

TAB

A guide to Wayne Cripps' tablature typesetting program

Leonard Williams

December 2006

Introduction: What is tab?

Tab is a freeware program developed and maintained by Wayne Cripps for typesetting music in tablature form. It has been used for small projects like personally edited favorites as well as a large number of professionally published collections of tablature. Tab will handle tablature and staff notation; and, as an added bonus, midi files can be created for a sound check on what you've just edited. The input files are small, easily portable (electronically) plain text files. The output is a postscript file, named by default "out.ps", which can be renamed and/or converted to a PDF file if preferred; sound files are midi format, "filename.mid".

Tab is a command line program, i.e., it is run from a Terminal Window (Mac OS) or a DOS Window (MS Windows). It can also be run "remotely" by e-mailing tab input files to Wayne Cripps' tab server at Dartmouth; an output file is promptly e-mailed back. Tab does nothing to the input file, which remains intact for future use or further editing.

The current (December 2006) version of tab, from which this manual is derived, is 4.3.50.

About this manual

I started writing this for the sole purpose of helping myself to remember how to get tab to do all the neat things it does. If it works for somebody else, so much the better. As a "publication", it is intended to familiarize tablature users with the tab program; it is not the *Strunk and Whites's Elements of Style* for tablature typesetters. The goal is to reveal the "many splendors" of tab, and how to integrate them to produce anything from a single page print-out of a personal favorite to a professional quality anthology. A lot of information is from Wayne's Tab Reference (ReadMe), Wayne himself (e-mail), and Rainer aus dem Spring's guide for an earlier version of tab; a little experimentation played into it, as well. Michel Caprace contributed some useful observations and was a great volunteer copy editor.

All examples in this guide were generated with the tab program; however, the original PostScript output examples have been subjected to conversion, reduction, and assorted abuses on their way to these pages. Their quality does not adequately reflect the fine, clear, very readable output of Wayne's tab program.

In usage tab is to some degree multi-layered. There are various points in the creation of a tab input file at which refinements to the output can be made: at the command line, in a header which affects the whole document, or within the body of the document itself. I am attempting to isolate these various modes in order to simplify learning tab, though that may mean at times seeming to skip some relevant functions only to introduce them at a later point. There will of necessity be some overlap.

An important and valuable characteristic of tab is that you can experiment with it and learn some tricks to meet your needs. The worst tab will do is to give you an error message and let you get back to your editing. It will neither crash nor destroy your input file.

We'll start with files in which all (most) of the information is in the body of the document, using tab's default settings; we'll then make alterations via a file header; command line "switches" will come last. Finally we'll get into the tab and staff format and generating midi files.

Contents

Tab Basics (And More)	1
Heading	1
Rhythm Flags	2
Notes and Chords	4
Staff Elements	6
Ornaments and Graces	8
Underlines and Held Notes	10
Beyond French Tablature	13
Italian Tablature	13
Interconversion of French and Italian Tablatures	14
Guitar (Milán) Tablature	14
Baroque Tablature	15
German Tablature	15
ASCII Tablature	16
Page Formatting	17
Heading	17
Line Breaks	18
System Length	18
Page Breaks	18
Pagination	19
Vertical Spacing	19
Editorial Notations	20
Right Hand Fingerings	20
Left Hand Fingerings	20
Chord Alignment Bars	21
Counting Measures	21
Original Page Turns and Line Breaks	21
Sectional and Other Labels	22
Text Above the Staff	22
Text On the Staff	22
Text Below the Staff	22
Highlights	22
Comments Within the Tabfile	24
Troubleshooting Your Tabfile	24
Refinements to the Tablature From the Header	25
Flag and Character Styles	25
Staff Size (Number of Courses)	29
General Presentation and Layout	29
Page Orientation and Size	29
System Spacing	29
Reducing Size of Page Contents	29
Staff Line Thickness	30
Text Fonts	30
Copyright Notice	32

Tab and Staff Notation	32
Notes of Music.....	32
Clef and Key	33
Time Signature	33
Style	33
Text With Staff Notation.....	36
Two Staves of Music With Tablature	37
Converting Tablature to Staff Notation	44
Command Line Options	45
Tab Midi	46
Index	47
Appendix: Wayne's <i>ReadMe</i> for Tab	49

Tab Basics (And More)

Tab can typeset several styles of tablature: French, Italian, Baroque and modern guitar are possible, with staves of 4, 5, or 6 lines. Most of the early examples in this manual will be in French tab, which seems to have become the predominantly used format among lutenists. Tab's default settings will be explored first, but there are a number of other options for refining tab's output to be treated later.

You should create a tab file in a plain text editor, such as Mac's TextEdit or Windows' Notepad. Whatever you use, be sure all text formatting is turned off; some "text wrap" functions can also be deleterious: best to turn off wrap if possible. All characters, including spaces and tabs, are critical. Tab interprets these "invisible" characters as specific elements of the finished tablature. Tab also processes characters in the input file in their context: a single letter may be interpreted in several different ways depending upon where it is or what is next to it. For example, **t** alone makes tab angry, but followed by a numeral creates triplets.

A fixed width font such as Courier is helpful (though not necessary) in visually aligning characters as you proceed. Fixed width fonts resemble typewriter fonts: all characters, from the skinny **i** to the fat **w** have the same width.

The essential portion of the tab file is the *body*. This contains all the information tab needs to typeset a tablature using the default settings. Information at the beginning of a tab file giving tab alternate directions on how to process the following data is the *header*. We'll leave the header for later (for the most part) while we work from the defaults. The file must end with an **e** (end-of-file) and an **[ENTER]** (or **[RETURN]**). Though on some systems tab will process a file just fine without the **e**, I like to use it just the same. It can be a useful tool in troubleshooting as well. Before tab will process the file, you must "save as", using "tab" as the extension; *i.e.*, newfile.tab.

If you are new to Wayne Cripps' tab, perhaps his own description of the input file will provide an orientation for visualizing what you are doing:

"I see this as a representing tablature on its side (through a mirror)."

While tablature flows horizontally across a page, the tab input file is a long column of information with rhythm flags at the left edge and music information beside it. Turn the page 90 degrees so the tab data is along the top edge; turn the page to face a mirror; *voila!*

⇒ In the tab file samples that follow, spaces are denoted by this character `_` for clarity! None of the examples are intended to be music. Don't try too hard to play them.

I've tried to juxtapose sample tab files with their respective output in a way that will make comparing the two easier. This has generally necessitated the reduction in size of the output samples.

HEADING

The simplest element of the body is the title of the tab file. You can start your basic tab file with the title in braces, thus:

```
{Title_of_this_Tablature}
```

Tab will actually run without a title, if you don't have one. More about formatting a title and including additional information will follow.

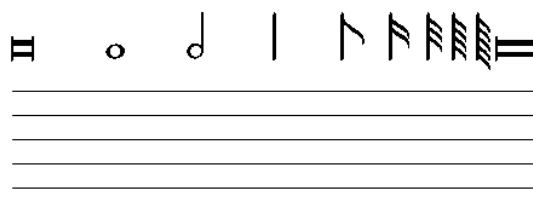
RHYTHM FLAGS

Rhythm flags are entered in the first (and sometimes second) place on each line of the tab file. Numeric values for rhythm flags correspond to the number of flags/ streamers/ leaves on the stem. Generally, the longer the note value, the greater the horizontal spacing allotted by tab to that note. Following is a list of input characters and the corresponding symbols in the tablature (the longest value, *longa*, belongs at the end of a piece—really):

[Three elements below are not part of the rhythm components (`-1_3_in` and `-`) : more about them later, pages 18; and 4 and 6, respectively.]

```
{Example_1:_Default_Tab_Rhythm_Flags}
-1_3_in"
B-
W
w
0      [zero, not the letter O]
1
2
3
4
5
L
e      [end of file]
```

Example 1: Default Tab Rhythm Flags



Note that there must be a rhythm indicator for each note or chord, even if a flag is not desired in the output. Many tablatures introduce a rhythm flag only when the value changes or a new line is begun. For rhythms that will not get a flag, use an **x**; this will apply the same durational value, hence horizontal spacing, as for the last named value, but no flag will be printed.

Rhythm flags cannot be connected with a tie, with the exception of a dot and tie combined, as described immediately below.

A dotted flag is indicated by putting a dot, “ . ”, (period) after the character. Note that a dot after a flag value is still read as part of the rhythm notation. Add to a flag a *dot and a tie* to the next note using **ƒw** rather than **ƒ.** (where **ƒ** is the flag value). For a samples of these, see Example 3, *Notes and Chords*; and Example 4, *Staff Elements*, below.

An additional print option which is used in the rhythm column is **!** (exclamation point): following a character in the *first* column, it will render that character unprinted, or “invisible”, as it were. It is very useful in creating extra spacing or manipulating bar counts. While this may be a little more advanced tab coding, I introduce it here because it comes up regularly throughout, beginning on page 6. As will be seen on page 8, **!** has a much different significance in the note section of the tabfile.

Flags can be entered individually or with *beams* connected in *grids*, a style useful in clarifying beat placement in long measures of eighth or sixteenth notes. Enter # after the first flag value in a proposed grid to initiate the grouping, using x for the remaining values in the group. The grid can be two, three, four notes long—whatever works! The grid will break when a new numerical flag value is entered. Flag values within a grid can be halved by entering | after an x, or dotted by adding * after an x; for a dot on the first note in a grid use * in place of # after the numeral. If a grid will not close properly it may need a character after it, like a bar line (an “invisible” character, described above, will work at the end of a system with an open measure). See Example 2, *Grids and Triplets*, for usage of the grid format.

Triplets in any particular flag value can be denoted with **t_n**, where **n** is the flag value. See Example 2. On page 12 of the section *Underlines* find two more ways to print triplets. In Baroque tablature (page 15) the triplets print in the style of the London Weiss MS, with the rhythm flag of the first note omitted.

{Example_2:_Grids_and_Triplets}

-l_ "5.5_in"

2# [start a grid]

x

Example 2: Grids and Triplets

x

x

1#

x | [halved value]

x |

2* [initial dot]

x |

x*

x |

3#

x

x

3#

x

x

0.

t1_a

1_b

1_d [Notes! How-to in next section]

t2_ a

2_ c

2_ a

e

Tab has several styles of rhythm flags: they will be discussed in depth later.

NOTES AND CHORDS

The second through tenth spaces to the right of the “rhythm section” are dedicated to the strings or courses. Technically a *hyphen* (-) separates rhythm from notes, but is only necessitated in a few special circumstances, which will be treated as they arise. In the examples it is omitted unless required. In our default French tablature the highest course is immediately to the right of the flags. A space is interpreted as an unplayed course: thus, two spaces followed by an **a** indicates that only the open third course is played. Spaces are not required after the right-most note entered. Letters **a** through **p** can be used, but **j** never appears in tablature. Lowercase letters are used, but a certain few uppercase characters will print in the style of the *Board MS*: **F**, **G**, **H** and **I** (see page 26).

Bourdons are indicated beginning at the seventh space after the rhythm notation with forward slashes (/) to indicate courses past the seventh. (Tab uses both slashes and numerals for basses in Baroque settings, page 15.)

