is good for mixing with PostScript fonts, whose medium series is slightly blacker than the corresponding CM and EC fonts usually used with LaTeX).

3 Font installation

In order to use the Greek CB fonts and the extensions provided with this package, you need to install them. You can freely download those fonts from CTAN[2], where you can find also the collection of driver files for generating them with METAFONT, but since you are unlikely going to use all families, series, shapes and sizes, and since the cbleipzig and metrics fonts are lacking the driver files, I suggest you to follow the instructions given in the file `cbgreek.txt`; in facts the CB fonts need only an interface between METAFONT and the `.tfm` and `.pk` files, in order to overcome the METAFONT limitation that the output files must have the same name of the input file (not considering the filename extension); therefore all the driver files contain the same instruction:

```
input cbgreek;
```

and differ only in the name; the latter is made up of four letters and four digits; the four letters identify the Greek fonts by family, series and shape, while the four digits identify the size in points, multiplied by 100 and left padded with a zero in order to complete the four digits.

The cbleipzig fonts have their shape identified with the letter `l`, therefore for a 9pt font you need a file named `grml0900.mf` containing the above mentioned single instruction; for the metric symbol fonts at 12pt you need a file named `gmtr1200.mf` containing the above mentioned single instruction. So, should the cbleipzig and the metric symbol METAFONT driver files be missing, you don't have too much work to do to create them.

[2]At the time of writing the CTAN archives contain a huge compressed file that includes all the `tfm` metric files, all the METAFONT files and all the postscript `pfb` files, inclusive of the `cbgreek.map` file needed to instruct dvips and pdflatex for using the scalable postscript fonts.

α	β	γ	δ	ε	ζ	η	ϑ	ι	ϰ	λ	μ	ν	ξ	ο	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω
a	b	g	d	e	z	h	j	i	k	l	m	n	x	o	p	r	c	s	t	u	f	q	y	w

Table 1: Keyboard correspondence between Latin and Greek letters

Greek	᾿	῾	¨	´	`	῀	ͺ
Latin	>	<	"	'	`	~	|

Table 2: Correspondence between the Latin keyboard symbols and Greek diacritical marks

4 Inputting Greek text with a Latin keyboard

In order to input Greek text with a Latin keyboard some simple and mostly obvious key substitutions are performed according to the correspondence shown in table 1.

Notice that there is the possibility of inputting `c` in order to get the final sigma ς, but the CB fonts are conceived with the non-Greek typist in mind, so that it is even possible to input `s` at the end of words, because the whole software is smart enough to detect the word boundary and to use the correct shape of the letter sigma within or at the word end. This mechanism is so "sticky" that it becomes difficult to type an isolated initial or middle sigma; to this purpose the CB fonts contain an invisible character, `v`, that may be used for several purposes, one of which is to hide the word boundary after a sigma; therefore if you type `sv`, you get σ without any effort.

The invisible character `v` may be used also as a support for (apparently) isolated accents, especially when macros have to be used; if you type `\={v}` you get ¯, while if you omit the invisible `v` you get ˍ.

Accent, spirits and dieresis are input *before* each letter (prefix notation) without using any particular control sequence; the correspondence between the Latin symbols and the Greek diacritical marks is shown in table 2; all "upper" diacritical marks must be prefixed (in any order), while the iota subscript must be postfixed. Therefore if you input `>'a|`, you get ᾄ.

Macrons and breves are just single glyphs and do not appear in combination with any letter, due to the limitation of 256 glyphs per font; but they may be input by means of the standard LaTeX commands `\=` and `\u` respectively in order to use them as accents. Together with the macros for inserting such symbols, a complete set is available for inserting any combination of diacritical marks over or under any letter, not only vowels: see table 3. Of course the results may not be comparable with the ones one can obtain with the regular ligature mechanism; the advantage of the accent macros is twofold: (a) it is connected only to the possibility of inserting macrons and breves and/or to set the various combinations over or under any letter, even if it is a consonant; (b) for all accent vowel combinations that have a spe-

