Extensions to abc

The abc language was originally invented as a simple way to transcribe folk music, and has proved very effective for this purpose. There is continual pressure, however, from users who wish to use it to represent other kinds of music, and the program developers have responded to this by adding additional features to the language in order to satisfy the demand. In some cases, these extensions have proved popular enough to be widely adopted by other programs. In others, the extension has remained program-specific, making the transfer of tunes between users who use different programs difficult. The extensions described in this file are not part of the abc standard, and if you use them for music to be published on a web site or mailing list you may find that you get complaints from users of other programs. As I describe each extension I will try to give you some idea of how likely the users of other programs are to be able to deal with it.

INLINE FIELDS

This is an alternative method of inserting fields into the tune proper, without breaking the line of text, or the line of music. Any field which can legally be used in the tune proper can be placed within square brackets within the line of text. e.g. [M:6/8] [L:1/8] [K:G Dor]

One limitation (in BarFly) is that V: fields, whether in square brackets or not, should only be used at the start of a line.

Inline fields are widely supported among modern abc programs.

NON-METRIC MUSIC

If you wish to transcribe early music without barlines, or music which changes metre so frequently that it's not worth notating, you can use M:none as the metre field. In this case you can put the bar lines anywhere you want, or even use none at all without the program reporting errors. Note, however, that to display the music BarFly always needs a barline at the end of each line, so you must at least put that in. If you use M:none without an L: field the default note length will be 1/8. This extension is widely supported among modern abc programs.

TEXT ANNOTATIONS and GUITAR CHORDS

Guitar chords in abc are written between double quotes e.g. "Am", and BarFly can draw them either above or below the staff (it's an option in the Viewer Preferences dialog). This has often been used to place text annotations in the music. However, this causes problems for programs which play or transpose the guitar chords. In order to mark such text annotations out as different from guitar chords you can precede them with one of the four characters ^ _ < or >. These will not be drawn in the music, but determine where the annotation is to be drawn, above, below, to the left or right of the symbol which follows respectively. The < and > symbols place the annotation level with the head of the note, and are intended for use when placing fingering markings on chords. You may have to use the invisible rest character y here (see below) to make space for the numbers. One special word is 'segno', which when it is used in an above-staff annotation is replaced by the symbol to which it refers, so for example you can write "^D.C. al segno" at the end of a piece, to signify that it should be repeated from the beginning up to the position of the segno (which is marked with a W). BarFly's player won't recognise this instruction - it's for human readers only. You can use text annotations within chords (e.g. F#diminished on guitar is ["<1"^D"<2"A"<1"C"<3"^f]), on bar-lines and on visible or invisible rests. Several other programs support this extension (at least the ^ and _ symbols), but it's not yet widely available.

INVISIBLE RESTS

The letters x and y can be used as 'invisible rests' i.e. they can be placed in the music anywhere that a rest can, and take up the same amount of horizontal space, but no symbol is drawn. You can use them to adjust the spacing of the notes, for example to help align notes with words or to make space for a complicated sequence of guitar chords. x and y behave like rests when included in beams - they will break the beam if they have a time value of a quarter note or greater. The difference between them is that x is actually played as a rest (and its time value counts when error checking), while y is completely ignored by the player. The use of x as an invisible rest is supported in many other programs, while the use of y is (I think) unique to BarFly.

CLEFS

Clefs are not mentioned at all in the abc standard, the assumption being that everything in that document refers to music written on a staff with a treble clef. Many programs now permit the use of clefs other than the treble, and to do this you place e.g. clef=bass in the key field following its normal contents. BarFly supports four modern clefs (treble, alto, tenor and bass) plus the eight clefs used in Gregorian chant. Since the names of the clefs are unique, most programs permit the clef= part of the directive to be omitted, leaving only the clef name. Of the programs I have looked at recently abc2ps (and its derivatives abcm2ps, abctab2ps, jaabc2ps and jcabc2ps) support the treble, alto and bass clefs, abc2nwc supports the four modern clefs, Muse and abc2win support treble and bass clefs only, while abcMus and abc2midi do not display music and are therefore unconcerned with clefs.