Rests rarely appear in tablature, gaps among the notes often being simply a heldover note in a polyphonic melange. However, they can be indicated by several means, and are more often seen in accompaniments to songs. Many tablatures used only a flag with no note or chord below it, as in Example 1. If a glyph for a rest is desired, use **Rf** or **fR**, where **f** is the flag value to be associated with the rest. A variation on the latter method is to use **fR!**, which will print a rest with no flag, with spacing corresponding to the value of **f**. (Recall that **!** renders a character unprinted.) Rests with values **5** through **0** get a hook-style glyph like those found in Dowland’s books, c. 1600. Lower case **w** has a “hat”, and **W** a hanging block. **1R** and **R1** print a modern eighth rest which looks like a seven. *Nota bene*: **R** alone will not work.

If additional horizontal spacing is required between notes, the letters **i** or **j** can be entered in the rhythm column. **i** is slightly wider than **j**; they will not print, but may cause a hang up in tab’s midi function (page 44). The **i** is used in Example 4 on page 7.

{Example_3:_Notes_and_Chords}

1_abd

xfd_e

x_cc_a

3l

xh

xk

xl

3k

xl

x_n_o

x_m_p

1!

1_f_F_a

2_g_G_/a

x_h_H//a

wW_i_I///a

b

1

2R!

R2

R1

R0

wR

WR

e

Example 3: Notes and Chords

[invisible flag!]

[Board...

...and...

...bourdons]

[bar line]

[invisible flag with rest]

[R before

...or...

...after]

STAFF ELEMENTS

Here are included numerous elements which, along with lines and notes, make up a system of tablature: bar lines, time signatures, fermatas, and others. All of these are encoded in the far left column of the tab file; they are, in effect, part of the rhythm of the piece.

Bar lines are indicated with the letter **b**. A heavy bar is **B**. (Herein lies the reason for the *hyphen* in Example 1: a **B** with nothing after it is interpreted as a heavy bar line; the hyphen, which serves as a divider between rhythm and notes, implies that something will follow. Hence **B-** is interpreted as a brevis rather than a bar.) A tie can be placed over a bar line with **bT** or **BT**. A pair of thin barlines can be printed with **bb** (side by side) or vertically placed. Tab will not allow **B** in horizontal pairing.

Notation of *time signatures* is very flexible. **C** and **c** are common time and cut time, respectively. Also available is **Sn**, where **n** is a numeral from **2** through **6**, which will print a large numeral before the staff. Other possibilities are combinations using **Sn-m**, where **n** and **m** are both numerals, which will print a meter in fractional format; i.e., $\frac{3}{4}$ (which looks proper in tab output). **C** and **c** will print on the staff, but the various functions with **S** print before the beginning of a system (when used at the start of a piece). To print the latter style of time signature on the staff, begin with a bar line (**b**) or an **i** (indent). If desired, the large numeral style of time signature can be printed with a bar through it using **Sn |**.

Note that the time signature is for your reference: tab doesn't count beats and couldn't give a hoot if you put five beats in a six beat measure. In this way tab can help you amuse your friends and confound your enemies with things like an extra eighth note in a middle of a recerar. This characteristic of tab has a less nefarious side to it: as will be seen later, it allows one to alter spacing and positioning of other elements through the use of "invisible" notes that would otherwise stymie a program which counts beats per bar.

There are four *Spanish time signatures*, such as those used by the vihuelists. These actually indicate relative tempo rather than meter. Use **SC**, **Sc**, **SO**, and **So**: see the output in Example 4 for appearance and definition. These characters, like the other **S**'s, will print before the beginning of a staff unless preceded by a bar line or indent.

Fermatas are of two styles, indicated by **Y** or **y**. These can be combined with bar lines to produce a fermata above a bar: use **Yb** or **yb**; in this situation, **B** is not an option for a fermata over a bar. **Y.** prints the fermata above a column of dots. The fermata can also go above notes, but is best used only with stemless ones: **L**, **B**, **W**, **w**; **Y**, producing a fermata with no center dot, is better used over a stemmed rhythm flag if needed. Often the flag was simply omitted: **y** stands alone nicely. It is generally safest to follow **y / Y** with a hyphen rather than a flag character.

Repeats are formed with combinations of dots and **b**'s. Several samples are included in Example 4. As with the double barline, tab will allow horizontal combinations with **b** and dots, but not with **B (.b., .bb. and .b.b.** will work); these can also be laid out vertically. A repeat with **B** is in Example 4.

First and second endings are created with **A1** and **A2**. Place the **A1** as the last bar before first ending material, and the **A2** right after the repeat sign ending that section. These characters will print a thin bar line along with the overhead square-root thingy. Precede **A2** with a **.B** for a nice repeat sign. (See *Editorial Notations* regarding correct measure counts in pieces with multiple endings). (In older versions of tab, when either of these initiate a system, they must be preceded by a character or tab will report an error. **3!** works nicely, being a flag value that tab will take up little space; and it will not print because of the **!**; this is explained under *Editorial Notations* (page 21); see also bar 4 of the second system in Example 4).

Go back to the top with a **D** to print *D.C. (Da Capo)*. For a *dal segno* arrangement, use **d** to print the segno glyph; at the point where "dal segno" is to be printed, use **[D.S.]**. An **f** will print *fin*. These items will print above the staff.

You can finish your composition with a *flourish* using a final **q**, **Q** or **m** (Mesangeau's flourish). (But don't forget the end-of-file **e** !)

{Example_4:_Staff_Elements}

i [indent]

S3-2

3! [make a space]

w_____a

YW_cdc Example 4: Staff Elements

b

B

. S6

b

b

S3 |

d

1__d

2a

x_d

1_c

A1

1_bdc_/a

0_bdc

.

B

A2

wffdca

W_____a

[D.S.]

.bb.

C

c

yb

B

m [Mesangeau flourish]

b

Sc [despacio: slowly]

1__c

1__b

bT

2__b

2__b

1__a

D

b

B

SC [moderato]

1._c

1W_d [dot and tie]

b

2_d

1a

1.c

b

B

Q

B

b

So [despacio: slowly]

bb

SO [apriesa: quickly]

f

b

⇒ ⇒ ⇒ ↗ B
e

ORNAMENTS AND GRACES

Tab can print a variety of symbols for ornaments. Characters used in the tab file are entered to the left of the note they adorn, and most are by default also printed to the left of the note. Some characters used as ornaments are also used as modifiers to rhythm flags (*, #, - and @): when they are used with a note on the top course, use - (hyphen) as a separator between a rhythm flag and the decorated note. (The separator can actually be used thus throughout a tab file, but it is only really necessary to eliminate contextual ambiguity. If you can spell in Italian, you may find the hyphen similar to the letter “h” in usage.) The ornaments are best described through display (Example 5); tentative naming for some is from Stan Beutens’ study of English ornament signs.

The first three bars demonstrate default usage and print-out of the characters: entered left of the note, printed left of the note. Observe that the @ in the third measure needs the hyphen as separator, since it is otherwise used in editorial work discussed below (if in doubt, use a hyphen after the rhythm character!). When using the ornaments in this way, tab does not count the space taken up (in the file) as a course; notes can be closely stacked in print even though they may be separated by ornament symbols in the tabfile.

The fourth measure shows how an ornament symbol can be printed above, below or right of the note using special “operators”. First is the “infix operator” ! . Used before a character, this tells tab to print the symbol instead of a note: therefore above or below the affected note; *i.e.*, “in” the stack. Both ways are shown in the sample. Note, however, these limitations: 1) An ornament thus treated takes the place of a note, so that another note cannot be printed immediately below a bottom-decorated note, nor can it be printed immediately above a top-decorated note; 2) A top-decorated note cannot be on the top course; 3) A top-decorated note which is solitary or uppermost in a chord must be preceded by the separator -. Within a chord there may result some ambiguity, as in chord 4 of bar 5. In general, if an ornament (or other symbol) falls vertically within the chord stack, it must be counted as if it were a played course when counting spaces for placement of notes; it can restrict placement of vertically adjacent notes. This can be seen in the example. Check your output for correct vertical spacing! A further note on the ! : it can be used to print any character which follows it.

A second operator is the “postfix operator” & . Use this to print a symbol right of the note. Since this operation keeps the symbol out of the chord stack, notes can be placed vertically adjacent. This is sometimes used in conjunction with the default placement to print double ornaments: signs fore and aft of the note.

The Panmure 5 MS of Mesangeau’s works indicates direction of strumming of chords: dot below, strum down; dot above, strum up. To place a dot above a chord with a note on the top line, use a B after its rhythm flag. Note the need for a - in the first chord of bar 4.

In the second system, the first bar uses two ornaments which by default print above the note. These can also be moved around using operators. First used is the “prefix operator” “ (double quotes) which allows the symbol to be printed left of the note, taking it out of the chord stack. The next two samples of these symbols use the infix and postfix operators.

In the next measure is a symbol which by default prints a “smiley” immediately below the affected note. This is a Baroque “hammer on”, and rightfully goes beneath the note; it does not affect the chord stack, and a note can be placed immediately below. (This, too, can be moved around if desired.) The “comma” of Bar 2, printed *right* of a note, is a Baroque “pull off”, opposite of the “hammer on”. Another Baroque sign follows: a backslash between notes indicating something like a strum. (More about the Baroque format on page 15.)

Third bar of the second system has an ornament that is entered in place of rhythm values in the first space. A wavy line (arpeggio? trill?) is printed.

The antepenultimate bar displays some other unusual markings that may be used as ornaments: the mysterious ? and >, which must be seen! Usage? Baroque?

Penultimate measure has stacks of ornaments. Not all of the ornaments respond as expected with the several placement operators. Experiment! Check vertical placement of notes!

Very last is more a treatment than an ornament: o (uppercase) in the rhythm prints a squished gray chord as found in some Capirola pieces (errata, second thoughts?).

{Example_5:_Ornaments}

-l_"4.5_in"

Example 5: Ornaments

b

1+a

x_*b

[Dowland fall; Board relish, shake]

x_#c

[Dowland relish, shake]

x_..xd

[Board fall]

b

x\$a

x_<b

x_'c

[appoggiatura]

x_..`d

b

1-@b

[tremolo?]

x_^b

x_=c

[Dowland double fall]

x_..%d

bb

1-.dda

[Mesangeau Panmure 5]

1Baacc

1ffh.

bb

1_..d!^

[infix operator]

x_..!^d

[Mudarra's held note symbol]

x_..d&^c

[postfix operator]

2_b!+c

x-!+b

bb

-l_"5.5_in"

b

1;e

[can't go on top course]

x_~c

x";e

[prefix operator]

x_"~c

x_..e!;c!~

[infix]

x_..e&;c&~

[postfix]

bb

1_c

[Baroque fall]

x_ca

1_..c&`

[Baroque pull-off]

x_b_\\d

bb

2_..c

~_..b

~_..c

~_..b

bb

1_..?a

x_..?a

x_..>c

x_..">c

bb

0+a*b#cxd\$e<f [stacks!]

x

0a&%b&@c&`d&^e&=f&'

Ocdda [Capirola weirdness]

b

B

e

UNDERLINES AND HELD NOTES

Tab has several ways of printing the *underlines* in tablature used to show the duration of a held note. To print *straight lines* you can use the pairs { } and () (braces and parentheses). The opening half of each pair begins the line, the closer ends the line. The lines can connect two notes, or they can simply end. By default these two pairs begin and end *below* a note. However, by using the postfix and prefix operators (&, ") the line can be placed *between* two notes : d&{, then " }d . The) or } alone will place the end of the line above or below a note on the "course" where the character is placed. These two pairs can be used in tandem for sets of lines that overlap: be sure to alternate the opener and closer of the two types so the lines will "find" their proper homes. When using lines, note that they take the place of a note or course when occurring within a chord stack, as do the ornaments. Check your output for correct vertical placement of notes! A number of possibilities are given in Example 6.