Example	Syntax	Example	Syntax
ὰ	\‘{⟨*letter*⟩}	α͜ω	\ut{⟨*letters*⟩}
ά	\’{⟨*letter*⟩}	ᾰ́	\Ab{⟨*letter*⟩}
ᾶ	\~{⟨*letter*⟩}[1]	ᾰ̀	\Gb{⟨*letter*⟩}
ϊ	\"{⟨*letter*⟩}	ᾰ̔́	\Arb{⟨*letter*⟩}
ᾰ	\u{⟨*letter*⟩}	ᾰ̔̀	\Grb{⟨*letter*⟩}
α͝ι	\U{⟨*diphthong*⟩}	ᾰ̓́	\Asb{⟨*letter*⟩}
ᾱ	\={⟨*letter*⟩}	ᾰ̓̀	\Gsb{⟨*letter*⟩}
ἁ	\r{⟨*letter*⟩}	ᾱ́	\Am{⟨*letter*⟩}
ἀ	\s{⟨*letter*⟩}	ᾱ̀	\Gm{⟨*letter*⟩}
ΐ	\Ad{⟨*letter*⟩}	ᾱ͂	\Cm{⟨*letter*⟩}
ῒ	\Gd{⟨*letter*⟩}	ᾱ̔́	\Arm{⟨*letter*⟩}
ῗ	\Cd{⟨*letter*⟩}	ᾱ̔̀	\Grm{⟨*letter*⟩}
ἅ	\Ar{⟨*letter*⟩}	ᾱ̔͂	\Crm{⟨*letter*⟩}
ἃ	\Gr{⟨*letter*⟩}	ᾱ̓́	\Asm{⟨*letter*⟩}
ἇ	\Cr{⟨*letter*⟩}	ᾱ̓̀	\Gsm{⟨*letter*⟩}
ἄ	\As{⟨*letter*⟩}	ᾱ̓͂	\Csm{⟨*letter*⟩}
ἂ	\Gs{⟨*letter*⟩}	ᾱ̓	\Sm{⟨*letter*⟩}
ἆ	\Cs{⟨*letter*⟩}	ᾱ̔	\Rm{⟨*letter*⟩}
ι̯	\c{⟨*letter*⟩}	ᾳ	\iS{⟨*letter*⟩}
u̯	\semiv{⟨*letter*⟩}[2]	π̣	\d{⟨*letter*⟩}
ḁ	\ring{⟨*letter*⟩}[2]	ῐ̈	\bd{⟨*letter*⟩}

Table 3: Accent macros

REMARKS

[1] The circumflex accent may be obtained with \~ only if attribute polutoniko was specified for the Greek language with *babel* v.3.7.

[2] Most commands may be used also with latin letters.

cific glyph in the font, the actual accented symbol is used so that kernings and ligatures are maintained; as shown elsewhere there is a noticeable difference between *αὐτός* and *αὐτός*. In this example the first word is typed in as `a>ut'os`, while the second may be typed as `a\s{u}t\'os`, or `a\us␣t\oa␣s`, or even in mixed form `a\us␣t'os` thanks to the fact that there is no kerning between ‘tau’ and ‘omicron with or without acute’

5 Ligatures

It should be clear from the previous section that the ligature mechanism is the one that offers the best results with most accented vowels; in any case it speeds up the keying in of the text to be typeset; nevertheless there are situations where you might be unsatisfied.

Fore example compare *αὐτόν* with *αὐτόν*. The small spacing difference between tau and the accented omicron is hardly noticeable, but the spacing difference between alpha and the marked upsilon is remarkable. Where does that difference come from? It comes from the fact that the smooth spirit marker inhibits kerning between the previous alpha and the resulting ligature from the spirit marker and the upsilon. In other words, by inputting `a>ut'on`, as it is suggested in the previous section, the spirit marker and the acute accent inhibit the kerning mechanism with the previous letter. In most instances the lack of such kerning is hardly noticeable, but in others it strikes your attention.

For this reason a set of macros has been defined such that it is possible to input the accented characters directly, without resorting to the ligature mechanism. Such macros have a common structure; they are formed with the letters that make up the complex glyph in a certain order, precisely every macro is made up as such:

> the first character, obviously, is the backslash character `\`;
>
> the next character is the name of the vowel, one of a, e, h, i, o, u, w;
>
> the next optional character is the code for dieresis, smooth or rough spirit, with one of the letters d, s, r;
>
> the next character is the code for the circumflex, acute, or grave accent with one of the letters c, a, or g;
>
> the last optional character indicates iota subscript with the presence of an i.