Here's an example written on a staff with a tenor clef. The tenor and alto clefs are both C clefs - they use the same symbol, and the centre of the symbol represents the note of C. In the tenor, this is placed on the second line down from the top of the staff, while in the alto it's on the middle line. Since I have recorded my own preferences in the %%Bfly line, this tune will display at the same page size, and spacing that I used, and if you play it it will use a flute sound regardless of your own settings.

```
X:1
T:Coventry Carol
C:Pageant of the Shearmen and Tailors 1591
M:4/4
Q:1/8=175
%%Bfly 0 350 4 3 255 255 128 0 800 750 10 0 128 74 \
 1 0 1 1 1 6 1 113 142 -332 31 24 0 412 -18235 512 5 0 0
K:GDor tenor
P:Chorus
[G4D4G,4][G2D2G,2][^F2D2D,2] |\
M:3/4
[G4=B,4G,,4][B2D2G,2] | [A3F3F,3] [AFF,] [G-C A,-][GDA,]|\
[^F6D6D,6] | [G2D2G,2] [A2F2F,2] [B2D2G,2] |\
M:2/4
L:1/8
 [c2_E2C,2] [A-DD,-][ACD,] |
W:Lully,  Lul-          la, thou litt-   le  tiny        child,\
    by by lul-          ly   lul-
M:3/4
[G4=B,4G,,4] [d2F2B,,2] | [c3F3F,3] [cFF,] [B2D2G,2] | [A6D6D,6] |\
[G2D2G,2] [^F2D2D,2] [G2B,2G,2] |\
M:2/4
L:1/8
 [c2_E2C,2] [A2D2D,2] |\
M:3/4
[=B6D6G,6] |
W:-lay,thou little  tiny      child,    By    by lul-\
          ly lul-           lay
```

```
P:Verse
M:4/4
[G4D4G,4][G2D2G,2][^F2D2D,2] |\
M:3/4
 [G4D4G,,4][B2D2G,2] | [A4F4F,4] [G-B,_E,-][GCE,]|\
[^F6D6D,6] | [G2D2G,2] [A2F2F,2] [B2D2G,2] |\
M:2/4
L:1/8
 [c2_E2C,2] [A-DD,-] [ACD,] |\
M:3/4
[G4=B,4G,,4] [d2F2B,,2] |
W:O sis-ters          too,How may we          do     For to pre-\
        serve this              day This
[c4F4F,4][B2D2G,2] | [A4D4D,4][B2D2G,2] | [A4F4F,4][G-B,_E,-][GCE,] | [F6G6D,6]|\
[G2B,2G,2] [^F2D2D,2] [G2B,2G,2] |\
M:2/4
L:1/8
 [c2_E2C,2] [A2D2D,2] |\
M:3/4
 [=B6D6G,,6] |
W:pooryoungling, forwhom we do   sing       By      by lul- \
                ly lul-                 lay.
```

You can also change clefs in the tune proper, by inserting a K: field. In this case, if you don't want to change key at the same time you can simply write e.g. K:bass.

Unfortunately, not all programs interpret clef changes in the same way. BarFly retains the same relationship between pitch and symbol regardless of which clef is in use - the symbol C always means middle C, and if a treble clef is present, is drawn on a leger line below the staff. If a bass clef is present C is drawn one leger line above the staff. The pitch remains the same but the note symbols are moved up or down the staff according to which clef is present. Muse and abc2nwc also follow this convention. The abc2ps family however adopt a different convention, where the pitch relationship changes with the clef, so the symbol C means whichever C is closest to the centre line of the staff. This has the advantage that you don't have to type lots of commas after the notes when entering music on the bass staff, but becomes very confusing when dealing with music which has lots of clef changes. In future, it will also make it impossible to deal with musical lines which cross from one staff to another, as often happens in keyboard music. Music transcribed for the bass clef using this convention will display with all the notes standing on lots of leger lines above the staff in BarFly, and will also play two octaves higher than it should.