Curved lines are possible using the pairs U , X and [] (uppercase letters and brackets). These differ from each other only in that the former prints in a space, the latter on a line. The closers of these pairs must be placed at the same space (course) as the openers: the curved lines cannot be tipped. These two sets can be used for overlapping lines in the same way as the braces and parentheses are. They cannot be used with pre- and postfix operators.]v as a closer will print a curved line concave downward.

A third style of underline is really no underline at all, but *Capirola's* notation of when to hold a note, and when to release it. Use lowercase w for the "hold" and u for the "release". These produce a tiny psi (Ψ) and omega (Ω), respectively. Place these characters as for a seventh course, last in a chord; they print below the staff.

Mudarra indicated a note to be held with a carat (^) above a note (Example 5 above).

Sometimes necessity dictates the continuance of a line into the next system: tab must be tricked to allow this, otherwise there will be lines all over! Use the straight lines: { } and () . End the line at the end of a system by anchoring it where appropriate (see bars 10 and 11 of Example 6). At the start of the next system, make an invisible starting anchor using a small flag value and ! : this prints a space of a length proportional to the durational value of the flag, but with no flag. Next end the line with the matching closer character.

```
{Example_6:_Underlines}
```

```
b
```

```
2a_{}_a{
```

```
xc
```

```
xe
```

```
xf_{}_a}
```

```
b
```

```
2a_{}_a&{
```

```
xc
```

```
xe
```

```
xf_{}_)
```

```
b
```

```
2ab(_{}_a&{
```

```
xc
```

[cross-overs]

```
xd_{}_}
```

```
xf_{}_")d
```

Example 6: Underlines

```
b
```

```
b
```

```
3#_{}_d
```

```
x_{}_a
```

```
x_{}_c
```

```
x_{}_d
```

```
3#&_{
```

```
x_{}_{}_c
```

```
x_{}_{}_e
```

```
x_{}_{}_a
```

```
b
```

continued next page

3#_}_c([or 3#_c(with...
 x_d x_" }d but horizontal spacing may suffer]

xa [overlaps]

xc
 3#d{
 x_a)
 x_c
 x_}a
 bb

b
 0_d_aU
 0_c_X
 b
 0_c_e[
 x_c]

b
 2a_a[
 xcc_U

[overlaps]

xe_c
 xf_dX_]
 b
 b

1a_c_w [Capirola hold]

1_e
 1_c_u [Capirola realease]

ya_c
 b
 B

3#_a_d{
 x_a
 x_b
 x_d
 3#_a&(
 x_c)
 x_d
 x_}_a
 b

-1_"2.5_in"

b
 3!_([invisible flag as line "home" anchor]

3#_}_c{
 x_d
 x_a
 x_c
 3#_d&({
 x_a
 x_c
 x_}_e

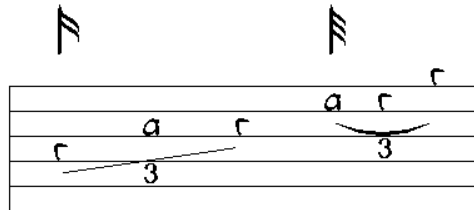
b
 B
 1_a[
 x_v_a
 b
 e

[reverse curve]

An additional use of tab's underlines is two alternative renderings of *triplets* (see also *Rhythm Flags*, page 3). These are only practical when space permits in the staff. The triplet connecting line is printed on the staff with the "3" below it, as seen in Example 6a. This will work for any other grouping (sextet, quintet) as well.

```
{Example_6a:_More_Triplets}
-l_"2.625_in"
b
2_._c{
x_._a_3
x_._c}
3_aU
x_c_3
xc_X
b
e
```

Example 6a: More Triplets



At this point you are more than prepared to lay down a goodly line of tablature. Try something simple. As you've seen in the example tabfiles, start it with a **b**, end it with a **b**, and press [ENTER] / [RETURN] twice to start a new line. Another **b** and a new line begins.

Beyond French Tablature

[Some of the following examples use vertical alignment bars described below in *Editorial Notations*.]

ITALIAN TABLATURE

This format was also used by most of the vihuelists (Milán used what now closely resembles the modern guitar tablature). At this point we need to start adding options via a header, but these will be kept simple for now. In order to provide appropriate spacing beneath the flags for courses beyond six, tab has to be told if you intend six-course or seven-course Italian tab: for seven-courses (or more) begin the tab file with `-s`. Tab will, by default, print notes on spaces. In order to print notes *on* lines, add `$line=o` (lower case letter *o*) to the header. (`$line=b` prints the notes *between* the lines, the default placement.)

Rhythm flags are entered as usual; notes are entered as numerals with bass next to the flags, treble six spaces away. In setting tablature for more than six courses, don't forget to leave a space for the bourdons!

High frets, of two digits, require a special sequence, since tab only recognizes one character per course in a chord: enter `Nxy`, where `x` and `y` are the digits of the fret. Occasionally a Roman numeral was used for "10"; enter `!x` (note the lowercase "x"). A trick is needed for Roman numerals above 10, as used by Mudarra: `!x&1` yields XI (note the lowercase "x" and the numeral "1"). There seems to be no way to print a Roman numeral 12, XII. For the best results with Roman numerals, head the file with `$numstyle=italian`. See Example 7.

```
-s
$line=o
$numstyle=italian
{Example_7:_Italian_Tablature}
-l_ "4.875_in"
b
2/2&{--_2
x_3.
x_3
x"}0
x_0_0.w
x_0
3_3
x_2.
x_0
x_2.u
b
1_0_3
x_2_5
x0_0_5
x_2_+4
b
b
29_____N10
x-!x_____N11
1N13_____N12
b
1//0_____!x
x///0_____!x&1
b
e
```

Example 7: Italian Tablature

The image shows a musical score for a six-course guitar in Italian tablature. The time signature is 2/2. The notation includes rhythmic flags, fret numbers, and Roman numerals. The tablature is written on six lines, with notes placed on lines and spaces. The bottom line (bourdon) has a fret number of 4. The notation is divided into measures by vertical bar lines.

INTERCONVERSION OF FRENCH AND ITALIAN TABLATURES

If you have a tab file in French tab and would like to see it in Italian, there is a simple procedure which obviates re-typing the whole thing. The switch **-h**, included in the header, will convert the output of your French tab file to Italian. If the tablature in question is for seven or more courses, use **-H**. Note that the switch repeated will undo itself! This will exchange letters for numerals in the output, and change the order from treble at the top to bass at the top. One additional parameter is needed for tab to move formerly French characters from spaces to lines: use **\$line=o** to put characters on lines (see immediately above, ITALIAN TABLATURE).

Tab can also, but with difficulties, go from Italian to French. The Italian tab file must be for six courses; use **-h** and **\$line=b** in the header. Peculiarities in tab's processing prevent it from properly aligning courses when attempting to convert seven course Italian tab to French tab.

It is also possible to very easily create "dual language" output with little extra typing. First, copy as much of the tab file as you will need (title, tab). Paste this below the body of the file. Be sure to delete (or disable with **%**) the original end-of-file **e** and add an **e** and **[RETURN]** or **[ENTER]** at the very end. As a header for the second half of the file, use **-h** (or **-H**) and **\$line=o / b** to translate the output. Other style changes can be added here as well (page 25).

If a page break between "language" versions is desired, see page 18.

Caveat transtabulator! If you know ahead of time that you may want to be able to convert a file, keep it simple: not all added symbols in a file get moved along with their associated characters. Also, keep in mind that French to Italian offers more versatility.

GUITAR (MILÁN) TABLATURE

This is also known as *Spanish* tab, but the fact is that only one of the Spanish vihuelists, Milán, used it. It has nevertheless survived as modern guitar tab. Everything is entered as with French tablature, but numerals are used instead of letters. The tendency in modern tab is to print a rhythm flag for every note, unlike the early style of printing a flag only with a change in rhythm value. (See pages 44 and 46 for reasons to use **\$numstyle=italian**.)

```
$line=o
{Example_8:_Guitar_Tablature}
-l_"4.875_in"
b
2#_2_4{
x_3.
2#_3
x_5}
2#_0_0
x_0
3#_3
x_2.
x_0
x_2.
b
1_3_0
15_2
15_00
1+4_2
b
b
2#N10
xN11
1N12
b
1_N10
1_N11
b
e
```

Example 8: Guitar Tablature

2	3 3	0	0 3 2 0 2	5	5 +4	10 11 12	10 11		
x_2.	x_0	x_2.	x_0	3	0 2	0 2			
4	5	0	0	0	2 0				

BAROQUE TABLATURE

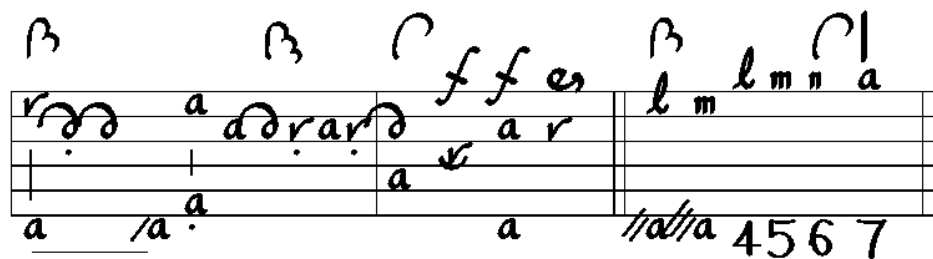
Tablature in the baroque acquired a different graphic style, not to mention the tuning differences. To “turn on” the Baroque font set, begin the file with `-b`. Data entry is as for French tablature, but there are some minor differences in output. Some of the ornaments do not print (? and >). Bases, in order from the seventh course down, are *a, /a, //a, ///a, 4, 5, 6, 7* (numerals represent the number of slashes that would be needed).

As noted on page 3, triplets print in the style of the London Weiss MS, with the rhythm flag of the first note omitted. These triplets are problematic in that the examples on which they are based are very few. The “cleanest” print is obtained by omitting the flags altogether; i.e., type in the value of the first flag after a `t` and use `x` for the following two flag values; no flag will print, but there will be an arc with a “3” above it over the three notes.

[The example is in Renaissance tuning!]

```
-b [use Baroque font]
{Example_9:_Baroque_tablature}
-l_ "5.125_in"
b
2_c_--_a{
x_d.
x_d
x_/_a}
x_a-_a.
x_a
3_d
x_c.
x_a
x_c.
b
1_d_a
xf_c [hammer on]
xf_a_a
xe'_c [pull off]
b
b
2_l_/_/_a
x_m_/_/_/_a
x_l_4
x_m_5
1_n_6
0_a_7
b
b
e
```

Example 9: Baroque Tablature



GERMAN TABLATURE

German tablature is not supported by tab.

