

Peter Yarensky's abc 2.2 Help File

abc File Structure.....	1
File Headers & Tune Headers.....	1
Important Information Fields.....	2
Determining Which Fields Get Printed.....	3
<i>The Default Print Settings</i>	3
<i>Modifying the Print Settings</i>	3
The Body of the Tune.....	4
Pitch Notation.....	4
Changing Meter, Key, or Default Note Length Mid-Tune.....	5
An Illustration: The abc Code for the Notating Pitch Example.....	5
Note Length, Timing, Rhythm & Rests.....	6
Musical Structure: Beams, Line Breaks, Bar Lines & Repeats.....	6
Parts Notation: Handling Variant Orders of Parts, Repeats and Endings.....	7
Modifying Notes: Ties & Slurs, Grace Notes & Triplets.....	9
<i>Ties & Slurs</i>	9
Accents, Decorations, Redefinable Symbols & Special Characters.....	9
<i>Special Characters</i>	10
Chords & Unisons.....	11
<i>Multiple Noteheads Per Stem</i>	11
<i>Chord Symbols</i>	11
<i>Using Voice Overlays with Unison Drones</i>	12
Page Layout and the Use of Text in abc Documents and Scores.....	13
Text in abc Documents.....	13
<i>Text Outside the Staff</i>	13
<i>Text Inside the Staff</i>	13
Blank Space in abc Documents.....	14
Blank Space in the Resulting Score.....	15
Printing & The Use of Fonts in abc Notation.....	16
<i>Some Issues Related to Printing</i>	16
<i>Use of Fonts With abc Notation</i>	17
<i>An Example To Illustrate Many of These Concepts Together</i>	18
<i>Text to the Left of an Indented Staff</i>	19

<Continued on the next page>

EasyABC Shortcuts	20
Musical Notation Shortcuts	20
<i>Bar Lines</i>	20
<i>Automatic Upper/Lower Case:</i>	20
<i>Add Matching Right Symbol:</i>	21
<i>Some Additional Shortcuts</i>	21
<i>Playback Shortcuts</i>	21
Playable Tunes: Combining abc Notation, CSS & HTML Code	21
An Introduction to Playable Tunes on the Internet	21
<i>The Big Picture</i>	22
<i>Setting Up The HTML File</i>	22
<i>The abc Code</i>	23
Making abc and CSS Work Together	23
An Example	24
Some Additional Useful Links	27

Peter Yarensky's abc 2.2 Help File *version 2.0*

This help file is intended to answer many questions about abc notation. It is centered around the as yet incomplete **abc 2.2 standard** but most of it should apply to the more widely used **abc 2.1 standard**. I wrote this gradually over a period of a few years for my own use but it occurred to me others might find it useful too. Parts of it paraphrase what's in the published standard <<http://abcnotation.com/wiki/abc:standard:v2.2>>. Some of it I wrote to highlight topics not so prominent in the standard or to give illustrations of various topics. The choice of topics is strongly influenced by issues that have caused me confusion in writing out abc notation for fiddle tunes.

- ▶ I expect it will be useful as a reference, to find the answer to a question, or to learn more about a topic that is covered here.
- ▶ It is *not* intended to be used to learn abc notation.
 - ★ For that I'd recommend [How to get started with abc notation](#) on the [abc Notation Home Page](#).

abc File Structure

All abc tunes share a common structure. A tune should begin:

- ▶ %abc-2.2 (or 2.1); This is optional but it causes strict interpretation according to the abc standard which may be important if you use any abc code that's specific to a particular version.

All abc documents should have the extension `.abc` (all lower-case)

File Headers & Tune Headers

A file may have a **file header** at the top to set default values for a tune.

- ▶ It is likely to contain **stylesheet directives** to specify the layout and other

This abc help file was written by Peter Yarensky <peter@pyarensky.com>. This version is © 2022. You may download a PDF version. It may be printed, and it may be distributed as long as it is unchanged. I hope you find it useful. ~ Peter

variables.

- ▶ Stylesheet directives begin with %% and are followed by the name of a parameter (e.g. `titlefont`) and the value of the parameter (e.g. `Palatino 18`). Thus they take the form:

```
%%parameter_name value
```

- ▶ The header is defined by a blank line separating it from the **body** of the tune.

Each tune has a **Tune Header** with **Information Fields**. These identify the tune and describe its characteristics.

- ▶ The first two should be X & T; the last should be K.

Important Information Fields

- X: Number *[required]*
- T: Title (second title prints smaller) *[required]*
- S: Source
- C: Composer
- F: File URL
- G: Group (for categorizing tunes for sorting, etc.)
- H: History — notes, stories about the tune; designed for multi-line entries
- O: Geographic Origin
- N: Note — general annotation
- R: Rhythm (in words)
- Q: Speed of playback
- M: Meter (in numbers) — M: none or omit M field for free meter
- L: Length of default note *[required]*
- U: User defined (e.g. U: T = !trill!)
- Z: Transcription (who, and contact information)
- K: Key (must be last) *[required]*
- +: Continues lines. For example, H: followed on the next line by +: is considered a single entry

Determining Which Fields Get Printed

The Default Print Settings

By default the following fields get printed:

- T: (Title) — centered above the tune; second title under it in smaller print
- C: (Composer) — on the right just below the title
- O: (Origin) — in parentheses following Composer
- P: (Part) — If there is a Part field in the header describing the parts, each part should be printed immediately before the part starts, and the order of parts should be at the top of the tune on the left
- Q: (Tempo) — above the tune at the start of the section to which it applies
- w: (Words) & W: (Words) — under each line or under the tune respectively

Modifying the Print Settings

It is possible to (1) print additional fields, (2) suppress printing of default fields, and (3) modify how fields are printed.