Here's a rare example of a piece of jazz transcribed into abc by Michael Methfessel, the author of abc2ps:

```
X:2
T: Blue Boy
C: Barney Kessel
T: (Herzel's Bass Line)
M: 4/4
L: 1/4
Z:Michael Methfessel
Z:with a little editing to modernise the abc by Phil Taylor
K:C bass
"C"c2 c>c- | c4 | c2 c>c-|c4 |"F"f2 f>f-| f4 | "C"c2 c>c-| c4 |
(3"G"g/^f/g/ -g z "F"(3f/"E"e/"F"f/-|f z "F"f ^f|1  "C"cef^f|"G"gz z2:|2"C"cef^f|"G"g^gab||
"C"c'dec|"Dm"df"Eb0"_ed|"C"cdef|"Gm7"gf"C7"ec|
"F7"fgag|fg"Fm7"_af|"Em7b5"edc_b,|"A7"a,b,^ce|
```

```
"Dm7"dcb,a,|"G7"g,a,b,d|"C"c>a,-"A7"a,>_b,-| "Bb"b,>b,-"B"b,2 |
"C"c"C#"^c"C"=c^f,|"F"f,"F#"^f,"F"=f,b,|"C"c"C#"^c"C"=c"C#"^c|"C"cdeg|
"F"f"F#"^f"F"=fa|"Bb"b"Eb"_e"Ab"_a_e|"Db"_d"Gb"_g"F"f"Bb"_b|"Em7b5"e_b"A7"ae|
"Dm7"defg|"A7"aed_d|"C"c>a,-"A7"a,>d-|"Dm7"dzz2|
"C"c2 c>c- | c4 | c2 c>c-|c4 | "F"f2 f>f-| f4 | "C"c2 c>c-| c4 |
(3"G"g/^f/g/ -g z "F"(3f/"E"e/"F"f/-| f z "F"f^f| "C"cef^f|"G"gz z2 |
(3"G"g/^f/g/ -g z "F"(3f/"E"e/"F"f/-| f z "F"f^f| "C"cef^f|"G"gz z2 |
(3"G"g/^f/g/ -g z "F"(3f/"E"e/"F"f/-| f z "F"f^f| "C"c zz c|z c z  Hc|]
L:1/8
M:3/4
 (3"G"g^fg -g2 z2 |"F"(3f"E"e"F"f -f2 z2 |\
L:1/16
M:2/4
"F"f4 ^f4|\
L:1/8
M:4/4
 "C"c2e2f2^f2|"G"g2z2 z4 |
L:1/4
 (3"G"g^fg -g2 |\
M:3/4
"F"(3f"E"e"F"f -f |\
L:1/16
M:2/4
"F"f4 ^f4|\
M:4/4
L:1/4
 "C"c zz c|z c z  Hc|]
```

In order to solve this problem, BarFly (and Muse) allow you to change the relationship between symbol and pitch using some extra directives placed in the K: field. The "middle" directive allows you to state which abc note symbol corresponds to the middle line of the staff, so if you change the K: field of the tune above to read

K:C bass middle = d

the tune will display correctly. (The normal relationship in BarFly for a bass clef is middle = D,) You can abbreviate the middle directive to m= if you want, but in this case the two characters must be written without a space. The middle directive affects only the display, so the tune will still play at the wrong pitch. In order to change the pitch there is another directive "transpose". This is also placed in the K: field, can be abbreviated to t=, and changes the pitch as played by the specified number of semitones. So, to fix the above tune, you can change the K: field to read:

K:C bass middle = d transpose = -24

or

K:C bass m=d t=-24

There is also a third directive "capo" which works in exactly the same way, and affects only the playing of guitar chords. Since BarFly doesn't play guitar chords it will have no effect.

All three of these directives can be used both in the header K: field, and in K: fields in the tune proper.

MACROS

BarFly supports a system of macros which allows you to write definitions for how certain symbols are to be played, or displayed. This is described in a separate file:

file://:Macros

Macros are peculiar to BarFly, and no other programs support them, although there is a Perl script to expand them on those systems which support this.

REDEFINABLE SYMBOLS

Music has hundreds of symbols, all of which someone somewhere finds essential to the particular musical genre which interests him. abc as currently defined has to represent all of these with the letters H to Z. This version of BarFly recognises the following extra symbols:

T Trill
M Mordent
P Lower Mordent (Pralltriller)
L Emphasis mark (or vertical episema in Gregorian notation)
H Fermata
I Inverted fermata
S Short phrase mark
R Medium phrase mark
Q Long phrase mark.
K Horizontal episema (in Gregorian notation only)
W Segno (You can also include the segno in text annotations above the staff, see text annotations above)

All of these are treated as 'accents', in other words the letter must immediately precede the note to which it applies. All of them are drawn in the music, and the two fermata are also played: you can set the factor by which they lengthen or shorten the note to which they apply using the Player Preferences dialog. This is a multiplication factor and can have any value between 0.1 and 9.9 (1.0 would leave the note length unchanged). The trill and the mordents can be played by means of macros, and there is a macro file for this purpose in the Macros folder. The emphasis is currently not played, and the three phrase marks are used to mark alignments between words and music, and so have no audible effect.