That means that, for instance, `\asai` stands for ᾄ. For your convenience such macros are collected in table 4. Such set may introduce incompatibilities with other packages or even with the primitive TEX commands. Of course one can always resort to the accent–vowel combination as exemplified at the end of the previous section; the above example ᾄ may be obtained also with `\As{a}|`.[3]

What I suggest is to typeset your paper with the regular accent vowel ligatures and to substitute them in the final revision with the accented vowel macros only in those instances where the lack of kerning is disturbing.

6 Other Greek symbols

Other Greek symbols may be obtained with ligatures or explicit commands; table 5 contains such ligatures and symbols; notice that some of these are

[3]Postfixed markings do not pose any problem with kernings and ligatures; this is why the postfixed ligature for the iota subscript may still be used also when the accent–vowel combinations are used.

`\aa`	ά	`\ag`	ὰ	`\ac`	ᾶ	`\ai`	ᾳ	`\ar`	ἁ	`\as`	ἀ
`\asa`	ἄ	`\asg`	ἂ	`\asc`	ἆ	`\asi`	ᾀ	`\aai`	ᾴ		
`\ara`	ἅ	`\arg`	ἃ	`\arc`	ἇ	`\ari`	ᾁ	`\agi`	ᾲ	`\aci`	ᾷ
`\arai`	ᾅ	`\argi`	ᾃ	`\arci`	ᾇ	`\asai`	ᾄ	`\asgi`	ᾂ	`\asci`	ᾆ
`\ha`	ή	`\hg`	ὴ	`\hc`	ῆ	`\hi`	ῃ	`\hr`	ἡ	`\hs`	ἠ
`\hsa`	ἤ	`\hsg`	ἢ	`\hsc`	ἦ	`\hsi`	ᾐ	`\hai`	ῄ		
`\hra`	ἥ	`\hrg`	ἣ	`\hrc`	ἧ	`\hri`	ᾑ	`\hgi`	ῂ	`\hci`	ῇ
`\hrai`	ᾕ	`\hrgi`	ᾓ	`\hrci`	ᾗ	`\hsai`	ᾔ	`\hsgi`	ᾒ	`\hsci`	ᾖ
`\wa`	ώ	`\wg`	ὼ	`\wc`	ῶ	`\wi`	ῳ	`\wr`	ὡ	`\ws`	ὠ
`\wsa`	ὤ	`\wsg`	ὢ	`\wsc`	ὦ	`\wsi`	ᾠ	`\wai`	ῴ		
`\wra`	ὥ	`\wrg`	ὣ	`\wrc`	ὧ	`\wri`	ᾡ	`\wgi`	ῲ	`\wci`	ῷ
`\wrai`	ᾥ	`\wrgi`	ᾣ	`\wrci`	ᾧ	`\wsai`	ᾤ	`\wsgi`	ᾢ	`\wsci`	ᾦ
`\ia`	ί	`\ig`	ὶ	`\ic`	ῖ	`\ir`	ἱ	`\is`	ἰ		
`\isa`	ἴ	`\isg`	ἲ	`\isc`	ἶ	`\ida`	ΐ	`\idg`	ῒ		
`\ira`	ἵ	`\irg`	ἳ	`\irc`	ἷ	`\idc`	ῗ	`\id`	ϊ		
`\ua`	ύ	`\ug`	ὺ	`\uc`	ῦ	`\ur`	ὑ	`\us`	ὐ		
`\usa`	ὔ	`\usg`	ὒ	`\usc`	ὖ	`\uda`	ΰ	`\udg`	ῢ		
`\ura`	ὕ	`\urg`	ὓ	`\urc`	ὗ	`\udc`	ῧ	`\ud`	ϋ		
`\ea`	έ	`\eg`	ὲ	`\er`	ἑ	`\es`	ἐ				
`\esa`	ἔ	`\esg`	ἒ	`\era`	ἕ	`\erg`	ἓ				
`\oa`	ό	`\og`	ὸ	`\oR`[1]	ὁ	`\os`	ὀ				
`\osa`	ὄ	`\osg`	ὂ	`\ora`	ὅ	`\org`	ὃ				

Table 4: Accented vowel macros

REMARKS

[1] As mentioned in the body of this text the command `\or` may produce incompatibilities with the primitive command with the same name.