Printing is modified with the `%%writefields` directive. Its form is:

```
%%writefields <list of field identifiers> [<logical>]
```

The default setting is `%%writefields TCO PQwW` which means to print each field identified, in the order listed (Title, Composer, Origin, etc.) Subsequent `%%writefields` directives *combine* with this list rather than replacing it.

- ▶ Thus `%%writefields Q false` means to suppress printing of the Q (Tempo) field, and `%%writefields N` would add the N (Notes) field to the list to be printed.

There are a couple other ways to format the Information Fields and determine which will be printed. One method is to use the `%%titleformat` directive. This is a fairly complex but powerful directive that determines which fields are printed and where they are placed (center, left align or right align). When used the `%%writefields` directive isn't used.

Further modifications may be made using the **font directives** and other relevant

formatting directives. For example, the following causes the History field to be displayed and sets it in Georgia 14-point text.

```
%writefields H
%historyfont Georgia 14 % Fonts will be discussed in
more detail near the end.
```

The Body of the Tune

Pitch is described by letters; the **duration** of a note by numbers.

Pitch Notation

Here are two ways of relating pitch as described in abc notation to other standard descriptions of pitch. First, a table:

abc Notation	Description	Scientific Pitch Notation	Helmholtz Pitch Notation
C, to B,	octave below staff (G clef)	C6 to B6	c''' to b'''
C to B	octave in lower part of staff	C5 to B5	c'' to b''
c to b	octave in upper part of staff	C4 to B4	c' to b'
c' to b'	octave in the positions	C3 to B3	c to b

In abc notation, additional commas/apostrophes raise or lower pitch by an octave. Be sure to turn off smart quotes.

Accidentals: ^, = and _ before a note produce sharp, natural or flat symbols.

Another way to look at it is in terms of the keyboard:

Notating Pitch

G, A, B, C D E F G A B c d e f g a b c' d' e'

The image shows a musical staff with four measures. Above the staff, the notes G, A, B, C, D, E, F, G, A, B, c, d, e, f, g, a, b, c', d', e' are aligned with the notes on the staff. The time signatures are 3/8, 7/8, 3/8, and 3/8. The notes are: G (quarter), A (quarter), B (quarter) in 3/8; C (quarter), D (quarter), E (quarter), F (quarter), G (quarter), A (quarter), B (quarter) in 7/8; c (quarter), d (quarter), e (quarter), f (quarter), g (quarter), a (quarter), b (quarter) in 3/8; c' (quarter), d' (quarter), e' (quarter) in 3/8.

Changing Meter, Key, or Default Note Length Mid-Tune

Many tunes include key changes. In a crooked tune there are mid-tune changes in meter. These can be handled by making a new line in the abc code for notating the change or by including the code within a line of music.

- ▶ **Use of a New Line.** Start a new line of abc code with a K:, M: or L: field. Then start the music on another new line.
 - ★ This would be appropriate, for example, to indicate key changes between the parts of a tune.
- ▶ **Within a line of music.** At the beginning of the measure where the change occurs, put the K:, M: or L: field in brackets.
 - ★ For example, in a crooked tune different measures might switch between 4/4 and 6/4 meter. This could be indicated by [M:4/4] and [M:6/4].
- ▶ To keep the printed music on same line, use \ at end of previous line of music.

An Illustration: The abc Code for the Notating Pitch Example

Here is the abc code for the *Notating Pitch* example. It illustrates (1) the structure of abc code, (2) how it corresponds to the standard musical notation, and (3) changing meter during the tune.

- ▶ The confusing array of letters, single and double quotation marks is the note name in double quotation marks for display above the staff followed by the same note name to produce the notation. Thus "C"C"D"D is "C" C "D" D.

```
X:1
T:Notating Pitch
L:1/8
M:3/8
K:C
"G,"G, "A,"A, "B,"B, |\
M:7/8 % The first method for changing meter
"C"C "D"D "E"E "F"F "G"G "A"A "B"B |\ % Below, the second method
"c"c "d"d "e"e "f"f "g"g "a"a b"b | [M:3/8] "c'"c' "d'"d' "e'"e'|]
```

- ★ *Note:* The note names are treated as chords. Technically they should be annotations, but as such they were displayed at different heights which seemed less desirable.

Note Length, Timing, Rhythm & Rests

Note Length: The **unit note length** is the length of a note specified by a letter with no modifier. It is specified in the tune header.

- ▶ For example, L:1/8 would mean that unless otherwise specified each note would be an eighth note.
- ▶ It's generally best to choose a unit length so that most of the notes in the tune are that length or longer.

To change the note length, do one of the following.

- ▶ A number after the note is a multiplier for the length specified in L:
 - ★ e.g. for L:1/8, A2 is a quarter note; for L:1/4, A2 is a half note.
- ▶ A [/] + a number after a note divides the default duration.
 - ★ e.g. for L:1/8, A/2 is an eighth note; for L:1/4, A2 is an eighth note.
 - ★ A shortcut for A/2 is A/.

If the default note length isn't specified, it is computed automatically. This may or may not be the best length, so it's best to avoid this.

- ▶ To find default note when not specified: divide meter; e.g. 2/4 meter divides to 0.5. If the result is <.75, default = 1/16; if ≥.75, default=1/8.

Dotted Rhythm: Put a < or > between two notes: e.g. A2>B2 = A3B.