Page Formatting

HEADING

The *heading* for the tablature can be positioned differently than indicated thus far, and other information can be added to it. The default position of text is in left alignment; however, it can be centered thus:

```
{\CL/Centered Title}
```

or right aligned, thus:

```
{/Right Aligned Text}
```

In addition, text can be split to appear at each side of the page:

```
{Left Side/Right Side}
```

Note: tab will not wrap text in a line that is too long. To break the line, press Enter and continue within a new pair of braces.

Vertical spacing between lines can be added with an empty pair of braces.

A typical heading might look like this:

```
{\CL/Title}  
{Source, _pages/Composer_(dates)}  
e
```

And here is the output:

	Title	
Source, pages		Composer (dates)

See page 30 for more “advanced” text options.

Tab can print special European language characters, but they must be “coded” with special control sequences shown in the following lists. Some punctuation also requires special coding.

Special letters:

<i>f</i>	<code>\sl</code>	<i>œ</i>	<code>\oe</code>
<i>ß</i>	<code>\ss</code>	<i>Œ</i>	<code>\OE</code>
<i>æ</i>	<code>\ae</code>	<i>ø</i>	<code>\oo</code>
<i>Æ</i>	<code>\AE</code>	<i>Ø</i>	<code>\OO</code>

Diacritical marks:

<i>é</i>	accent aigu	<code>\' + [letter]</code>	
<i>è</i>	accent grave	<code>\` + [letter]</code>	
<i>â</i>	circumflex	<code>\^ + [letter]</code>	
<i>ü</i>	umlaut	<code>\" + [letter]</code>	
<i>ē</i>	macron	<code>\= + [letter]</code>	
<i>ě</i>	breve	<code>\u + [letter]</code>	
<i>ě</i>	double acute	<code>\H + [letter]</code>	
<i>ñ</i>	tilde	<code>\~ + [letter]</code>	
<i>ç</i>	cedilla	<code>\c_ + [letter]</code>	<i>Note the space!</i>
<i>ž</i>	dot above	<code>\. + [letter]</code>	
<i>š</i>	caron	<code>\v + [letter]</code>	

Punctuation (some of these may entail the use of extra spaces to avoid crowding):

<code>ç</code>	<code>?` or ></code>	<code>[</code>	<code>\[</code>
<code>ï</code>	<code>!` or <</code>	<code>]</code>	<code>\]</code>
		<code>{</code>	<code>\{</code>
		<code>}</code>	<code>\}</code>

Here's a small example of usage:

```
{\`E_di_S~ao_Paolo_Fran\c_cois_Bi\ssler?_!`Claro!}  
e [Note the space in the cedilla sequence!]
```

yields this:

È di São Paulo François Bißler? ¡Claro!

Nota bene: Not all special characters are available in all fonts. If, for instance, an accented e doesn't print correctly, and the entry sequence is correct, try a different font (page 30).

LINE BREAKS

The default length of a system of tablature is 6.5 inches. Unless you tell it otherwise, tab will try to cram everything on one system, finally giving you a warning of “too many notes on one line, system x”. To begin a new system, simply leave a blank line; i.e., press **[ENTER]** twice: once for the break and once to begin the next entry. See this in Example 3. Tab may warn about too many notes before the system gets too crowded for clarity: that's fine, tab will complain but will still print your tablature. To respace things—add to a system or reduce its content--simply cut and paste from the end of one system to the start of another; the Mac lets me highlight and drag very nicely.

SYSTEM LENGTH

Left to its own devices, tab will neatly distribute notes across a 6.5 inch system spaced according to their durational value. However, tab will not shorten a system when it is “underfilled”, as often happens at the end of a piece. This doesn't look good. To shorten an underfilled system, at the head of that system use

`-l_“xx_uu”` *Note the spaces and double quotes! Lower case “ell”!*

where **xx** is a floating decimal (3.5, 54, 5.25, etc.) and **uu** is one of the following: **in** (inches), **mm** (millimeters) or **pt** (points, at 72 per inch). See Examples 1, 2, 3, 5 and 6 for usage, but note that the actual systems will not measure accordingly because of reduction for layout of this manual. This instruction will affect all subsequent systems until a new page is begun, or the length is reset using **-l_reset**.

Length of a line of text need not be specified, but must be manually broken, as noted above.

PAGE BREAKS

This is automatic, but to force a new page, simply type **p** (lower case) between two line breaks, thus:

```
[ line break ]  
p  
[ line break ]
```

This is useful when there are several compositions in a single file that you want separated.

PAGINATION

Automatic page numbering can be added by entering **-p** at the beginning of the file (header area). Odd page numbers will appear at upper right, even at upper left. This input is included in Example 13's tabfile, but it does not appear in the cropped output.

Page numbering can be done manually, also. This technique allows irregular page counting (for whatever reason), use of Roman numerals, or combinations of numerals and letters. Up to five characters can be used. Characters will print at the top of the page as above; letters (when used sans numerals) print to the left. At any line break, a **P** (upper case) following a **[RETURN]** or **[ENTER]** (i.e., a *line break*) will signal the input; follow with a space and the required characters. Both the line break and the space are essential. Note that the **P** sequence can be entered at any line break on a given page: it will still print at the top of that page.

```
2__a__c
b
[Return/Enter]
P_1234b          [print page # 1234b]
%system 2
b
2_____a
```

VERTICAL SPACING

Additional spacing can be provided between systems with

V_ "xx_uu" *Note the spaces and double quotes! v is upper case! No hyphen!*

Usage is the same as for **-l_ "xx_uu"** (page 18); note that a hyphen is not used with **v**. See page 29 for more related layout information.

Editorial Notations

Under this rubric I am including, perhaps arbitrarily, items which are not essential to the music but assist in performance or interpretation. Some features make tab very amenable to transcribing from primary sources and indicating editorial changes. You could argue that some of these items belong elsewhere, but please don't.

RIGHT HAND FINGERINGS

Right hand fingerings are generally indicated with dots: a single dot beneath a note indicates index finger, two dots middle, three dots ring, and a short vertical line is the thumb. Enter the appropriate character (., :, ;, |) right of the affected note to print the symbol beneath the note; to the left will print the symbol above the note. A double "pipe" (||) will print two short verticals beneath the note. Be aware that these symbols will affect note stacking; they cannot be printed to the side as can the ornaments. If printed above a note, the characters may interact with rhythm characters: be sure to use the hyphen (-) to separate them if tab argues. (Example 10 illustrates graphics, not technique.)

```
{Example_10:_Right_Hand_Fingerings}
-l_"2.0_in"
b
1a:_c|
2c:
xe.
0f;d:a|
x:ea.c||
b
e
```

Example 10: Right Hand Fingerings

LEFT HAND FINGERINGS

Left hand fingerings are indicated by numerals 1 – 4 printed to the left of the note and representing the index (1) through little (4) fingers. Enter them in the tab file before a note using `\[numeral]` as in Example 11. Because they print beside a note, the left hand fingerings do not affect position of notes in a chord stack.

```
{Example_11:_Left_Hand_Fingerings}
-l_"2.0_in"
b
0_\3e\4f\2e\1c_/a
b
0_\1b\3d\2ca
b
0\4f;\2d:a|
b
e
```

Example 11: Left Hand Fingerings

CHORD ALIGNMENT BARS

In manuscripts and some early printed tablatures *vertical alignment* of notes was sometimes problematic. For clarity, vertical lines were sometimes included between bass and treble notes to assure their being played together. While this is usually not a problem in modern printed tablature, a very crowded system can be confusing at times. To print an alignment bar between notes in tab, use a hyphen (-) between the notes. More than one hyphen can be used, thus lengthening the bar; placement closer to the upper note is more effective than placing one right at the lower note (but this is subjective). Notice that the bar affects positions within the chord stack. Example 12 shows various numbers and placements of bars, as well as some previously discussed features.

```
{Example_12:_Chord_Alignment_Bars}
-l_ "2.25_in"
b
2d_--_a
xc
xa_-_a
xc
xd-_a
xf
2\1c_---_a|
x_\3d.
xa--_a|
x\1c.
x+d-_a|
x\1c.
b
e
```

Example 12: Chord Alignment Bars



COUNTING MEASURES

Measure numbers can be handled several ways in tab. Automatic labelling can be printed *below* appropriate bar lines by inserting a run-time option (more about those later) at the beginning of a tab file. **-c** (lower case) prints below the first bar line of each system; **-C** will print below every fifth bar line; and **-CC** will print below every bar. Tab counts as a measure of music a barline followed by notes or flags: a succession of barlines, as used in repeat signs, for example, are not counted as multiple measures of music. When working with first and second endings (**A1**), check to make sure the count is as expected (see *Staff Elements*, page 6, and below in this section).

Measure numbers can also be added manually if desired by typing a numeral after the **b**: this will print a numeral *above* the system at the barline. (Tab will deliver a message about this as though you've done something wrong—it's just a control issue that has it worried.)

If your source is missing barlines, you can use *invisible barlines* to give a musically correct count of measures while having your modern edition look like the original. Add an invisible barline (counted but not printed) using **b!**. (Example 4 has an example of **!** used with a flag to make it an invisible spacer.) Also, if you wish to add a barline, but want it left *uncounted*, use **bx** (bar after 6, Example 13, page 23). These features are useful when keeping equivalent counts among several parts which may not all include multiple endings; e.g., use **bx** right before **A2** (see page 6).

Example 13 counts every measure so that the uncounted measure will be readily seen.

ORIGINAL PAGE TURNS AND LINE BREAKS

Some editors of primary source material like to include in their transcriptions indications of original line breaks and page turns. Tab can print these markings at barlines using **bv** and **BV**. The former indicates an original line break with a small v above the barline and a carat (^) below; the latter is for a page break with two stacked v's above, carat below. See Bars 3 and 7, Example 13.

SECTIONAL AND OTHER LABELS

Section labels, such as *A*, *B*, etc., can be printed using ***A**, ***B**, etc., before the bar where the label is desired.

TEXT ABOVE THE STAFF

Text above the staff can be printed using **[text]** as demonstrated in *Staff Elements* (page 6) for printing *dal Segno*. This can be used for musical directions or notes on the source, such as “ink blot”. (Comments must be ten characters or less—see Example 13 for a trick to create longer comments.)

TEXT ON THE STAFF

Tab can print directly on the staff, if desired: **T-text** at the beginning of a line in the file will print the text on the third line down. Use the [TAB] key to move the text down successive lines of the staff. [This is found frequently in the Balcarres MS.—Wayne]

TEXT BELOW THE STAFF

Printing text below the staff is handy for adding lyrics, or musical directions in the style of Alonso Mudarra. After the lowest note in a chord enter **Tyour text**. Note that there is no hyphen as used for text on the staff. In this operation the text needs to be entered after a chord or flag, so if you want text below without chord or flag, start the line with something like **4!**, an invisible flag after which tab will accept the text.

