Bad Tunes
(and how to fix them)

The abc language was designed to be written by people and read by both people and computers. Unfortunately, people make mistakes, and you will find many examples on the net of tunes which do not play or display properly, because they contain errors. Locating and fixing these errors so that the tunes become usable is a daunting task for beginners (and sometimes even for experts too). This file contains example tunes with errors to show you how to identify and fix the common errors. There are actually no bad tunes - just bad transcriptions.

Display Errors.

Errors picked up by the display parser generally result in in an error message drawn in the music pane of the split-screen display. If the error is in the abc header you will see no music, just an error message, and hopefully this will tell you exactly which field of the header is wrong. If the error is in the tune itself you will usually get part of the tune displayed, stopping on the last note before the error. If you click on the head of the last note displayed, the equivalent symbol in the abc text will be selected, and you should look at the text immediately following this to locate the error.

Playing Errors.

The play parser generally tries to work around any errors it finds, and most tunes will play something, even if it's not quite what the author intended. Playing a tune with errors (or silently error checking it by clicking on the smiley face icon at the top right of the window) will result in the smile changing to a frown. If it's frowning, click on it again and a new window will open containing a list of errors. Usually these will be sufficiently explicit to identify where the errors are, and what you have to do to fix them.

In Extremis.

If neither of these approaches works, you can fall back on the old programmer's technique of commenting out suspect parts of the tune to locate the error. Start by selecting everything from the start of the second line of the tune (i.e. the second line after the K: field) to the end of the tune, and choose Comment out Selection from the Edit menu. You have now removed from consideration everything but the first line of the tune - if this line works OK, you know that the error is somewhere in the commented-out section, so you can work through it, removing the % sign from the start of each line until you locate the line with the error. If it's not obvious where in that line the error is, you can start to break the line up by placing the insertion point after a bar line in the middle of the suspect line of music, and typing a couple of returns to take the second half of the line out of the tune altogether. Continue using this process of elimination until you locate the error.

The Tunes.

Most of these are real examples picked up on the net. I have removed the attributions, if any, to avoid embarrassing the authors, and in some cases edited them to remove errors other than the error that I'm demonstrating. In each case I have inserted a description of the symptoms in a red T: field before the real tune title, so these are what will show up in the index when you are in split-screen mode. You can use this to search quickly for an example which matches the tune you are trying to fix. Otherwise, look through the tunes one by one and see if you can figure out the error before reading the answer in the text below.

```
X:1
T:No Tune!
T:The Great Eastern
R:reel
M:C|
L:1/8
```

```
K:D

D2FD CEAG | Fddc dAFA | GBeG FAdA | GFED CA,CE |
D2FD CEAG | Fddc dAFA | GBed cAGE |1 EDCE DABC:|2 EDCE D2fg ||
|: a^gaf =g~e3 | dfec dAFA | GBeG FAdf | gfe^d efgb |
a~f3 g~e3 | dfec dAFA | GBed cAGE |1 EDCE D2fg:|2 EDCE DABC ||
```

The abc standard says that the tune starts on the next line after the K: field and ends on the first blank line. This tune has a blank line immediately after the K: field, so there's no tune at all. Remove the blank line to fix it.

```
T:I don't exist at all!
T:Jimmy Batty's
C:John Carty
R:reel
M:C|
L:1/8
K:D
dAAB AGFE|DFAB ADFA|B~E3 B/2c/2d ef|gaaf edfe|
dAAB AGFE|DFAB ADFA|BEEA B/2c/2d eg|fdec defe:|
|:dAAB AGFG|Aage f/2a/2f ed|Beed edef|g2fg edBc|
dAAB AGFG|Aage f/2a/2f ed|dfed BAFB|AFAB defe:|
```

This is a common fault in many tunes which are posted to mailing lists and usenet groups. BarFly simply does not recognise it as a tune - it won't play or display as music. If it is the only tune in the file BarFly won't switch to split-screen mode, and it won't be counted in the number of tunes shown at the top of the window.
The reason is that it lacks the X: field at the start. The abc standard says that this is a mandatory field, and BarFly enforces that strictly. In fact BarFly uses the X: field to distinguish tunes from any text which precedes them. To fix it, just type X:2 on the line above the first T: field.

```
 X:3
 T:I don't exist either
 T:Bonny Gateshead Lass
 M:4/4
 L:1/4
 K:F
 E2|F2F2|GEFG|ABcc|BGF2|E2C2|
 E2F2|F2GE|FGAB|c2c2|d2B2|
 c2A2|B2cc|dBcA|G2F2|B2G2|
 c2F2|F2E2|D2C2|F2F2|G2E2|
 F2G2|A2B2|c2c2|B2A2|F4||
```

The reason for this is fairly obvious when you see the tune next to other text - every line starts with a space character. It's less obvious when this tune is all there is in the file. Fields in an abc tune must be written with the field identifier hard up against the left-hand margin, or they won't work. This error is something that Netscape does when displaying abcs from some sites, and if you use Copy and Paste to transfer the tune to BarFly this is what you'll get. To fix it, edit out the leading spaces from the header part of the abc (leading spaces don't affect the tune proper unless it contains in-tune fields). If you have a whole file of tunes like this, hand editing is really hard work, but you can use BarFly's Replace Text command to do it all in one go:

First, edit out the space on the first line by hand.
With the insertion point in the first line, choose Replace Text from the Edit menu.
Enter ¥13¥32 in the Search For box. (Use Option-8 to get the bullet mark ¥)
Enter ¥13 in the Replace With box.
Check the Replace All checkbox.