Rests: Symbolized by z or x. Duration described as with regular notes.

- ▶ z = visible rest, X = invisible rest
 - ★ Invisible rests are useful for making the playback of a tune correct.

Musical Structure: Beams, Line Breaks, Bar Lines & Repeats

Beams: Group notes without spaces between them to beam them.

Line Breaks: By default a line break occurs after each line of abc code, although that may be modified.

- ▶ \ = *continue* the line with what's on the next line of abc code

- ★ See *Notating Pitch* (previous page) for an example.

Bar Line & Repeat Symbols: The following is adapted from the abc 2.2 standard.

- | bar line
- |] thin-thick double bar line
- || thin-thin double bar line
- [| thick-thin double bar line
- |: start of repeated section
- :| end of repeated section
- :: start & end of two repeated sections

- ▶ **First And Second Repeats:** begin the measure with [1 or [2.
 - ★ If adjacent to bar lines, just |1 or |2 with no spaces.
 - ★ Thus, the following is correct: |1 <notes> :|2 <notes> |]
 - ★ Whereas the following is not: | 1 <notes> :| 2 <notes> |]

Parts Notation: Handling Variant Orders of Parts, Repeats and Endings

Unusual or complex orders of parts, repeats and endings may be handled using the P: (Parts) field. It may be used in the **Tune Header** and again in the **Tune Body**. The use of Parts notation tells both the reader and the abc software the correct sequence for playing the tune.

- ▶ **Order of Parts.** This is handled in two ways.
 - ★ Order is specified in the tune header. For example, for *Chorus Jig*:
P: ABCB % *Play parts in the order ABCB*
 - ★ In the body of the tune, parts may be identified when useful or even necessary. For example:
P: C % *Identifies the beginning of the C part*
- ▶ **Repeats.** Complex repeats are much easier to set up and display using Parts notation in the Tune Header. This is done through a combination of numbers and parentheses.
 - ★ **Numbers** following a Part letter: number of repeats of that part.
 - ★ **Parentheses** are used to repeat entire sequences of parts. Parentheses may be

nested for more complex sequences.

- ★ **Dots** may be inserted anywhere in a header P: field to improve legibility; they have no other function.

- ★ **Some Examples:**

T: La belle Catherine

P: A3B2.A3C2

‡ *Play the A part 3X, the B part 2X, then the A part again 3X, and the C part 2X*

T: Les cinq jumelles

P: (A.B)2.C.D2.E2

‡ *Play A, then B, then repeat A & B and do C, and finally two each of D & E parts*

And suppose we had to play *Les cinq jumelles* seven times through for a square dance. Then the P: field would look like this:

P: ((A.B)2.C.D2.E2)7

- ▶ **Endings.** Parts notation may be used in both the Tune Header and Tune Body to set up very flexible patterns of endings.

- ★ Suppose a tune has four repeats each of the A part and B part. We can set up a variety of possible endings for each part. The Tune Header would be
P: A4B4

- ★ In the Tune Body we would specify the relevant **repeat pattern**. Here are a couple examples for the B part.

- ★ **Different ending with each repeat:**

P:A ‡ *(at the beginning of the A part)*

<A-part notes> | [1 <first ending> :| [2 <second ending> :| [3\
<3rd ending> :| [4 <4th ending> |]

- ★ **Different ending for odd, even repetitions:**

P:A

<A-part notes> | [1,3 <notes> :| [2 <Notes> :| [4 <Same Notes> |]
‡ *Endings 2 & 4 are separate to get the correct
‡ ending symbols for each.*

★ Here are a couple other possibilities for specifying repeats:

[1-3 <notes> :| [4 <notes> |]

For 8 repetitions: [1,3,5-6 <notes> :| [2,4,7 <notes> :| [8 <notes> |]

Modifying Notes: Ties & Slurs, Grace Notes & Triplets

Here are several ways of modifying how we play notes.

Ties & Slurs

- ▶ **Ties:** 2 notes of the *same* pitch may be tied (within/across bar): “-” between notes
 - ★ There must *not* be a space between the first note and the tie.
- ▶ **Slurs:** Connect two or more *different* notes (within/across bar): put () around notes
 - ★ Spaces within parentheses are OK between notes but not next to the parentheses sign.

Grace Notes: Use {} symbols around grace notes; no parentheses are needed for slurs.

Triplets: Use (3 followed by the notes of the triplet. To slur, put additional () around whole triplet. Thus:

Without slurs: (3def or with slurs: ((3def)

Accents, Decorations, Redefinable Symbols & Special Characters

Here are some commonly used accents and decorations. See the **abc 2.2 Standard** section 4.14 for many more.

P	Uppermordent (e.g. for an “a” note, Pa would be played aba)
T	Trill
.	Staccato (period)
u	Upbow
v	Downbow

!0!–!5! Fingerings
 !+! Left-hand Pizzicato
 !slide! Slide up to a note

Redefinable Symbols. The letters H–W and h–w, and the symbol ~ may be assigned as shortcuts for commonly used symbols, and for annotation text. This is done using the U: field.

- ▶ Once defined, the letter shortcut may be used any time the defined symbol or text is to be used.
- ▶ If defined in the **File Header**, the shortcut applies to all tunes in the abc document.
- ▶ If defined in the **Tune Header**, the shortcut applies only in that one tune.
- ▶ Here are a couple examples to illustrate redefinable symbols:

U: S = !slide! % DEF SG>F GB *puts a slide before the G*
 U: p = "^+" % ABcA F2pE2 *puts a + over the E*

Special Characters

The following characters should be understood. Enter what is shown under *Encodings* to get what is shown in *Examples*.