`''`	’	`((`	«	`))`	»
`\GEodq`	„	`\GEcdq`	“	`:`	:
`\GEoq`	‚	`\GEcq`	‘	`?`	;
`\ENodq`	“	`\ENcdq`	”	`;`	·
`\stigma`	ϛ	`\varstigma`	ϛ	`\Stigma`	Ϛ
`\coppa`	ϙ	`\koppa`	ϟ	`\Coppa`	Ϙ
`\sampi`	ϡ	`\Sampi`	Ϡ	`\permill`	‰
`\digamma`	ϝ	`\Digamma`	Ϝ	`\euro`	€
`\f`	ϝ	`\F`	Ϝ	`\shwa`	ə

Table 5: Greek symbols

specific additions introduced with this extension package.

I draw your attention on the necessity of using the ligature `''` for producing the simple apostrophe, which, by the way, in Greek typography must always be followed by a space. The single tick mark **'** produces an acute accent, not an apostrophe, this is why it is necessary to use the double tick mark ligature.

The Milesian numerals should not worry anybody, because they are seldom used as isolated symbols; the greek or polutonikogreek option or the polutoniko attribute of the Greek language with the *babel* package offer the commands `\greeknumeral` and `\Greeknumeral`, that convert common arabic positive numbers in the Milesian counterparts within a Greek section of your document; the corresponding commands followed by an asterisk change the digamma glyph with the stigma one[4]:

> if you type `\greeknumeral{1996}` you get *͵αͳϙϝʹ*
>
> if you type `\Greeknumeral{1996}` you get *͵ΑͲϘϜʹ*
>
> if you type `\greeknumeral*{1996}` you get *͵αͳϙϛʹ*
>
> if you type `\Greeknumeral*{1996}` you get *͵ΑͲϘϚʹ*

7 New commands

This extension package introduces many new commands for typesetting Greek in a philological way. Most of such commands are collected in table 6.

A short remark on the command `\ap`: this useful command inserts *anything* as a superscript of anything else; it works both in text mode and in math mode[5]. In particular while typesetting a philological text in different languages and with different alphabets, `\ap` typesets the superscript with the current language and alphabet; if any change is required the `\ap`'s argument can contain any language or alphabet specific declaration. You can typeset things such as *Βαχύλιδες*$^{\mathrm{a}}$ by switching language and alphabet as required; the specific declarations and the commands contained in table 6 come handy also in these cases.

8 Metrics

Philological writings often require the description of metrics; for this purpose a new font has been developed that contains most of the frequent metric signs; the corresponding macros have been defined so as to set the metric glyphs as

[4]The stigma version is the standard one with the *babel* language support for Greek; with this package we adopted the digamma as the "regular" sign with the value of 6, and attributed stigma to the "variant" representation of Milesian numbers.

[5]Numerical superscripts or apices do not require math mode; numerical footnote labels are automatically inserted by LaTeX's `\footnote` command; non numerical footnote labels are easily inserted with LaTeX's `\footnotemark` and `\footnotetext` commands with their optional arguments.