Hit OK.

```
X:4
T:The line break demon
T:Raemona's Waltz
M:3/4
R:Waltz
C:Brian Hebert
N:C Ricker's Old Teacher.
K:G
dc (3BcB G>E||D2 B3 D|_E2 B3 _E|=E2B3d|c2 B2A2|G2E3G|F2GA3|  D6  |
D3B,C^C|D2B3D|_E2B3_E|=E2B3d|c2B2A2|G2E3G|F2GA3|  G6  |G3D (3EFE|D2B3D|
_E2B3_E|=E2B3d|c2B2A2|G2E3G|F2GA3|D6|D3B,C^C|D2B3d|_e2B3E|=E2B3d|ed Bd BA|G
2E3G|F2GA3|G6|G3def||
|:g6|f2dB3|c3d (3efg|d6|e3f (3gfe|de dc B2|A2 ^C2 (3efe|d3 d ef|g6|
f2dB3|c3 d (3efg|d4 dB|c3c2c|c2B2F2|1  G6  |G3 d ef:|2  g6  |G3 B, C^C||
```

In this case, there was nothing wrong with the original transcription - the tune has been sent by email, or transferred by some other program which has introduced a spurious line break into the longest line of the tune. Sometimes this causes no problems, but when it separates symbols which the abc standard says must be written adjacent to one another it can generate all sorts of errors. In this case the note G at the end of the third line has been separated from its length (2) at the start of the fourth line.
This causes the player to report a bar check error, and the display to report an illegal character "2" in the tune. This is a very common problem, and can cause various symptoms, so it's sometimes difficult to spot. Look for long lines which end in the middle of a bar, followed by a short line.

To fix the problem, place the insertion point before the 2 at the start of the fourth line of tune and type a backspace to re-join the long line. If you are making abc transcriptions of your own, you can avoid problems of this kind by keeping the lines short, splitting long lines up by means of the continuation character "\". If that problem line had been written like this the tune would have survived undamaged:

```
_E2B3_E|=E2B3d|c2B2A2|G2E3G|F2GA3|D6|D3B,C^C|D2B3d|\
_e2B3E|=E2B3d|ed Bd BA|G2E3G|F2GA3|G6|G3def||
```

```
X:5
T:Illegal colon in tune
T:Bonnie Kate
M:4/4
L:1/8
Q:140
S:Harrison & Wall MS,Ashover,Derbyshire,1762-75
R:Cotillion
O:England
A:Derbyshire
K:A
a>bag ae-ef | edcB cA-AB | (cA A)B/c/ (dB Bc/)d/ | c>edc B4:|!
:(cA A)B/c/2 dBec | (fd d)e/f/ g2 ef/g/ | a>baf egaf | e>fec A2A2:|!
```

Although this tune will play OK it will not display completely as music because of the use of a single colon as a repeat marker at the start of the second line. This is a very common error in abcs generated by the program abc2win. This is a popular abc program for the PC and has many users. Unfortunately it has very little error checking, and accepts almost any old garbage as valid abc, resulting in a number of characteristic errors, of which this is one. Note also the exclamation marks at the ends of the lines. abc2win uses these to force the line to be drawn right-justified. The exclamation mark is not part of the abc standard, and strictly speaking should be marked as an illegal character;

however, there are so many tunes on the net like this that it's not worth the irritation, so BarFly simply ignores it.

To fix the leading colon problem you can either replace it with the correct abc symbol for a forward repeat (|:), or simply delete it - BarFly treats forward repeats after a backward repeat as optional, and the tune will repeat from the same point even in the absence of a forward repeat marker.

```
X:6
T:Illegal bar constructions
T:Britons Strike Home
M:3/4
L:1/4
S:J.Winder MS,Wyresdale,Lancs.,1789
R:Air
O:England
A:Lancashire
K:D
dde|f2a|def|efg|agf|e3:|:e3|eef|!
e2z|f2z|ffg|fga|g2f|fe3/2 (d/4e/4)|f3|a2z|!
a (b/2a/2) (g/2f/2)|g2z|a2z|efg|fga|bag|fga|(g/2f/2) e2|dd2:|]
```

This tune contains a couple more abc2winisms. The only legal bar line/repeat symbols in the abc definition are: |, ||, :|, |:, ::, [|, |].

These are illegal: :|:, :||, :], :|], ||:, [:, [|: (and probably a few more I haven't thought of)

This is legal: :||: (since it consists of the legal symbols :| and |: drawn back to back), however, it's a waste of space since the abc definition contains a symbol for repeat-both-ways (::), and BarFly will mark it as an error.