Accent	Examples	Encodings
grave	À à è ò	\`A \`a \`e \`o
acute	Á á é ó	\'A \'a \'e \'o
circumflex	Â â ê ô	\^A \^a \^e \^o
tilde	Ã ã ñ õ	\~A \~a \~n \~o
umlaut	Ä ä é ö	\"A \"a \"e \"o
cedilla	Ç ç	\cC \cc

Accent	Examples	Encodings
ring	Å å	\AA \aa
slash	Ø ø	\O \o
breve	Ă ă Ę ę	\uA \ua \uE \ue
caron	Š š Ž ž	\vS \vs \vZ \vz
double acute	Ő ó Ű ú	\HO \Ho \HU \Hu
ligatures	ß Æ æ œ	\ss \AE \ae \oe

From the abc 2.2 standard

- ▶ In my experience, if the abc reader uses *abcm2ps* to interpret the abc code (e.g. *EasyABC*), the *Encodings* are necessary. If the software uses *abc2svg*, it's generally OK to use the *Examples*.

Chords & Unisons

Two or more notes played or sung together constitute a **chord**. Chords may be written as chord symbols above (or below) the melody line, or as more than one notehead on a single stem. If a chord contains more than one of the same note (e.g. on a fiddle, an open E and an E on the A string), it is a **unison**.

Multiple Noteheads Per Stem

Use `[]` around notes to produce multiple noteheads per stem.

- ▶ If both notes are the same it will be a **unison**.
- ▶ Modifiers to length can come after brackets: e.g. `[AC]2 = [A2C2]`.

Chord Symbols

Use double quotes for chords; e.g. `"Am7"`

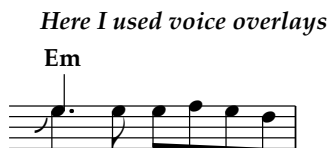
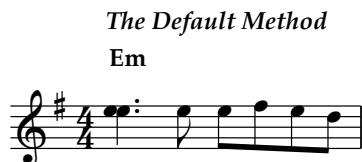
- ▶ Again, be sure to turn off smart quotes.
- ▶ Chords may be accompanied by bass notes, e.g. to indicate a bass line.
 - ★ For example, an A7 chord with an E bass note would be `"A7/E"`
- ▶ Here is the format for more complex chords:
 - ★ `<chord name><accidental><type></bass>` where
 - `<chord name>` = letter name of chord ("A")
 - `<accidental>` = b or #
 - `<type>` = m, dim, 7, etc. (see table of types below)
 - `</bass>` = "/" followed by bass note name
- ▶ **Usable chord types** are as follows:

<code>m, min</code>	minor
<code>maj</code>	major
<code>dim</code>	diminished
<code>aug, +</code>	augmented
<code>sus</code>	sustained
<code>7, 9</code>	7th, 9th or other number

Using Voice Overlays with Unison Drones

Unison drones are written out as stems with two noteheads. In my experience the default in fiddle music is one notehead with two stems, one pointing up and the other down. In abc notation the default seems to be one stem with two noteheads. Displaying unisons with a single notehead may be accomplished with **voice overlays**: temporary use of multiple voices, as shown below.

Here are a couple ways to set up unison drones.



- ▶ The first method uses the following code, which is a fairly standard way of notating unisons:

```
"Em" [ee]2 > e2 efed |
```

- ▶ The second uses **voice overlays**:

```
(& "Em" se2>x2 x4 & e2>e2 efed &) |
```

- ★ The (& [abc code] & [abc code] &) structure allows for two temporary voices in the same measure, separated by the middle "&".
- ★ As there's only one drone note, I used invisible rests ("x") to fill out the first voice.
- ★ The s in this and the last example was redefined as a slide with the code:

```
U: s = !slide!
```

- ▶ The third example also uses voice overlays, but I put all the notes in the first voice:

```
("Em" se2>e2 efed & e2>x2 x4 &) |
```

- ★ From this we can see that the first voice has upward stems and the second voice has downward stems.

Page Layout and the Use of Text in abc Documents and Scores

Text in abc Documents

Text Outside the Staff

An abc file may contain **free text** and **typeset text** outside of the staff.

- ▶ **Free text** is intended to be for annotation and is not generally printed.
 - ★ It can occur anywhere in an abc file but must be separated from surrounding text (e.g. the abc code for a tune) by **blank lines**.
- ▶ **Typeset text** is intended to be printed; the details are governed by **text directives**.
- ▶ **Comments**. Anything that occurs after a % sign will be ignored. It may be either in the middle of a line or at the beginning of a new line.
 - ★ To include a % symbol in the text to be displayed or printed: Type \%.
- ▶ The easiest way to designate text to be printed is to start a new line with %%text followed by a text string; each new line must start with %%text.
- ▶ For a more detailed discussion of typeset text, see the abc 2.2 standard, section 11.4.

Text Inside the Staff

There are two main categories of text inside the staff: **annotations** and **lyrics**. This guide doesn't deal with songs, so we will look at **annotations**.

Annotations: Text may be added within the staff. Annotations are surrounded by double quotations like a chord, but the text string is preceded by a symbol that indicates where it should be placed:

- ▶ "**^** text": prints "text" above the staff, very much like a chord
- ▶ "**_** text": prints "text" below the staff
- ▶ "**<** text" and "**>** text": print "text" to the left or right of a note
 - ★ So "**<**(" **>**)" e would put parentheses around an "e" note.
- ▶ "**@<x>,<y>** text" (where <x> and <y> are numbers): prints "text" offset by <x> and <y> points from the position of the symbol being annotated.
 - ★ Note that the annotation of two different notes could have the same

description but be at different heights relative to the staff.