Example	Syntax	Example	Syntax
Βαχύλιδες	(declaration)	abcde	(declaration)
Βαχύλιδες	\textLipsias{⟨*text*⟩}	{*αβγ*}	\lesp{⟨*text*⟩}
Βαχύλιδες	\textDidot{⟨*text*⟩}	•	\LitNil
text	\textlatin{⟨*text*⟩}	g̑	\cap{⟨*letter*⟩}
(*Βαχύλιδες*)	\frapar{⟨*text*⟩}	)——	\Coronis
(	\lpar	⌞	\lmqi
)	\rpar	⌟	\rmqi
(?)	\qmark	⌞*αβγ*⌟	\mqi{⟨*text*⟩}
…	\Dots[⟨*number*⟩]	⌜	\lmqs
. . .	\DOTS[⟨*number*⟩]	⌝	\rmqs
– – –	\Dashes[⟨*number*⟩]	⌜*αβγ*⌝	\mqs{⟨*text*⟩}
– – –	\DASHES[⟨*number*⟩]	$\overparen{\alpha\beta\gamma}$	\zeugma{⟨*text*⟩}
$^{\text{foo}}$	\ap{⟨*text*⟩}	$\underbrace{\alpha\beta\gamma}$	\siniz{⟨*text*⟩}
⫯	\sinafia	——	\paragr
⁚	\:	═	\dparagr
⁝	\;	⊗	\FinisCarmen
⁞	\?	†	\crux
::	\antilabe	ˋ*αβγ*′	\apici{⟨*text*⟩}
|	\\\|	ˈ	\apex
||	\dBar	∼	\responsio
|||	\tBar	∫	\Int
[	\lbrk	**a*	\star
]	\rbrk	***a*	\dstar
[*αβγ*]	\ladd{⟨*text*⟩}	****a*	\tstar
⟦*αβγ*⟧	\lladd{⟨*text*⟩}	|| ||	\,
⟨*αβγ*⟩	\Ladd{⟨*text*⟩}	|| ||	\!
⟨⟨*αβγ*⟩⟩	\LLadd{⟨*text*⟩}	0123456789	\OSN{⟨*digits*⟩}
$\overparen{\alpha\beta\gamma}$	\nexus{⟨*text*⟩}	*α̂β̂γ̂*	\nesso{⟨*text*⟩}
A‿B	\Utie{⟨*2 letters*⟩}	*ahβ*	\h
ajβ	\yod	*aəβ*	\shwa
aqβ	\q	*AFB*	\F
aϝβ	\f	i̯	\semiv{⟨*letter*⟩}
h^{v}	\skewstack{⟨*base*⟩}{⟨*apex*⟩}	ē̩	\md{⟨*letter*⟩}
ẹ̆	\Ud{⟨*letter*⟩}	ē̜	\mO{⟨*letter*⟩}
ĕ̜	\UO{⟨*letter*⟩}	ẹ̜	\Open{⟨*letter*⟩}
ę	\nasal{⟨*letter*⟩}	đ	\cut{⟨*b*|*d*|*g*⟩}
⊢	\dracma	⋇	\denarius
ϟ	\stater	∟	\etos
c	\hemiobelion	ɔ	\tetartemorion
s̈	\splus	š	\stimes
k̓	\kclick		

Table 6: Extended commands

if they were text; but, most important, a new definition command has been introduced so as to enable to declare new control sequences to represent complete metric feet or even complete verse metrics.

The metric glyph names are collected in table 7, while the declaration command is described hereafter.

The syntax for that definition command is similar to that of `\newcommand`;

`\newmetrics{`⟨*name*⟩`}{`⟨*definition*⟩`}`

where ⟨*name*⟩ is a control sequence name made up of letters (as usual with LaTeX) with the exception that it may start with one of the digits 2, or 3, or 4. Of course the ⟨*definition*⟩ must reflect the replication by 2, or 3, or 4; moreover if the ⟨*name*⟩ starts with a digit, when it is used by the typesetter, *it must be followed by a space.* Some examples follow:

```
\newmetrics{\iam}{\barbrevis\longa\brevis\longa}
\newmetrics{\2iam}{\iam\iam}
\newmetrics{\4MACRO}{\longa\longa\longa\longa}
```

The above definitions produce the following results (notice the space before the colon):

`\iam`: ⏑̄–⏑–

`\2iam` : ⏑̄–⏑–⏑̄–⏑–

`\4MACRO` : ––––

The definitions may contain also some symbols collected in table 6, such as ||, for example, and other symbols from the other tables.

Another important metric command is the following:

`\metricstack{`⟨*base*⟩`}{`⟨*superscript*⟩`}`

which is meant for superimposing some superscript (generally a number) over some metric symbol, which may be a single symbol or a metric foot; since the superscript gets printed in math mode, the superscript hiatus $^{\mathrm{H}}$ may be obtained with `\Hiatus` when it falls between two metric symbols, but must be well described as a math roman element when it is superscripted over something else; similarly any other superscript which is not a math symbol must be suitably set as a math roman object. In any case this command makes it easy to get something such as –$\overset{48}{\smile\smile}$.