HIGHLIGHTS

Tab can *highlight* flags or notes to indicate changes from a primary source. **Q** is the highlight marker: place it *after a flag* or *before a note*. When working with highlights and notes on the top course, tell tab if **Q** goes with flag or note by using **—** as a separator: *eg.*, **2Q-d** or **2-Qd**. When highlighting a dotted flag, use **Q** *instead of a dot*; do not use **Q** (see a warning about highlighting dotted flags in the next paragraph). *Highlight a barline* with **bQ**.

Tab has *three modes of highlighting*. The default mode prints the character in gray rather than black. For a more obvious highlighting, begin the file with **-highlightparen** (tabese for “highlight parentheses”). This option will print brackets around the affected note or flag. Be warned that highlighting a dotted flag with brackets looks a little messy, since the flag and its dot are highlighted separately, making two interlocked pairs of brackets. The third mode is **-highlightred**, which prints the affected characters in red. This is a very effective highlight if you are printing in color. It can also be used in pieces like Mudarra’s where the vocal line is picked out in the tablature: red shows up better than his dots.

Example 13 uses only the bracketed highlight.

```

-p
-CC
-highlightparen
{Example_13:_More_Editorial_Stuff}
-1_"5_in"
%system 1

```

Example 13: More Editorial Stuff

```

b
T-play[TAB]here
2_a
T-[TAB]now[TAB][TAB]rest
R2

```

```

*A
bQ
6!Ttext
1_c
bv [line break]
[ink blot]
%guess flag values

```

```

2QQa_cc
3Q_0e
3_c
b
1_aTmore text
*B

```

```

b
5! [make a little space]
1@_a
2._a
b

```

```

%system 2
V_".5_in" [more space for text]
-1_"4_in"

```

```

b6
[This is]
2! [make a bigger space]
[_____too_]
2_____a
[_____long]
2_____c

```

```

bX [don't count it]
2__d
2_____a
bv [page turn]

```

```

%2nd course in ms
2-Qd
2-Qc
%bar 9

```

```

b
1_bdc
b
2__b__d
2__a__c
b
e

```

COMMENTS WITHIN THE TABFILE

Tab files can be very long and confusing when you are trying to edit them. Fortunately it is possible to add reference comments that will not be printed. Of special value are bar numbers and system numbers, but whatever helps you is fine. Begin the comment with a % in the first space of a line; tab will not try to process it. There are several samples in Example 13. *Note:* In the various tabfile samples, italicized comments within brackets to the right are not part of the tabfile, as are comments beginning with % at the left.

Troubleshooting Your Tabfile

If you run your tab file and get an error message, first check the obvious: Too many notes on one line? No end-of-file **e**? If you can't think of anything obvious, try inserting an **e** at various points in the file, running the file with each successive placement until no error is reported; that will help home in on the problem.

If there is a suspicious statement in the header (or anywhere in the file), entering % in front of it will let tab skip over it to see if the file then runs properly.

Did you run your file and find no changes? Perhaps you forgot to save first.

If your file processed properly but you want to check it for accuracy, test it by ear using tab's midi function (page 46). The tab to staff notation may also be helpful (page 44).

Occasionally a new version of tab will change some function—write to Wayne to see about a fix or workaround. Or simply experiment on your own.

Refinements to the Tablature From the Header

These refinements to the default tab output comprise a variety of fonts for both the flags and the notes, some of which are based on historical examples. These are entered in the header following a **\$** : e.g., **\$flagstyle**. If you find a particular combination of options that you like for a style, you can copy it and save it as a plain text file for use via copy-and-paste to head up future tabfiles.

FLAG AND CHARACTER STYLES

Tab has a number of different fonts for both flags and characters. These are best described through the examples which follow. Each example will combine a flag style and character style; it should be obvious which is which. These can be recombined in any way; also, numeral and character styles can both be specified; e.g., Robinson letters with Italian numeral basses and thin flags. The *standard flags* and *characters* are in fact the defaults; they generally don't need specification.

Note that **d**'s are printed several times in each sample: this demonstrates how they recline or stand upright depending on situation. When character styles with two d's are chosen, one can choose to have only one type used: enter **-allDsdwn** or **-allDsup** in the header.

```

$flagstyle=standard
$charstyle=standard
$numstyle=standard
{Standard_Flags,_Characters_and_Numerals}
-l_ "4.2_in"
b
W_a__0
w_b__1
0_c__2
1d_d
xdd__3
2_e__4
3_f__5
4_g__6
5_h__7
x_i__8
x_k__9
x_l__N10
x_m__N11
x_n__N12
b
e

```

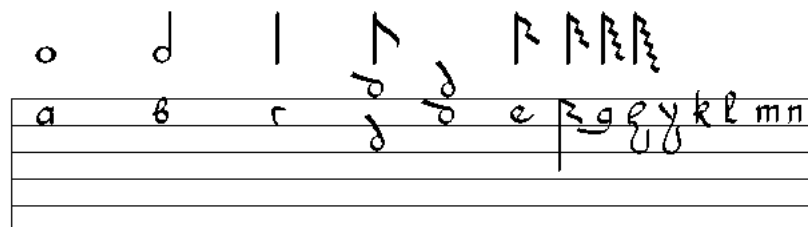
Standard Flags, Characters and Numerals

o	d		d	d	f	f	f	f				
a	b	c	d	e	f	g	h	i	k	l	m	n
0	1	2	3	4	5	6	7	8	9	10	11	12

The *Board flags* and *characters* offer a manuscript style (*Board Lute Book*) from c. 1620. Refer to *Ornaments* (pages 8 and 9) for several Board ornament glyphs and their usage. *Notes and Chords* (page 4) describes how to print the special Board characters when using the default character set.

```
$flagstyle=board
$charstyle=board
{Margaret_Board_Lute_Book}
-l_ "4.2_in"
b
W_a
w_b
0_c
1d_d
xdd
2_e
3_f
4_g
5_h
x_i
x_k
x_l
x_m
x_n
b
e
```

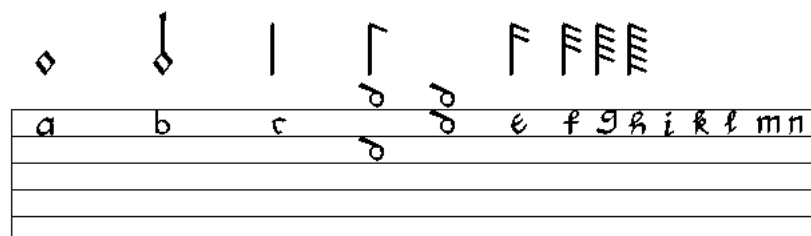
Margaret Board Lute Book



The *thin flags* and *Robinson characters* are a c. 1600 English printed style. Note Robinson's **d**'s are all upright. Robinson published *Schoole of Musicke* (1603) and *Newe Citharen Lessons* (1609).

```
$flagstyle=thin
$charstyle=robinson
{Thin_Flags,_Robinson_Characters}
-l_ "4.2_in"
b
W_a
w_b
0_c
1d_d
xdd
2_e
3_f
4_g
5_h
x_i
x_k
[etc...]
```

Thin Flags, Robinson Characters



Thomas Mace published *Musick's Monument* in 1676. The *contemporary flag* set is modern contemporary.

```

$flagstyle=contemp
$charstyle=mace
{Contemporary_Flags, _Mace_Characters}
-l_ "4.2_in"
b
W_a
w_b
0_c
1d_d
xdd
2_e
3_f
4_g
5_h
x_i
x_k
[etc...]

```

Contemporary Flags, Mace Characters

Capirola flags resemble the thin set; unfortunately tab does not offer the color coding used by Capirola's gifted scribe Vidal (c. 1515). The *Italian numerals* will print a nice uppercase "X" for the Roman numerals. The example is for six course Italian tablature.

```

$flagstyle=capirola
$numstyle=italian
$line=o
{Capirola_Flags, _Italian_Numerals}
-l_ "4.2_in"
b
W___0
w___1
0___2
1___3
2___4
3___5
4___6
5___7
x___8
x___9
x_!x_N10
x_!x&1_N11
x___N12
b
e

```

Capirola Flags, Italian Numerals

Here are the standard *Italian flags* in a seven (or more) course setting using the `—s` option. Note that the space immediately right of the “rhythm section” in the tabfile is for the bourdons. This switch creates an extra space between top line and the rhythm flags for bourdon notation.

```

$flagstyle=italian
$numstyle=italian
$line=o
-s
{Italian_Flags,_Italian_Numerals_(seven_course)}
-l "4.2 in"
b
W/0__0
w__1
0__2
1__3
22__4
3__5
4__6
5__7
x__8
x__9
x_!x_N10
x_!x&l_N11
x__N12
b
e

```

Italian Flags, Italian Numerals (seven course)

										X	XI	
0	1	2	3	4	5	6	7	8	9	10	11	12

The *small Italian flags*, it should be noted, go up to four “leaves” only.

```

$flagstyle=smallitalian
$numstyle=italian
$line=o
{Small_Italian_Flags,_Italian_Numerals_(six_course)}
-l "4.2 in"
b
W__0
w__1
0__2
1__3
2__4
3__5
4__6
x__7
x__8
x__9
x_!x_N10
x_!x&l_N11
x__N12
b
e

```

Small Italian Flags, Italian Numerals (six course)

										X	XI	
0	1	2	3	4	5	6	7	8	9	10	11	12

Baroque flags and *characters* have been discussed on page 15 of **Beyond French Tablature**. They are invoked with the header option `—b`.

STAFF SIZE (NUMBER OF COURSES)

Available in addition to different tablature fonts are staves of different numbers of lines. The default is six, but four, five and even seven line staves can be printed. The former is often used in the mandora books (Skene MS), and for four-course guitar (Mudarra, Fuenllana); it can also be used for mandolin or banjo. The five line staff was common in 16th C. publications, e.g., Attaignant, with the sixth course entered beneath the staff. Thysius uses a seven-line staff. These options are entered in the header with **-4**, **-5** and **-7**, respectively. [**-0** (zero) will print six lines, the default.] Printing Italian tablature requiring notation for more than six courses is discussed above on page 13.

General Presentation and Layout

Some general layout options have already been discussed (**Page Formatting**, pages 17-19): heading style, line breaks, page breaks, vertical spacing. These were introduced as “local” changes; tab also offers alternatives that function on a global level and are introduced in the tabfile header (some of the following can also be used as command line options). Page orientation, staff spacing, and different fonts are among those to follow.

PAGE ORIENTATION AND SIZE

The default page set-up for tab is portrait. *Landscape* (portrait rotated 90°) is also available. Near the beginning of the tabfile enter **-e**. In this orientation the default staff length is 9.0 inches. If you decide to switch to this option after you have begun your tabfile, you will need to do some cut-and-paste editing to fill the staff lines out nicely. And vice versa, if you have begun in landscape and decide you don't like it.

Tab's default page size is American letter, 8.5 by 11.0 inches. Tab can also accommodate the European standard *A4* (approximately 210 mm by 297 mm) by entering **-a4** in the header.

SYSTEM SPACING

Spacing of systems can be managed on an individual basis as needed (page 19), or it can be done globally from the file header. Enter **\$sys-skip=nm**, where **nm** is a floating decimal in *inches*; e.g., **\$sys-skip=1.2**. This example will give a spacing of 1.2 inches between systems. This may be used to better accommodate text or to balance the page layout better.