The use of text annotations will be illustrated in the next section.

Blank Space in abc Documents

For purposes of legibility it is often desirable to leave blank space both in an abc document and in the resulting musical notation. Here are several categories of blank space and how to achieve them, followed by examples.

- ▶ **The abc code document.** There are two easy ways to leave blank space.
 - ★ Space *within* a line helps legibility. Just use the spacebar as seems desirable. There are a few places where that affects the score but usually it can be worked around.
 - ★ Space *between* lines is also helpful. A blank line doesn't work inside a tune but you can insert a blank comment — a line containing only a percent sign.

Here are a couple examples of leaving blank space in an abc document.

Leaving Space: The abc Document

Here are some examples of leaving space in a document to improve readability. (This is the ending of the B part of Marcel Messervier's Reel des Accordéonistes.)

First, this is what it looks like with minimal use of space (code first, then resulting score).

```
"D/A"A2FA "Bm"d2cd|"Em"efec "A7"dcBA|1  
"D"FA"A7"Bc "D7"d2:|2\  
"D"FA"A7"Bc "D7"d2de|]
```

The image shows two staves of musical notation. The first staff is in G major (one sharp) and 2/4 time. It contains a melody with four measures. Above the notes are the chords D/A, Bm, Em, and A7. The second staff is also in G major and 2/4 time. It contains a melody with two measures. Above the notes are the chords D, A7, and D7. The first measure is marked with a '1' and the second with a '2', indicating first and second endings. The notation is clean and well-spaced.

That code is pretty hard to read. Here I've added some space in the abc file to improve readability. A couple other changes were made for space reasons but they don't affect the score.

```
"D/A" A2FA "Bm" d2cd      | \  
"Em"  efec "A7"  dcBA      | 1  
"D"   FA  "A7"  Bc  "D7"  d2   : | 2 \  
"D"   FA  "A7"  "Bc" "D7"  d2de  | ]
```

Normally there would be more than one measure on a line but there's not room here.

[The score is unchanged so it's omitted here]

It can improve readability to add space to line up bar lines.

Blank Space in the Resulting Score

Again we need both horizontal and vertical space; and we can insert breaks in a staff as well.

- ▶ **Horizontal space** may be created by typing a "y", or more than one if desired. A "y" followed by a number indicates how much space; e.g. "y40" would mean to add 40 pixels of space.
- ▶ **Vertical space** is added with the *vskip* command. Start a new line and enter `%%vskip 20` to get twenty pixels of vertical space.
- ▶ Staff breaks (horizontal spaces between parts of a staff) are created with the *staffbreak* command. On a new line, you may enter `%%staffbreak 2cm` to get a break two centimeters wide.

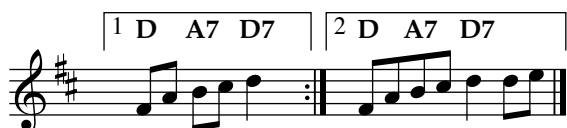
Here is an example illustrating the use of blank space in an abc score.

Leaving Space: The Printed Score

There's still a problem. In the second line the chords are really poorly spaced. Lets look at improving spacing in the resulting notation. (Note that as this is two-column text we must keep the lines shorter so they will fit in the allotted space; otherwise the text from the two columns will overlap.)

Let's start with adding in some *horizontal space* to get the chords a bit better spaced. This is accomplished with a y. We can use two or more y characters or use a number to indicate how much space.

```
"D/A" A2FA "Bm" d2cd | \  
"Em" efec "A7" dcBA | 1  
"D" FA "A7" Bc "D7" d2 y :|2\  
"D" FA"A7"Bc "D7" d2de | ]
```



We may want some *vertical space* between elements of the score. In the abc code I used the *vskip* command throughout for that purpose. For example, there's a `%%vskip 15` between the second paragraph and the abc code, and `%vskip 20` between the abc code and the music.

Finally, we might want a *break* in the staff, for example when illustrating two different variations. This is done with the *staffbreak* command.

```
| g4 fgaf ||\  
%%staffbreak .5cm  
| g2fg dgfg ||
```



Printing & The Use of Fonts in abc Notation

Although many people are unaware of it, and isn't always easy to find documentation, nearly all aspects of an abc document may be customized, often with considerable flexibility. Fonts fall into that category. There are some weaknesses, limitations and areas of potential confusion.

Some Issues Related to Printing

- ▶ If you try to print directly from an abc app (e.g. *EasyABC*), the result will generally be of low quality because it is likely to be based on the screen image.
- ▶ Quality printouts of music based on abc notation usually make use of either **abcm2ps** or **abc2svg** to transform the code into a high quality printout. There are some important differences between them.
 - ★ *Underlying mechanism:* abcm2ps uses Postscript and abc2svg uses svg images for the printout.
 - ★ Therefore abcm2ps uses the PDF format and makes use of Postscript fonts, whereas abc2svg (which is a Javascript application) can handle more modern web font formats.
 - ★ Currently abcm2ps may be slightly better at handling situations where precise layout is concerned, whereas abc2svg is better at handling modern fonts and is being actively developed.
- ▶ Below are cropped printouts to illustrate the differences that may occur.
 - ★ Printed from: *Left*—EasyABC; *Center*—abcm2ps; *Right*—abc2svg

Notating Pitch

G A B c d



Printed from EasyABC

Notating Pitch

G A B c d



Printed from abcm2ps

Notating Pitch

G A B c d



Printed from abc2svg

- ★ From this we can see that the EasyABC printout is substantially lower in quality than the others (although better than printouts directly from the abc software are sometimes); and there are font differences between the abcm2ps and abc2svg outputs.