The environment for setting metric sequences grouped with braces is described in the next section, since it is generally used within the composition of verses.

Command	Metric symbol	Command	Metric symbol
`\longa`	–	`\brevis`	⏑
`\bbrevis`	⏑⏑	`\barbrevis`	⏒
`\ubarbrevis`	⏓	`\ubarbbrevis`	⏖
`\ubarsbrevis`	⏑̲|⏑̲	`\coronainv`	⏑̇
`\corona`	𝄐	`\ElemInd`	𝄐
`\catal`	^	`\ipercatal`	+
`\anceps`	×	`\banceps`	×̄
`\ancepsdbrevis`	×̆̆	`\hiatus`[1]	H
`\iam`[2]	⏒–⏑–	`\chor`	–⏑⏑–
`\enopl`	⏑–⏑⏑–⏑⏑–	`\4MACRO`	––––
`\aeolchorsor`	–⏑͡⏑⏑⏑⏑͡⏑	`\hexam`	–⏑⏑–⏑⏑–⏑⏑–⏑⏑–⏑⏑––
`\2tr`	–⏑–× –⏑–×	`\pentam`	–⏕–⏕–‖–⏑⏑–⏑⏑–
`\ubrevislonga`	⏒	`\aeolicbii`	oo
`\aeolicbiii`	ooo	`\aeolicbiv`	oooo[3]

Table 7: Metric symbols

REMARKS

[1] A similar command `\Hiatus` produces the same visible result as `\hiatus`, except for the fact that it does not occupy horizontal space; it is useful in the definitions of full verse metrics where a hiatus needs to be inserted between two consecutive metric symbols; for example: $\overset{\text{H}}{-\,-}$.

[2] This extension package predefines some examples of metric feet and complete verses.

[3] Sometimes it might be convenient to use a shortcut for inserting the Aeolic bases by inputting `{\metricsfont I}` or `{\metricsfont II}` or `{\metricsfont III}` in order to get oo or ooo or oooo.

9 Poetry environments

In order to set poetry it is always possible to use the standard LaTeX `verse` environment; nevertheless such simple environment is not suited for philological purposes, except perhaps for very short citations. This extension package contains three new environments with various levels of complexity. Due to their relative complexity an example will be given for each one with both the input code and the corresponding result. All three environments require that any language change be declared before their opening statement, otherwise the language change lasts only to the end of the verse.

`versi` This environment does not actually set each verse on a separate line; it rather resembles an in-line list; it resorts to a command `\verso` that inserts a small vertical separator with a progressive number over it. Both the environment opening and the command `\verso` accept arguments according to the following syntax:

```
\begin{versi}{⟨label⟩}
```

```
⟨verses⟩
\end{versi}

\verso[⟨number⟩]
```

where ⟨*label*⟩ is a short text (let's say not more than 15 characters) indicating for example the poem title and the stanza number; the whole set of verses will be typeset with a left margin wide enough to contain ⟨*label*⟩; the optional argument ⟨*number*⟩ indicates the starting value for the verse enumeration; the default value is 1, but if it is specified, it is required only with the first occurrence of `\verso` or when the enumeration is restarted. In this environment the standard LaTeX command `\\` behaves normally as in regular text.

```
\begin{versi}{Meropis fr. 3}
\item[\textlatin{Meropis fr. 4}]
>'enj'' <o m`en e\ladd{>isplh} \verso[68] j`un
Mer'opwn k'ien. <h \ladd{d`e dia} \verso pr`o\\
a>iqem\hc i sj~htos \ladd{>'elassen.}
\verso <`o d'' >ex'equt''; o>u
g`ar \ladd{<omo~iai}\\
\ladd{>a} \verso j'anatai jnhta~isi bol\ladd{a`i kat`a}
\verso ga~ian >'asin.\\
prh\lladd{m}n\ladd{~hs d\dots} \verso thse. m'elas d`e
perie.\ladd{\dots}\verso
rw
\end{versi}
```