While **\$sys-skip** can add needed space, **-compressStaff** (non-variable) vertically compresses spaces in the output, sometimes allowing an additional system to fit on a page.

REDUCING SIZE OF PAGE CONTENT

The default size of the tablature printout is large, limiting how much can go on a single page. The whole works can be reduced in size to fit more notes per line and more systems per page, often eliminating awkward page turns or the number of pages on your music stand. Tab can reduce to the point where you may need to memorize the music! The various size reduction options, entered in the header, are as follows:

-R95	95%
-R9	88%
-R8	77%
-R7	66%

The length of a line in a “reduced” tab will remain at the default 6.5 (or 9.0 inches in landscape); however, note that local measurement changes introduced within the file, such as line length, are reduced proportionally with this option. That 5.125 inch line will be shorter, but it shouldn't matter unless you need a specific length to match something else; then just do some math or trial and error. Point size of characters is also proportionally reduced.

If you have a lot of notes of longer duration, which take up more space, tab can *reduce the proportional spacing* with **-Q** in the header. This allows you to fit more notes on a line.

STAFF LINE THICKNESS

If you are experiencing problems discerning the fine staff lines, this may help. The option is `–staff-line-thickness_“x.x_uu”`, where **x.x** is a floating decimal, and **uu** (units) can be **mm** (millimeters), **in** (inches) or **pt** (points). Note the spaces. Appropriate parameters are within tenths of a millimeter or hundredths of an inch; 1.0 pt is rather bold; 3.2 mm is funny. The default is 0.0057 inch, or about 0.14 mm. This option makes both horizontal and vertical lines bolder.

TEXT FONTS

Tab can print several fonts for text content of the tablature; working with them can be tricky (I’ve had Wayne explain it several times). Packaged with tab are a number of TFM files: text font metrics files. These tell tab the size and shape of the fonts. The actual fonts need to be available somewhere else for the printer to use them. Generally GhostView will have them; I’ve found that Mac’s Preview doesn’t have all of them. What difference does this make? How they appear in your finished product may depend on the fonts available to your previewing application. Preview converts the ps document to pdf before displaying it, and somehow misses some of the fonts; GhostView conversions from PS to PDF seem to be fine, also the fink text handling command *ps2pdf*. Dragging the file directly to your printer may work, but not for my HP Deskjet.

The fonts available to tab (those with tfm’s in the tab directory), have cryptic names like pncb, or ptmri. The “p” is for postscript, and the rest of the letters in blocks can be deciphered as follows:

nc = New Century Schoolbook
tm = Times
hv = Helvetica
pl = Palatino
cr = Courier
bk = Bookman
r = roman
i = italic
b = bold
n = narrow
o = oblique

...and more with combinations thereof

Pzcmi is Zapf Chancery Medium Italic, a lovely flowing font for titles (Mac’s Preview doesn’t have it!). Just look in the directory where you keep tab to see what’s available. Or, in a Terminal or DOS window, type “**tab –listfonts**” for a listing of what’s in the tab directory. The list will include both short and long names of the fonts. You may possibly create or download other tfm’s for tab to use.

There are a couple of ways to invoke these different fonts. First, you must direct tab to use something besides the default font for a particular context. Use the following in your file header to convey your intentions:

\$titlefont=fontname	Also referred to as F2 , for main title
\$titlesize=fontsize	(Default is 12 point)
\$altitlefont=fontname	Also referred to as F3 , for subtitles
\$altitlesize=fontsize	(Default is 12 point)
\$textfont=fontname	For text after a T
\$textsize=fontsize	(Default is 12 point)

Now tab knows what alternate fonts to use, but you must tell it where to use them. Enter **F2** before **{text}** that gets the new *title font*. Enter **F3** before any **{other text}** that gets the *altitle*

font, such as composer or source. The *textfont* is automatically assigned to any text following a **T**. See Example 14.

```

$titlefont=pplb
$titlesize=18
$altitlefont=pplr
$altitlesize=14
$textfont=ptmr
$textsize=10

$flagstyle=smallitalian
$numstyle=italian
$line=o

F3
{\CL/Example_14}
F2
{\CL/_1._Fantasia_de_pasos_largos_para_desenboluer_las_manos}
F3
{Tres_Libros..._I,_f._1/Alonso_Mударra_}

```

```

%Bar 18
b
1 3
x 2
x 0 5Tdedi
2 3
x 2
b
%Bar 19
x 0
x 3
x 1
x 0
x 3Tdosde
x 0
x 1
x 3
b
e

```

Example 14

1. Fantasia de pasos largos para desenboluer las manos

Tres Libros... I, f. 1 *Alonso Mudarra*

Note that there are extra spaces in the title and subtitle: for some reason, tab occasionally needs strategically placed spaces for correct alignment of titles with margins.

Another way to assign the fonts is to include *within the braces* **^01** (*textfont*), **^02** (*titlefont*) or **^03** (*altitlefont*). This method would also allow the use of the *textfont* in the heading (**F1** won't do it); thus, with the same header used above:

```

{^01\CL/Example_14}
{^02\CL/_1._Fantasia_de_pasos_largos_para_desenboluer_las_manos}
{^03Tres_Libros..._I,_f._1/Alonso_Mударra_}

```

Example 14

1. Fantasia de pasos largos para desenboluer las manos

Tres Libros... I, f. 1 *Alonso Mudarra*

COPYRIGHT NOTICE

You can have copyright information printed automatically on your tabs: Use **\$scribe=yourname** in the header along with **-G** to print the information at the bottom of each page (*except the first or only one*), as in Example 15.

-G

\$scribe=Leonardo dei Guglielmi

Example 15: Copyright Notice

will print this at the bottom of the pages: © TAB by Wayne Cripps 2005 *Leonardo dei Guglielmi*

Tab and Staff Notation

Tab was originally devised for the sole purpose of typesetting tablature: the setting of melody in staff notation is a later add-on, and lacks the flexibility and richness of tab's tablature output. Nevertheless, tab can print one or two melody staves in a variety of common keys, along with lyrics and tablature accompaniment. Even with its limitations, the output is a very readable, serviceable copy. *Nota bene*: tab will not print staff notation without a tablature staff below it!

In regular tab input for tablature, there are basically two columns: rhythm and notes. When adding a melody staff, two more columns are added: music and text, in that order. A sequence of input characters for music are initiated with an **M**, and for text with a **T**. The **M** and **T** can be replaced with a stroke of the **[TAB]** key, provided your text editor does not replace tabs with spaces! The **M** must immediately follow the tablature entry, and the **T** must immediately follow the music note entry. Here are the input sequences, followed by examples with one and two staves of music.

NOTES OF MUSIC

Each note of music is indicated by three characters denoting *duration*, *pitch* and *accidental*. All three must be used.

Duration follows the rule in tablature setting:

W = whole note	1 = eighth note
w = half note	2 = sixteenth note
0 (zero) = quarter note	L = longa (for anyone who can hold it that long)

Pitch is given by letters **A** through **G** and **a** through **g**. These cover a pitch range from A below middle c through g above the treble clef. Additionally, **?** and **@** print F and G below middle c, respectively; **1**, **2**, **3** and **4** will print a through d above the treble staff.

Accidental and *dotting* are denoted by the third character:

+ = sharp	^ = sharp plus dot
- = flat	v = flat plus dot
n = natural	N = natural plus dot
. = dot	0 (zero) = place holder

Thus **M0C^** (or **[TAB]0C^**) will print a quarter-note middle C#.

Accidentals can be printed *above the staff* using the option **-sharpUp**. The option **-sharpNorm** will return them to their positions in front of notes.

Beams and *ties* can be used to connect notes. Notes can be beamed together as in tablature using # to initiate the group and an **x** for each subsequent member of the group. For example: a beamed pair might start with **M1G0#** and end with **M1a0x**. A tie between notes is indicated with a – following the first note of the pair. (Sorry, only pairs, no lengthy slurs.) Within a beamed group, or grid, a tie is indicated by = *in place of* the **x**. Note that a tie cannot extend from the first note of a beamed group!

A *rest* is indicated with an **R**: **1R0** prints an eighth rest. Rests can be dotted.

An additional useful symbol is the *zig-zag* at the end of a staff which indicates the pitch of the first note on the following staff of music. Use **Z** for duration; hence **Zb-** will print the zig-zag at bflat, middle line.

Fermatas and *repeats* in the tablature will be duplicated in the staff notation; they cannot be printed separately in tablature and staff.

CLEF AND KEY

Clef is indicated by **G** or **F** (upper case) preceded by an **M** in the left column. These print a treble G clef or bass F clef, respectively. *Warning*: tab doesn't really like bass clef and will print bass notes according to their positions on the treble clef. You need to transpose in the tabfile. Also, tab's pitch range limits what can be done with bass clef.

MG8a or **MG8b** to will print the G clef with "8va" above or below, respectively.

Key signature is given by **k** followed by the name of one of seven keys:

kc = c	kb = b flat	kg = g	ka = a
kf = f	ke = e flat	kd = d	

If no key is indicated, the default key is c.

Key signature can be changed by entering it where necessary in the tabfile. If there will be no key changes during the piece, use **-autoKey** to set the clef and key signature on all lines as on first line.

Tab normally prints separate bar lines in the tab and music staves. They can be joined by using the option **-longbar**. The resulting bars don't always respect text between the staves, requiring the judicious use of **M000** as a spacer.

TIME SIGNATURE

Time signatures are entered in the left column and appear in both tablature and staff notation.

STYLE

The default note style is modern. Using the switch **-D** in the header (also see command line options, page 45) will yield output resembling a Dowland era lute song book. Each style has both good and weak points. If different styling is desired in the tablature, this can be accomplished with **\$-** switches discussed on pages 25-28.

Here is a bit of summary comparing the two styles:

Standard mode prints modern notes, but its thirty-second notes are of a different scale and its sixteenth rests print as the Elizabethan period hook. High (upper treble clef) beamed notes do not print inverted, and look a little strange up there. Tab flags are modern.

Dowland mode prints the Italian style diamond head notes, including thirtysecond notes. It will not print beamed notes, or grids; the coding for these are simply ignored. Tab flags are the thin variety. The *8va*, if used, prints quite separate from the clef sign. The natural sign looks a bit modern (did they have one then?). Bar lines do not print at the right ends of music staves, a feature of the Dowland songbooks. This mode can produce a nice facsimile-style output.

Both modes have occasional spacing problems which can be overcome with the use of **M000** in the music or an "invisible" tab flag (**2!**) in the tablature. Usage is demonstrated in the examples.

The following examples have staff notation only (with blank tab staff). An example with both text and tablature will follow a description of the inclusion of text.