Use of Fonts With abc Notation

With abc notation, Times, Helvetica and Courier and their stylistic variants are the default available fonts. Other fonts should be declared at the beginning of an abc document. For example:

```
%%font          Palatino-Roman
%%font          Palatino-Bold
%%font          Palatino-Italic
%%font          Palatino-BoldItalic
```

Different categories of text may then be declared, with corresponding fonts and sizes. These declarations may be modified within the tune as needed. For example:

```
%%titlefont     Palatino-Bold 20      % Font of title
%%subtitlefont  Palatino-Italic 16    % Font of second title
%%composerfont  Palatino-Italic 14    % As it says; also O: (on same
% line)
%%tempofont     Palatino-Roman 14     % As it says
%%gchordfont    Palatino-Bold 14     % Chord font
%%partsfont     Palatino-Bold 14     % Parts font (sequence above score
% label parts of tune (A, B, C ...))
%%historyfont   Palatino-Roman 14    % History (H:) + Source (S:), Notes
% (N:) & Transcription (Z:)
%%annotationfont Palatino-Roman 14   % For annotations, discussed above
%%textfont      Palatino-Roman 14    % For lines of text
%%infofont      Palatino-Bold 16     % Infoline font (combined R: & A:)
%%setfont-1     Times-Bold 16        % allows changing font within a
%%setfont-2     Times-Italic 16      % string. $1 through $4 to change,
% $0 to change back to the default
```

- ▶ Some of this I haven't seen documented anywhere (although likely it is and I just haven't seen it). In particular, I've not seen anything about formatting the Notes field, and had no idea it was done with the `historyfont` directive. But experimentation made it quite clear that's how it works.
- ▶ Fonts are handled much more flexibly by abc2svg in terms of which fonts may be used and how they have to be defined.

An Example To Illustrate Many of These Concepts Together

This example documents a couple details of New Hampshire fiddler Marcel Robidas' playing of the tune "You Married My Daughter But Yet You Didn't". It will be followed by the abc code that produced it.

Use of Annotations, Text, Spacing and More

Example from playing of New Hampshire fiddler Marcel Robidas

Here are a couple common melodic variations.

*Often in the A part (measures 1, 3, 5)
a half note is used for emphasis.*

*Marcel Robidas would pluck the
E string in the B part, measure 7.*



Here is the abc code that produced it. The font definitions on the previous page are all included at the beginning of the document; I won't repeat them here. The comments on the right explain most of what's going on.

```
X:10
T:Use of Annotations, Text, Spacing and More
T:Example from playing of New Hampshire fiddler Marcel Robidas
M:C|
L:1/8
K:D
%                               % Blank comments for readability
%%vskip 20                       % Adding in space
%%text Here are a couple common melodic variations.  % Text within the
%%vskip 75                          % document.
%
"@0,60 Often in the A part (measures 1, 3, 5)"\  % The next few lines
"@0,40 a half note is used for emphasis."      % format the annotations.
"@340,60 Marcel Robidas would pluck the"\
"@340,40 E string in the B part, measure 7."
%
|^A1 & A5" g4 fgaf ||\
%%staffbreak .5cm                               % Breaks in the staff
|^A3" f4 efge ||\
%%staffbreak 1.5cm
|^B7,8" F2!+!e2 dcBA | G4 :|\
```

The text annotations may need some explanation. The @0,60 tells the software to put the text that follows 0 points to the right of the note that follows it, and 60 points above. As there are four lines of annotations before the tune starts, all are with respect to the same note.

Text to the Left of an Indented Staff

Suppose a tune has an Introduction that isn't repeated again. One way to set it off is to indent the staff and put the word "Introduction" to its left. Here's an example from the square dance tune *Don't Dilly Dally*.

Don't Dilly Dally
Text to the Left of an Indented Staff

This is fairly easy to do using the annotations described above and the Indent parameter. Here is the code responsible for it with explanatory comments.

```
X:11
T:Don't Dilly Dally
T:Text to the Left of an Indented Staff
M:2/4
L:1/8
K:F
%%annotationfont Palatino-Roman 16           % Defines the annotation font
%%indent 150                                  % Indents the first staff by
                                                % specified amount
"@-170,0 Introduction" \                     % Moves annotation to the left by
                                                % specified amount.
"F" F2F>G | "F" AG F2 | "Bb" G2 GA | "Bb" BGAB |\
"C7" c2 d2 | "C7" A2 G2 | "F" F4- | "F" F3 ||
|: A |\
"F" c2 d2 | "F" A2 GF | "G7" G A2 G | "G7" D4 |\
"C7" E2 c>c | "C7" cBG^G | "F" A4 - | "F" A4 |
```

The indent command moves the first staff to the right, and the @ parameter moves

the annotation text to the left of the staff.

EasyABC Shortcuts

EasyABC is one of the most widely used abc applications.