Meropis fr. 4 *ἔνθ' ὁ μὲν ε[ἰσπλη]* 68 | *θὺν Μερόπων κίεν. ἡ [δὲ δια]* 69 | *πρὸ*
αἰχεμῆι σθῆτος [ἔλασσεν.] 70 | *ὃ δ' ἐξέχυτ'· οὐ γὰρ [ὁμοῖαι]*
[ἀ] 71 | *θάναται θνηταῖσι βολ[αὶ κατὰ]* 72 | *γαῖαν ἄσιν.*
πρη⟦μ⟧ν[ῆς δ. . .] 73 | *τησε. μέλας δὲ περιε.[. . .]* 74 | *ρω*

`Versi` This environment is very similar to the standard LaTeX environment `verse`; the difference is that `Versi` automatically enumerates the verses (displaying only verse numbers that are multiples of 5) with a number in the left margin. The syntax is as follows:

```
\begin{Versi}[⟨number⟩]
⟨verses⟩
\end{Versi}
```

where ⟨*number*⟩ is the starting value of the verse enumeration; of course each verse is separated from the next one with the usual command `\\`, which has been redefined so that it just divides the verses and provides to the possible display of the verse number; it accepts the optional information that the standard LaTeX command usually accepts, both the asterisk and the vertical space amount.

```
\begin{Versi}[45]
ta; pr'osje qeir~wn b'ian\\
\paragr de\ladd{'i}xomen;
t`a d" >epi'onta da\ladd{'imo}n srine~i.---\\
t'os" e>'ipen >ar'etaikmos <'hrws;\\
t{\rbrk}'afon d`e naub'atai\\
f{\rbrk}wt`os <uper'afanon\\[1ex]
j{\rbrk}'arsos; <Al'iou te gambr~wi q'olwsen >~htor
\end{Versi}
```

45 *τα· πρόσθε χειρῶν βίαν*
δε[ί]ξομεν· τὰ δ’ ἐπιόντα δα[ίμο]ν σρινεῖ.—
τόσ’ εἶπεν ἀρέταικμος ἥρως·
τ]άφον δὲ ναυβάται
φ]ωτὸς ὑπεράφανον

50 *ϑ]άρσος· Ἁλίου τε γαμβρῶι χόλωσεν ἦτορ*

`VERSI` This third poetry environment behaves similarly to `Versi` but it displays a double verse enumeration in the left margin. The principal verse enumeration is displayed when the value is a multiple of 5; the second enumeration, just to the left of the verses, may be turned on and off; when the secondary enumeration is on, the verses are flush left, while when it is off the verses are suitably indented. The turning on and off of the secondary enumeration is achieved by means of the commands `\SubVerso` and `\NoSubVerso`; the syntax is as follows:

> `\begin{VERSI}[`⟨*outer number*⟩`]`
> ⟨*verses*⟩
> `\end{VERSI}`
>
> `\SubVerso[`⟨*inner number*⟩`]`
> `\NoSubVerso`

where ⟨*outer number*⟩ is the starting value of the primary verse enumeration, while ⟨*inner number*⟩ is the starting value of the secondary enumeration. The commands `\SubVerso` and `\NoSubVerso` must be input at the very beginning of the verse they should be applicable to. The command \\ behaves as in LaTeX, and accepts the usual optional arguments.

With the environments `Versi` and `VERSI` when typesetting in two column format, you have the possibility of specifying `\BreakVersitrue` (and of course `\BreakVersifalse`) for allowing (or disallowing) line breaks of verses; broken verses are continued in the next line with a generous indentation so as to recognize them as belonging to the same verse; the verse counter is not incremented when breaking verses across lines.

```
\begin{VERSI}[40]
k'elomai pol\ua stonon\\
\SubVerso[18]
>er'uken <'ub\apex rin; o>u g`ar >`an j'eloi-\\
\NoSubVerso
m'' >'ambroton >erann`on >Ao\lbrk ~us\\
\SubVerso
>ide~in f'aos, >epe'i tin'' >h"ij\siniz{'e\lbrk w}n\\
s`u dam'aseias >a'ekon-\\
\NoSubVerso
ta; pr'osje qeir~wn b'ian\\
\SubVerso
\paragr de\ladd{'i}xomen;
t`a d'' >epi'onta da\ladd{'imw}n krine~i.\GEcdq\\
\SubVerso[1]
t'os'' e>~ipen >ar'etaiqmos <'hrws;\\
t\rbrk 'afon d`e na\ua batai\\
f\rbrk wt`os <uper'afanon\\
j\rbrk 'arsos;\Dots[4]
\end{VERSI}
```