`$sys-skip=.25 [make room for high notes at end]`

`{Staff_Notation:_Modern_Notes}`

`-l_"5.0_in" [adjust line length]`

`b`

`MG`

`kg [1 sharp]`

`SC [time signature]`

`M000 [extra space]`

`MW?0`

`b`

`Mw@0`

`M0A0`

`M0R0`

`b`

`M1B-# [start beam]`

`M2C+= [tie]`

`M2D0x`

`M1E0#`

`M1Fnx`

`MwR0`

`b`

`MWR0`

`MZG0 [zig-zag]`

`b`

`-longbar [through bars]`

`b`

`MG8b [down an octave]`

`kd`

`Sc`

`M000`

`M0G.`

`M1a0`

`M0bv`

`M1c0`

`b`

`M000`

`M0d^`

`M1e0`

`Mwf0- [tie]`

`b`

`M0g0`

`M0R. [dotted rest]`

`M210-`

`M220`

`M1R0`

`M230- [these needed`

`M240 more room]`

`MZd0`

`b`

`e`

Staff Notation: Modern Notes

Note that barlines (as well as repeats) are copied from tablature to staff and need not be entered separately in the music.

-D *[Dowland style]*
 \$sys-skip=.25 *[make room for high notes at end]*
 -sharpUp *[accidentals above]*

{Staff Notation: Dowland style}

-l "5.0 in"

b

MG

kf

S4

M000

[spacer]

MW?0

b

Mw@0

MOA0

MOR0

b

M1Bn#

[no beams

M2C0x

in Dowland]

M2D0x

M1E0#

M1F+x

MwR0

b

MWR0

MZG0

b

Staff Notation: Dowland style

-sharpNorm

[accidentals down]

b

MG8a

[octave up]

kb

S3

M000

[spacer]

M0Gv

[accidentals...

M1a0

M000

M0bN

...with...

M1c0

M000

M0d^

...dots]

M1e0

b

Mwf.-

M0g.

MOR.

b

MWR0

MOR0

M310

M320

M230

M240

M2R0

[“hook” rest]

MZd0

b

e

TEXT WITH STAFF NOTATION

As noted in the introduction to this section, the column of the tabfile devoted to text, or lyrics, is begun with **T** immediately following the **M** entry. As an alternative, a **[TAB]** can replace the **T**. The spaces created with the spacebar are indicated, as previously, with `_`; the larger spaces are created with the **[TAB]** key.

```
{Text_Entry}
```

Text Entry

```
-l_"4.0_in"
b
M0G0Ttext
M1G0Twith
M0G0T"T"
M1F+Tand
b
M1D0- text
M1E0
M1F+ with
MwG. \[_TAB\]
b
e
```

Special text characters and punctuation are “spelled” as explained on pages 17-18. Find there the explanation of the unusual spelling of “[TAB]” in the sample tab file. [The crowding of text upward against notes will be addressed below.]

The next step is, of course, to put tablature, music and text together. The easiest way to go about the project (opinion!) is to take it one step at a time: lay out the tablature; back to the top and add in the staff notation; finally, *da capo* once more to do the lyrics or other text.

Because tab interprets **M** and **T** in the same way as it does a **[TAB]** keystroke, I find it useful to use the latter to arrive at a visually more comprehensible tabfile. This makes checking it easier later on. Of course, since the start of a tabbed column is dependent on where the previous line of type ends, things don’t always line up 100%--but most of the time they do.

Something to remember about the **[TAB]** key is that it functions as a **[RETURN]** key within the music/lyrics columns of the tabfile. So, if, as in the example above, text is crowded from above by the staff notation, an extra **[TAB]** immediately after the **T** (or in addition to the **[TAB]** substituting for **T**), will create an extra space between staff and lyrics. Likewise, an additional **[TAB]** at the end of the lyrics entry will create an extra horizontal space between lyrics and tablature flags to avoid crowding at the bottom. The latter need be entered only once for each line of music.

```
{Text_Entry_with\_ \[_TAB\]\_'s}
```

Text Entry with [TAB]’s

```
-l_"4.5_in"
b
1aM0G0T[TAB] text[TAB]
1aM0G0T with
2eM1b0T "T"
2cM1a0T and
1aM0G0T "M"
b
2_a[TAB]1D0-[TAB][TAB]text
2_c 1E0
1_e 0F+ with
0_a wG0 \[_TAB\]\_'s
b
e
```

Note that spaces created with the space bar have been denoted throughout with `␣`. I have indicated only those **[TAB]**'s which introduce a new usage ; the remaining ones simply line up beneath those as wide spaces in the sample..

The default text font is a 12 point regular roman style. To change it, or its size, (using **\$textfont=** and **\$textsize=**), see page 30.

A few more items that work in the staff notation area:

In the text section of a row of the tabfile, text preceded by `^` is placed above the staff in the output if only a single melody staff is being printed..

In text, a word followed by `|` will be butted up against the following word in the output.

Also in text, the symbol `@` will print the repeat sign \curvearrowright representing repeated text.

In text again, a `]` after a word (or syllable) will create a tie to the following word: a little concave-down joining the words.

Tab and staff pieces can have the tablature converted from French to Italian, and *vice versa*, by following the directions on page 14.

Text below the tablature staff, as outlined on page 22, is not possible when music saves are inserted. It is possible to print text on the staff with **T-text** (also page 22), using **[TAB]**'s to push the text down, but alignment with notes will not be exact.

As tab replicates barlines and repeats from the tablature to the music staff, so it does with ties between notes in the tablature. This is not always a welcome occurrence, but there is a workaround: in the tablature, connect the notes in question with an underline instead of a tie. This has been done in the two stave example on pages 42 and 43.

The example which follows (beginning on page 38) is a complete song (though perhaps not all the verses!). Hopefully it will demonstrate the essential elements of coding for tablature with staff notation. In a toss-up decision with the aim of visual clarity in the tabfile, I have chosen to use mainly **[TAB]**'s in place of the **M** and **T**. However, I have also chosen not to delineate spaces as opposed to **[TAB]**'s; simply recall that the latter are not used in tablature, and spaces in the text will be obvious by requirement of the wording. The tabfile itself has been broken into two columns for economy of space.

TWO STAVES OF MUSIC WITH TABLATURE

To print two staves of music above a tablature staff, include **-twostaff** in the tabfile header. The order of data entry in a row of the tabfile is as follows:

```
tablature[TAB]music1[TAB]music2[TAB]text1^text2
```

Note the carat `^` separator between the two text entries. **music1** and **text1** will be the lower of the two staves; **music2** and **text2** are at the top of the system. The input sequences are the same as for a single staff of music.

An example of two staves of music notation appears on pages 42 and 43. Anyone with masochistic tendencies may wish to finish the piece!

```

-D
-R7
-C
-sharpUp
-autoKey
$title=16
$sys-skip=0.25
$textfont=pplri

{\CL/Tant que vivray}
{}
b
MG
kf
Sc
3!
0 d a wc0    Tant  Par^Soprano
1 daa 0c0    que  plus-
x abc 0b0    vi-  sieurs
b
0 dda wa.    vray  fois
1 dda
x dfa 0a0    en   m'a
b
x c a 0G0    \^a- te-
xa d 0c0     ge  nu
2aa c 0c0    flo- lan-
x a
1aa c 0bn    ris- guis-
b
2 cd awc0    sant, sant,
x c
x d
x a
x cd awR0
x b
x a
x c
b
x dda wa0    Je   Mais
x e
x a
x c
1 dda 0a0    ser- a-
x abc 0G0    vi-  pr\`es
M ZF0
b

%Line 2

b
2 aa cwF.    ray  deuil
x a
x c
x e
1 aa c
x aa c 0F0    d'a- m'a
b
1. df d      0b.  mour  faict
2 cd 1a0     le  r\`e-
1 ab d       0G0  dieu  jou-
x a 0F0     puis- ys-

```

```

b
0 dca WE0    sant  sant,
1 dc
x a
b
0 def wF0    En   Car
1 aa c 0F0    faictz, j'ay
x abc 0G0    en   l'a-
b
2 dda 1a0#    ditz, mour
x b 1G0x
x a 1F0x
x c 1E0=
1 def 0F0
x ab d       0G0  en   de
M ZG0
b

%Line 3

b
1 aa c 0G0    chan- la
x bac 0F0    sons  belle
x dca 0F0    et   au
x a 0E0     ac-  gent
b
3!
Y def WF0    cords. corps.
3!
.
b
B
b
.
1 ab 0F0     Son
2 ab 1F0#    a-
x ab 1G0x    li-
1 cd 0a0     an-
x d a 0a0    ce,
b
x ab 0F0     c'est
2 ab 1F0#    ma
x ab 1G0x    fi-
1 cd 0a0     an-
x d a 0a0    ce,
b
2 a 0c0     Son
x a
x c a 1c0#    c\oeur
x a 1b0x     est
0 da wa0     mien,
M Zc0
b

```

tabfile continued on page 40

Tant que vivray

Soprano

Tant que vi - vray en â - ge flo - ris - sant, je ser - vi -
Par plus - sieurs fois m'a te - nu lan - guis - sant, Mais a - près

5

ray d'a - mour le dieu puis - sant En faitz, en ditz, en de
deuil m'a fait ré - jou - ys - sant, Car j'ay l'a - mour de

10

chan - sons et ac - cords. Son a - li - an - ce, d'est ma fi - an - ce, Son cœur est mien,
la belle au gent corps.

15

Le mien est sien, Fi de tris - tes - se, Vi - ve li - es - se, Puis - qu'en a - mours

Puis - qu'en a - mours a - tant de bien.

20

%Line 4

b
 2 a 0c0 Le
 x a
 x c a 1c0# mien
 x a 1b0x est
 0 da wa0 sien,
 b
 1 ab 0F0 Fi
 2 ab 1F0# de
 x ab 1G0x tris-
 1 cd 0a0 tes-
 x d a 0a0 se,
 b
 x ab 0F0 Vi-
 2 ab 1F0# ve
 x ab 1G0x li-
 1 cd 0a0 es-
 x d a 0a0 se,
 b
 x daa 0c0 Puis-
 2 daa 1c0# qu'en
 x abc 1b0x a-
 0 dda wa0 mours
 M Zc0
 b

%Line 5

-l "5.0 in"
 b
 1 daa 0c0 Puis-
 2 daa 1c0# qu'en
 x abc 1b0x a-
 1 dda 0a0 mours
 x cd a 0G0- a-
 b
 x aa c 1G0#
 M 1F0x
 x ba wF0 tant
 x dca
 x dc 0E0 de
 b
 3!
 Y def WF0 bien.
 3!
 .
 b
 B
 e

Here are the lines 4 and 5 of the piece again for easier reference:

Le mien est sien, Fi de tris - tes - se, Vi - ve li - es - se, Puis - qu'en a - mours

Puis - qu'en a - mours a - tant de bien.


```

$flagstyle=thin
-autoKey
-twostaff
-compressStaff [make it fit!]
-R8
{\CL/Three Blind Mice}
b
MG
kc
i
S6-8
1. e wR. wR.
1. c
b
0. a wR. wR.
b
1. e wR. wR.
1. c
b
0. a wR. wR.
b
1. c 0E. wR. Three[tab]
1 a 0D. blind
2 a
b
0. e wC. wR. mice.
b
1. c 0E. wR. Three
1 a 0D. blind
2 a
b
1. e&{MwC. wR. mice.
1
2 c"}
b
1 d 0G. 0E. See ^Three
2 d
x c 0F0 0D. how^blind
x a
x c 1F0 they
b
1 d wE. wC. run!^mice.
2 c
1 c
2 c
b
2 d 0G. 0E. See^Three
x d
x d
x c 0F0 0D. how^blind
x a
x c 1F0 they
b
1 d 0E.- wC. run!^mice.
2 c
x c 0E0
x c
x c 1G0 They
b

```

```

b
1 d 0c0 0G. all ^See
2 d 1c0 ran
x c 1b0 0F0 af-^how
x a 1a0 ter
x c 1b0 1F0 the^they
b
x d 0c0 wE. far-^run!
x c
x c 1G0 mer's
1 c 0G0 wife,
2 a 1G0 who
b
1. e 1c0 0G. cut^See
M 1c0 off
M 1c0 their
1. c 1b0 0F0 tails^how
M 1a0 with
M 1b0 1F0 a^they
b
0. a 0c0 0E.- carv- ^run!
M 1G0 0E0 ing
M 1G0 knife.
M 1G0 Did
M 1G0 1G0 you^They
b
1. e 0c0 0c0 e-^all
M 1c0 1c0 ver^ran
1. c 1b0 1b0 see^af-
M 1a0 1a0 such^ter
M 1b0 1b0 a^the
b
0. a 1c0 0c0 sight^far-
M 1G0 in
M 1G0 1G0 your^mer's
M 0G0 0G0 life^wife,
M 1F0 1G0 as^who
b
B
e

```

[Space]'s have been used in the tablature entries, and [tab]'s in the music entries. Note the use of a [tab] after text1 entry at the beginning of each new system to reduce vertical crowding of text and flags.