You will probably not want to redefine these symbols yet, but if you wish to you can. For example, to have the letter N stand for a trill, you would write:

U: N = trill

This does not affect the original default definition; both N and T will now be recognised as a trill. To cancel the original definition you must either define T to mean something else, or write:

U: T = nil
(T will now be ignored)

The program will only accept one definition in each U: field; use multiple fields for multiple definitions.

The ability to have multiple letters stand for the same symbol can be useful, for example in the case of the trill, there are many different ways of playing it. If you need more than one version of the trill in the same tune, you can define several letters to mean 'trill', then write a separate macro for each letter. All will be shown on the music as 'tr', but each will play differently.

Symbol definitions can be written in the same places as macros; either in the tune header (apply to this tune only), in

the file header (apply to all tunes in this file) or in a global macro file (apply to all tunes). You can have symbol definitions and macros in the same file. Only the capital letters H to Z can be used in symbol definitions. The names of the symbols to which they apply must be written as a single word, although they are not case-sensitive. This is the current list of legal names:

trill
mordent
uppermordent
emphasis
fermata
invertedfermata
shortphrase
mediumphrase
longphrase
horizontalepisema
segno
nil

There is no agreement between developers as to how the U: field should be used, although the default meanings for the symbols T,M,P,L & H are now fairly standard.

MULTI-VOICE ABC

BarFly supports the use of the V: field to identify separate voices within a tune and play and display them appropriately. This is probably the single most important of all abc extensions, but unfortunately this is also where the various abc programs diverge the most. Multi-voice abc is described in this file:

file://:Multi-voice abc

ALIGNED WORDS

An alternative way of writing lyrics uses the w: (lower case) field. Here the words are aligned and centred on the notes, so for example the third word in the line is aligned with the third note. Where a word extends over several notes you can use the hyphen to separate the syllables, so "Hap-py Birth-day" would count as four separate words for the purposes of alignment, and the hyphens will be drawn. If you want a single syllable to match more than one note, use one or more asterisks after it. The asterisks will not be drawn, and the word will be aligned with the first note of the group. If you want more than one word to align with a single note, separate the words with the tilde character ~. The tildes will be replaced by spaces, but the group will be treated as a single word. Finally, you can use an underscore character in place of the asterisk; here the program will draw a line from the end of the aligned word to the last note of the group with which it aligns. It is important to make the spacing between the notes wide enough so that the words don't overwrite each other. Make the line lengths short, and if necessary you can use invisible rests to separate the notes which correspond to difficult words. Aligned words are widely supported by other abc programs.

TEXT ACCENTS AND DIACRITICAL MARKS

Because the first abc program abc2mtex made use of the unix text processor TeX, it had available a ready-made method of representing characters with special marks which are not part of the ASCII character set. Users were able to enter a two- or three- character escape sequence starting with a backslash \ to specify these characters. While this was not part of the standard as described, it was useful and remains so. The reason for this is that these characters are represented differently on different platforms, so if you transfer text from a Mac to a PC or Unix machine, the user of the other machine may see a different character from the one you intended. The TeX escape sequences are, however, standard on all platforms, and when used in the title, composer or words fields, BarFly will render them appropriately.

Here is the set which BarFly understands:

Escape sequence	character
\'a	‡
\'e	Ž
\'i	’
\'o	—
\'u	œ
\'A	ç
\'E	*f*
\'I	ê
\'O	î
\'U	ò
\`a	ˆ
\`e	
\`i	“
\`o	˜
\`u	
\`A	Ë
\`E	é
\`I	í
\`O	ñ
\`U	ô
\^a	‰
\^e	
\^i	”
\^o	™
\^u	ž
\^A	å
\^E	æ
\^I	ë
\^O	ï
\^U	ó
\"a	Š
\"e	‘
\"i	•
\"o	š
\"u	Ÿ
\"A	€
\"E	è
\"I	ì
\"O	…
\"U	†
\~a	‹
\~o	›
\~n	–
\~A	Ì
\~O	Í
\~N	„
\cc	
\cC	,
\AA	

\Aa Œ
\AE ®
\Ae ¾
\aA
\aa Œ
\aE ®
\ae ¾
\O ¯
\o ¿
\\ \

Note that if you want to include the backslash itself in one of these fields you will have to use two of them together. This extension to abc is widely supported by other programs.

GREGORIAN CHANT

BarFly can represent the traditional neumatic notation of Gregorian chant in abc. This is described in a separate file:

file://:Gregorian Chant

This extension is supported only by BarFly and Melody/Harmony assistant.

Back to the table of contents:

file://:Table of Contents