40 κέλομαι πολύστονον
18 ἐρύκεν ὕβˈριν· οὐ γὰρ ἂν θέλοι-
μ’ ἄμβροτον ἐραννὸν Ἀο[ῦς
20 ἰδεῖν φάος, ἐπεί τιν’ $\underbrace{\text{ἠϊθέ[ω}}$ν
21 σὺ δαμάσειας ἀέκον-
45 τα· πρόσθε χειρῶν βίαν
23 $\underline{\text{δε}}$[ί]ξομεν· τὰ δ’ ἐπιόντα δα[ίμω]ν κρινεῖ.“
1 τόσ’ εἶπεν ἀρέταιχμος ἥρως·
2 τ]άφον δὲ ναύβαται
3 φ]ωτὸς ὑπεράφανον
50 4 θ]άρσος·....

`bracedmetrics` This is an environment different from the preceding ones, although it always deals with verses. Its purpose is to set the verse metric lines grouped with a right brace, so as to show the variants of a certain metric scheme.

In order to align the metric variants and in order to place the right brace in the proper place it is necessary to fix specific lengths in terms of a unit that is compatible with the metric symbols; therefore the syntax of such spacing command and of the environment itself is the following

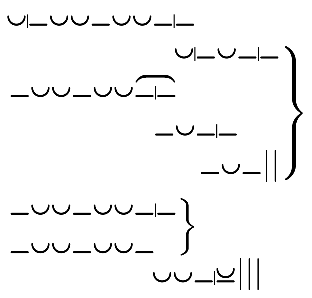
```
\begin{bracedmetrics}{⟨length⟩}
⟨metric lines⟩
\end{bracedmetrics}

\verseskip{⟨number⟩}
```

where ⟨*number*⟩ specifies the number of metric symbols the `\verseskip` should be equivalent to. Approximately the `\verseskip` will be as long as a sequence of ⟨*number*⟩ long syllables; the ⟨*length*⟩ specified as the width of the environment should equal the longest metric line contained in the block, and should be specified by means of the `\verseskip` command with its argument; but since the metric symbols are not all of the same length, it is wise to count the symbols of the longest metric line and to add a couple of units; after producing the first draft it is possible to review the number specified as the argument of `\verseskip`. Of course the same `\verseskip` command may be used to align the various fragments of metric lines within the environment. Examine the following example of input code:

```
\begin{verse}
    \brevis\svert\longa\brevis\brevis\longa
        \brevis\brevis\longa\svert\longa\\
    \begin{bracedmetrics}{\verseskip{13}}
        \Hfill \brevis\svert\longa\brevis\longa
            \svert\longa\\
        \longa\brevis\brevis\longa\brevis\brevis
            \zeugma{\longa\svert\longa}\\
        \Hfill\longa\brevis\longa\svert\longa
            \verseskip{2}\\
        \Hfill\longa\brevis\longa\dBar
    \end{bracedmetrics}\\
    \begin{bracedmetrics}{\longa\brevis\brevis\longa
        \brevis\brevis\longa\svert\longa}
```

```
       \longa\brevis\brevis\longa\brevis\brevis
          \longa\svert\longa\\
       \longa\brevis\brevis\longa\brevis\brevis\longa
    \end{bracedmetrics}\\
\verseskip{7}\brevis\brevis\longa\svert\ubarbrevis\tBar
\end{verse}%
```

which produces:



Acknowledgements

I warmly thank Paolo Ciacchi for his patience in driving me along the path of philological typography, which was totally unknown to me. He patiently chose examples from the best printed books, scanned them and sent them to me by e-mail; he patiently pointed out the flaws of my programming and suggested ameliorations. I am sure that his deep love for ancient classics will drive him to a position where he will have all the satisfactions he deserves.

Turin, 2008/02/10