Three Blind Mice

Three blind mice. Three blind mice.

e r a e r a r a a e r a a e

Three blind mice. Three blind mice.

See how they run! See how they run! They

See how they run!

See how they run! See how they

all ran af - ter the far - mer's wife, who cut off their tails with a

run! They all ran af - ter the far - mer's wife, who

carv - ing knife. Did you e - ver see such a sight in your life as

a e r a

CONVERTING TABLATURE TO STAFF NOTATION

Tab can print your tabfile in *grand staff* notation, or in *guitar* notation (single staff, octave up), using the option **-m** or **-guitar**. The results are rudimentary, with no means of refinement, but are useful in checking the tablature against a standard notation source.

First, tab needs to know how the instrument is tuned. Default uning is renaissance lute in g. However, other tablature tunings can be used by entering in the header **-tuning_x4y4z3...** from lowest to highest. Some standard tunings are listed below; others can be created using the same format (note the space).

-tuning_c4d4e4f4g4c3f3a2d2g2	renaissance (default)
-tuning_e4a3d3f+3b2e2	guitar (lute tuned)
-tuning_e4a3d3g3b2e2	guitar
-tuning_c4d4e4f4g4a3d3f3a2d2f2	d minor
-tuning_c4d4e4f+4g4a3d3g3b2d2f+2	sharp
-tuning_c4d4e-4f4g4a3d3g3a+2d2f2	flat

Tab seems to recognize twelve courses in assigning staff notes to tablature notes.

A few simple rules must be followed for tablatures other than French: 1) Tabs using numbers for notes in the tabfile must include the option **\$numstyle=italian**; this would apply to Italian and modern guitar (Milán) formats. 2) Italian tab can be for six courses only. 3) Guitar, or Milán, tab must use the additional option **-milan**.

For the *grand staff* transcription, enter **-m** in the header, or type **tab_-m_tabfilename** in a Terminal or DOS window. (See more about using command line options like this one under that heading on page 42). If the tabfile was created as a guitar tablature, use **-m** with **-milan** in the header, or type **tab_-m_-milan_tabfilename** at the command prompt.

To create a single staff *guitar notation* output, use **-guitar** instead of **-m** in the above entries.

Here are the first four bars of the tablature from *Tant que vivray* in Grand Staff and Guitar notations:

-m [or **-guitar**]

```

b
0 d a
1 daa
x abc
b
0 dda
1 dda
x dfa
b
x c a
xa d
2aa c
x a
1aa c
b
2 cd a
x c
x d
x a
x cd a
x b
x a
x c
b
e

```



Some additional observations on tab's staff notation function:

Bourdons: tab will not print accidentals for them, nor will it print the correct pitch of a fretted bourdon; *e.g.*, a "c" fret on a D bass will print a s a D rather than an e.

The lowest note recognized with **-m** is Bb below bass clef (b-4); a4 will print, but without ledger lines.

Tab's main function is the printing of music in tablature form; the staff notation utility is not its intended forte.

Command Line Options

Command line options are entered upon running tab on a file. In the Terminal or DOS window:

tab_-[option]_tabfilename

Some command line options have already been introduced above: those elements in the header or body of the file that begin with a hyphen (-) : measure counts (**-c**, **-C**, **-CC**), page numbers (**-p**), general reduction (**-Rxx**), page orientation (**-e**, **-a4**), etc. These are some of the most useful options which provide alterations in tab font or page layout without making alterations to the tabfile. However, the output of a very involved tabfile may not suffer some options lightly: check your outfile before printing it!

A particularly useful command line option is **-o** (lower case). Whereas tab's habit is to name all of its output "**out.ps**", this option allows you to give it a different name. This is great if you are trying out several options and want to compare results; tab will normally simply overwrite the previous out.ps leaving no basis for comparison. Here is its usage:

tab_-o_outfilename_tabfilename

Just as tab assumes ".tab" as the extension on the input filename, it automatically appends ".ps" to the outfile.

A string of options might look like this:

tab_-e_-p_-C_-R9_-o_musettetest_musette

Your Musette will then come out in a file named "musettetest.ps" in landscape orientation with pages numbered, every fifth bar labelled, and the whole works reduced to 88% of the original.

Wayne's own ReadMe file included with the tab source code lists every option available, with a description. Some can be incorporated into the header of the tabfile, others are for use only at the command prompt. The ReadMe packed with the latest tab version is included as the *Appendix* to this guide.

Tab Midi

A very useful utility of tab is the creation of midi files from tab files. The play-back is a nice editorial tool; for practicing it gives some of us a chance to hear how it's supposed to sound! The midi function works from the command prompt, and requires tuning instructions if the tabfile is for other than the default Renaissance lute (see page 44). Numeric entry tabfiles must have in the header **\$numstyle=italian**. Since tab's midi function automatically reads numbers as Italian tab, if the tabfile is entered in Milán's style (modern guitar tab), add **-milan** to the command line, as below.

At the command prompt enter this:

```
tab_-midi_tabfilename_>_midifilename.mid
```

or

```
tab_-milan_-midi_tabfilename_>_midifilename.mid
```

Tempo can be controlled using the header option **\$tempo=n**, with the default value being 2. Larger is faster.

You can experiment with the *instrument* or *timbre* of the output using **\$midi-patch=n** in the header, with **n** being a number from 0 to 127. You can also invoke this feature at the command prompt using **-\$midi-patch=n**. Wayne has suggested the following choices, the results seem to vary with the midi playback utility in use:

```
“000 - 005 pianos ->006 hpsichord 010 other keyboards
 014 marimba
 016 tube bell
 030 nyguitar
 032 acguitar
 066 voices
 0100 soprano sax
 0110 piccolo
 0112 recorder
(note - these are octal numbers – 1008 = 6410 )
```

Currently there is no way to transpose the pitch other than working with the -tuning option above.” [123 rates a special recommendation.]

A word of warning: out-of-place characters in the tabfile will stymie the midi function, even if the typesetting function doesn't mind. For example, an earlier version of tab required a work-around to make multiple ending symbols print properly, but I had used characters (6! – an invisible and non-existent six flag rhythm symbol) which tab didn't recognize. It printed everything, but couldn't get past that point in midi generation. [The problem has since been fixed]. “The moral,” says Wayne, “is that if you tell me about problems rather than working out your own hacks it may well save trouble in the long run.” Anyway—if the midi isn't working, check for odd stuff in the tabfile. Another example is the **i** and **j** in the rhythm column used for spacing. Any such item can be temporarily commented out with a **%** for the purpose of generating the midi.

Index

Accidentals (staff notation)	32	Flags, tablature (con'd)	
Alignment bars	21	styles	25-28
ASCII tab	16	tied	2
Attaignant	29	triplets	3, 12
Bar line	6	values	2
fermata	6	Flourish	6
heavy	6	Fonts	
invisible	3, 21	tablature	15, 25
music	34	text	30, 31
numbered	21	French tablature	1
repeat	6	convert to Italian	14
tie	6	Fuenllana	29
Baroque	3, 15	German tablature	15
Bars (see measures)		Graces (see Ornaments)	
Basses	4, 15	Grid	3
Beams		Guitar	
music	33	staff	44
tablature	3	tablature	14
Board, Margaret	4, 26	Header	1, 24, 25
Bourbons (see Basses)		Heading	1, 17
Break		Highlight	
line	12, 18, 21	bar line	22
page	18, 21	flag	22
Capirola	8, 27	note	22
Characters, special (see Text)		Hyphen	4, 6, 8, 22
Chords	4	Indent	6
Clef	33	Invisible elements	2, 6, 10, 21, 46
Command line	45	Italian tablature	13
Comments	24	convert to French	14
Copyright notice	32	Key signature	33
Courses, number of	29	Labels	22
D.C., Da Capo	6	Layout	17-19, 29
dal Segno	6	Lines, staff	
Dot		notes between	13
music notes	32	notes on	13
tablature flags	2, 4	number of courses	29
fingering	20	Mace	27
Dowland	4, 9, 33	Measures, counting	21
Editorial notations	20	Melody	32
Endings, 1 st , 2 nd	6	Mesangeau	6, 8
Fermata	6	Meter (see Time signature)	
Fingering		Midi	46
left hand	20	Milán tablature	14
right hand	20	Mudarra	10, 29
Flags, music (see Notes, music)		Music (see staff notation)	
Flag styles		Notes, music	32
music	33	accidentals	32
tablature	25-28	dotted	32
Flags, tablature		pitch	32
beam	3	values	32
dotted	2, 3		
invisible	33, 46		

Notes, tablature	4	Time signature	
held	10	music	33
Numbers		tablature	6
bar lines	21	Triplets	3, 12
basses	15	Troubleshooting	24
fingering	20	Underlines	10
Operators		Vihuelists	6, 13, 14
infix	8	Zig-zag	33
postfix	8, 10		
prefix	8, 10		
Ornaments	8		
Pagination	19		
Repeat			
music	34, 37		
tablature	6		
Rest			
music	33		
tablature	4		
Robinson	26		
Skene MS	29		
Spacing			
horizontal	4, 18, 16, 29, 33		
proportional	29		
vertical	19, 29, 36		
Special character (see Text)			
Staff			
line thickness	30		
number of courses	29		
size	29		
Staff notation			
grand staff	44		
guitar staff	44		
with tablature	32		
Staves, music	32, 37, 44		
Style options			
characters	25-29		
flags	25-29		
layout	17-19, 29		
music	33		
tablature	13-16		
System parameters			
length	18		
vertical spacing	19, 29		
Text			
formatting	17-18		
music	36, 37		
special characters	17-18		
tablature	22, 37		
Thysius	29		
Ties			
music	33, 37		
tablature	2, 37		
text in music	37		