- ▶ EasyABC was written by Nils Liberg. Several years ago he had to stop developing it, and it was taken over as open source software.
- ▶ Documentation for EasyABC and most of the shortcuts below and others used to be listed at <https://www.nilsliberg.se/ksp/easyabc/>. It's not there now (March 2022); I don't know if that's permanent.
- ▶ Documentation for the more recent updates, including more useful shortcuts and techniques is found at <http://easyabc.sourceforge.net>.
- ▶ To download the current version of EasyABC, go to <https://sourceforge.net/projects/easyabc/?source=directory>. The current version for Macintosh is 64 bit and works on the latest version of the operating system.

Here are some shortcuts and tricks I've found (or figured out). Some may be usable in other software. Some don't work unless explicitly enabled (and some may make things more difficult for some people or in some circumstances).

Musical Notation Shortcuts

Bar Lines

- ▶ Type a *<Tab>* to insert a bar line.
- ▶ If enabled, when a measure is full typing a *<Space>* will insert a bar line. To enable: *Settings ABC Typing Assistance → Add Bar*

Automatic Upper/Lower Case:

- ▶ Choose *Settings → ABC Typing Assistance → Automatic uppercase/lowercase* to enable.
- ▶ If the case is incorrect, delete the letter, hold the *<Shift>* key, and the case will be reversed.

Add Matching Right Symbol:

- ▶ Choose *Settings* → *ABC Typing Assistance* → *Add Matching Right Symbol* to enable.
- ▶ Typing a “(”, “[”, “{”, double quotation mark results in the matching right symbol is entered automatically.

Some Additional Shortcuts

Slurs: Select two or more notes and type “(“ or “s” to add a slur around the selected notes (or remove it if already present).

Dotted Rhythm: Select multiples of two notes and type “<“ or “>”. For example, selecting “AB CD” and typing “>” produces “A>B C>D”.

Playback Shortcuts

To play the Entire Tune: Click on the *Play* button (, upper left).

Loop playback: Double-click on the *Play* button. (This actually works on the 64-bit Mac version.)

Skip repeats: *Control* + *Play* button (but this doesn't).

To play from the middle: Use the *Play position* slider.

Playable Tunes: Combining abc Notation, CSS & HTML Code

One of the most important developments with abc notation is the ability to combine it with HTML and CSS code to produce tunes that are displayed on a web page in standard notation and which may be played by clicking on them. There are several ways to accomplish approximately this goal, but the one that I find most approachable uses Jean-François Moine's **abc2svg**.

You can use abc2svg by going to Jean-François' ABC-edit page at <http://moinejf.free.fr/js/edit-1.xhtml>.

An Introduction to Playable Tunes on the Internet

First of all, the term *playable tune* is my description; I'm not sure what the correct term is or if one exists.

The Big Picture

There are two components needed: (1) one or more tunes written out in abc notation, and (2) the capability to edit HTML and CSS code. Once created, the result takes the form of a web page that can be opened in any modern browser.

Starting with code for a basic web page, in the `body` section of the page we can create a section in which the code is treated as abc notation rather than as HTML code.

The abc code is placed in one or more such sections, including the *file header* code (if any) and the *tune header* code.

When saved as an HTML file and opened in a web browser, the tunes appear in standard notation, and clicking on a note starts playback from wherever you click.

Setting Up The HTML File

There are a couple things that need to be added to an HTML file beyond the usual to make this work.

- ▶ A very basic HTML page would start like this; likely there would be a bit more included:

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
```

- ▶ In order for the web browser to be able to interpret the abc code a couple lines need to be added to the `<head>` section. These refer to JavaScript code written by Jean-François Moine. The first deals with interpreting the abc code, the second deals with making the code playable:

```
<script src="http://moinejf.free.fr/js/abcweb-1.js"></script>
<script src="http://moinejf.free.fr/js/snd-1.js"></script>
```

- ★ If you're working with a **secure website** (`https://`), your server may not load a script from a nonsecure site. In that case you'll have to download the scripts and put them on your server.
- ★ From the URLs in the scripts, the files are named **abcweb-1** and **snd-1**; the part before that is the address of the page where they can be found.

- ▶ There's one more line that has to go in the `<style>` section:

```
<style>
  svg { display: block }
</style>
</head>
```

- ★ That allows for music that's more than one line long. You can of course put other code in the `<style>` section; we'll get to that later.

The abc Code

The abc code goes in the `<body>` section. It has to be enclosed by code that tells the browser to use the `abcweb-1` and `snd-1` scripts to interpret it as abc code. I've also added a line that alerts you if JavaScript isn't running (e.g. on an iPhone).

- ▶ The following accomplishes that purpose:

```
<body>
<noscript>Please activate Javascript!</noscript>
<script type="text/vnd.abc">
  all abc code goes here
</script>
</body>
</html>
```

- ▶ All abc code goes in the `text/vnd.abc` section. That includes any formatting parameters as well as the abc code for the tune.
- ▶ There can be abc script sections, with regular HTML code in between.
 - ★ If you have information about the tunes it can go in an HTML section between tunes; that makes it much easier to format the text. For example, line breaks can be up to the browser rather than fixed and likely not optimal.

Making abc and CSS Work Together

One of my favorite aspects of looking at abc notation in this fashion is the ability to use CSS to style aspects of the tune in ways that would be difficult through abc code by itself.

- ▶ The basic idea is to define styles (e.g. for text) using CSS and then to call up those styles inside the abc portion of the web page.
 - ★ These styles also go in the `<style>` section of the page.

An Example

Here is an example of an HTML document that would produce playable tunes with formatting in part from the CSS in the <style> section. I'll use CSS and abc comments to explain some of what I've done.

- ▶ I'm including this code as a text file and as an HTML file. They differ only in their file extension. If you open the text file you can see the code more easily. If you open the HTML file it should open in a web browser with well formatted text and a tune.
 - ★ The fonts I'm using are likely to be on most Macs and PCs; feel free to change them if not.

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <script src="http://moinejf.free.fr/js/abcweb-1.js"></script>
  <script src="http://moinejf.free.fr/js/snd-1.js"></script>

<style>

/* Begin Type Styles */

  .title {
    font-family:"Palatino Linotype", "Book Antiqua", Palatino, serif
!important;
    font-size:24px !important;
    font-style:bold !important;
    margin-right: calc(100% - 816px);
  }
  .subtitle {
    font-family:"Palatino Linotype", "Book Antiqua", Palatino, serif
!important;
    font-size:18px !important;
    font-style:italic !important;
    line-height: 1.2 !important;
    margin-right: calc(100% - 816px);
  }
  .text-lg {
    font-family:"Palatino Linotype", "Book Antiqua", Palatino, serif
!important;
    font-style:normal !important;
    font-size:16px !important;
    margin-right: calc(100% - 816px);
  }
  .text-sm {
    font-family:"Palatino Linotype", "Book Antiqua", Palatino, serif
!important;
    font-style:normal !important;
    font-size:14px !important;
    margin-right: calc(100% - 816px);
  }
~ 24 ~
```

```

    }

/* All those "!important;" endings apparently matter; removing them caused
the formatting to become generic.. */

    svg {
        display:block;
    }

/* ----- Box for Body ----- */

    div.body-box {
        display:block;
        height: auto;
        max-width: 816px;
        width: 95vw;
        padding: 10px;
        border: none;
    }

</style>
</head>
<body>
<noscript>Please activate Javascript!</noscript>

<br>

<div class="body-box">

<p class="text-sm">The "body-box" is a box to keep the text from getting wider than
the tune. Without it the text would go out to the edge of the window. Next comes
the first abc section which contains the formatting parameters buut nothing that
prints.</p>

<script type="text/vnd.abc">

%abc-2.2
I:abc-charset utf-8

%%pagewidth      8.5in
%%leftmargin     0.5in
%%rightmargin    0.5in
%%titlefont * * class=title           % This lets an abc formatting command
%%subtitlefont * * class=subtitle     % use a style defined by CSS
%%composerfont * * class=subtitle
%%tempofont * * class=text-sm
%%gchordfont * * class=text-lg
%%annotationfont * * class=text-lg    % For annotations within the staff
%%textfont * * class=text-sm          % For lines of text
%%infofont * * class=text-lg         % For info fields (e.g. T:, C:, I:)
    %%topspace      0.0cm
%%titlespace      1.0cm
%%composerspace   0.2cm
%%musicspace      0.2cm

%%stretchlast    0.5

```

```

%%setfont-1      Bold                % Allows changing font within a
%%setfont-2      Italic              % string. $1, $2 to change, $0 to
%%setfont-4      Courier 16

%%lineskipfac    1.4
%%staffsep       50
%%maxstaffsep    50
%%subtitlespace  0.3cm

%%MIDI program   74                  % GM Flute for the melody
%%MIDI chordprog 1                  % GM Acoustic Grand Piano for
    accompaniment
%%MIDI gchordon  % turns on accompaniment

```

```
</script>
```

```
<p class="text-sm">Here's some non-abc text in between the two abc sections. This was
    part of an effort to develop a better set of chords for this tune than we had
    available previously. </p>
```

```
<script type="text/vnd.abc">
```

```

X:30
T:Gone A Rovin'
C:Ralph Page
M:C|
L:1/4
R:March
K:G
%%text $2Chords by Peter Yarensky based on Randy Miller's revised chords.$0
GA |\
"G" B>c BA | "G" AG Bd | "G" g>a gf | "C" e2 ef |\
"C" g>a ge | "G" dB Bc | "A7" d>e dB | "D7" A2 GA |
"G" B>c BA | "G" AG Bd | "G" g>a gf | "C" e2 ef |\
"C" g>a ge | "G" dB Bd | "D7" c/d/c BA | "G" G2 ||
fg |\
"D" a>b ag | "D" fd ga | "G" b>c' ba | "G" g2 gf |\
"C" e>d ef | "C" ag fe | "G" d>e dB | "D7" A2 GA |
"G" B>c BA | "G" AG Bd | "G" g>a gf | "C" e2 ef |\
"C" g>a ge | "G" dB "Em" Bd | "Am" cd/c/ "D7" BA | "G" G2 ]

```

```
</script>
```

```
<br>
```

```
<p class="text-sm">And that's the end. We've used CSS styles, abc code, and mixed abc
    and HTML together. You could easily change settings or paste in a different tune
    if you wanted to experiment with this. Just pay attention to whether there are
    any extra line breaks due to copying from a PDF.</p>
```

```

</div>
</body>
</html>

```

Some Additional Useful Links

Here are a few useful links for abc notation.

The abc Notation Home Page: <http://abcnotation.com>

Software recommendations from Chris Walshaw: <http://abcnotation.com/blog/2017/11/12/6-of-the-best-a-guide-to-abc-software/>

abc Software Downloads: <http://abcnotation.com/software>

The abc Standard, versions 2.1 (current) and 2.2 (in progress):

<http://abcnotation.com/wiki/abc:standard:v2.1> and

<http://abcnotation.com/wiki/abc:standard:v2.2>

This help file: <https://fiddle-tunes.nhcountrydance.com/resources/Other-Resources/Text-Based/abc-2.2-help-file.pdf>

abc Users Group List: <https://groups.io/g/abcusers>