Illegal barline symbols like this don't cause too many problems, and generally BarFly will do the right thing when asked to play or display them. They will wipe the smile off the error icon when you play the tune though. One exception is the use of :|: for repeat-both-ways. BarFly will mark the second colon as illegal, and not display the tune from that point on.

To fix this tune you should change :|: in the first line to ::, and change :|] at the end to :|.

```
X:7
T:Tie used as slur (1)
T:The 29th of May
T:King Enjoys His Own Again
M:4/4
L:1/4
S:J,Winder MS,Wyresdale,Lancs.,1789
R:March
O:England
A:Lancashire
K:D
AAdd|e/f/g/a/f e/-f/|gBc d/-e/|fe{d}d2:|
e/d/c/d/e>f|e/d/c/d/e>f|gBcd|e/d/c/B/ A f/g/|
a/b/a/g/ f e/d/|e/f/g/a/f e/f/|gBc d/-e/|f e{d} d2:|
```

The abc language uses separate symbols for ties and slurs - conventional music doesn't, and this confuses some users. The abc symbol for a tie is the hyphen '-', and it should only be used between notes of the same pitch, causing them to be played as a single long note. So A-A is legal, while A-B is not. Slurs are represented using parentheses (), so to slur A to B you should write (AB). Slurs may mean that the slurred notes are to be played legato (there is an option to set the length of slurred notes separately in the Player Preferences) or may simply mark phrasing. In abc (unlike conventional notation) you can slur identical notes (AA), however, this produces a misleading musical display, as a musician reading it will assume that a tie is intended.

This error produces few problems (BarFly simply works around it and does the right thing), but it is an error, and you should fix it in the tune above by changing e/-f/ to (e/f/) etc.

```
X:8
T:Tie used as slur (2)
T:Tommy Gunn's
R:Reel
S:From Sean Smyth: The Blue Fiddle
M:4/4
L:1/8
K:G
DF | G2BG AGBG | G2BG GEDF | GFGB A^GAB | BAGE EDDF |
G2BG AGBG | G2BG GEDF | GFGB A^GAB | BAGE EDD^c ||
~d3 B-~B2 AB | d^cdB- ~B2AB | ~d3 B-~B2cB | BAGE EDD^c :|
dgbg agbg | dgbg ageg | gfge d^cdB |1 BAGE EDD^c :|2 BAGE EDDg ||
edBg edBg | edBA G~E3 | BEGB edBA | GFGE EDDg |
edBg edBg | edBA G~E3 | dA (3B^cd edg2 | GFGE EDDF :|
```

This is a more subtle problem, and in fact there is no error in the original transcription; however, sometimes BarFly will report a "Tie used as slur" error here, even though the tied notes are at the same pitch. This is a side effect of the macro used to play the rolls. If you use the Clare fiddle roll macro when playing this tune, the ninth bar, written as ~d3 B-~B2 AB | actually gets played as de/4d/c/4d B-c/4B/A/4B AB | (after the macros have been expanded). Now the tie is between notes of unequal pitch, and the program reports an error. You can spot this by looking at the error report, as the line of abc shown there is the expanded version, and is not the same as the original line in the tune.

You can avoid this problem by turning macros off in the Player Preferences dialog before error checking (unless you specifically want to check macros themselves for errors).

```
X:9
T:Bar check error
T:Fitzmaurice's Polka
C:John Kimmel
S:Buttons & Bows
M:2/4
L:1/16
R:Polka
K:G
D|:G2G2 FGAF|Gg efge|dedB GABG|ABAF DE (3FED|
G2GG FGAF|Gg efge|dedB GABG|1 AFDF GDDD:|2 AFDF G3G|]
K:D
A2FG ABcd|D2c2 c2 z2|GGGG BBBB|F2FF F3A|
A2FG ABcd|B2GA ABcd|B2GA BGBd|gfge cABc|dBAF D3A:|
f4 fedB|A2A2 A3A|B2B2 BcdB|A2d2 f2d2|c6 Bc|d6 c2|B4- BcdB|
e2c2 B2A2|f2f2 fedB|A2A2 A3A|B2B2 Bcde|f4 f2f2|
g2g2 g2fe|a2a2 a2fd|B2 z2 c2 z2|1 d2A2 B2A2:|2 d2DD E2D2|]
K:G
G2G2 FGAF|Gg efge|dedB GABG|ABAF DE (3FED|
G2GG FGAF|Gg efge|dedB GABG|AFDF G3 z|]
```

BarFly's player reports a bar check error when it encounters a bar which doesn't add up correctly. In this case the transcriber has made a simple error in bar 2, and then used copy and paste to copy over the bad bar to the repeated sections, resulting in four separate but identical errors. In each case the program will tell you that there is 1/8 note missing from the bar. The tune is transcribed in 2/4 time, meaning two quarter notes per bar. The default note length is 1/16, so there should be eight default notes in each bar. The bar in question reads |Gg efge| and it's clear that there are only six default notes' worth of time there, and the missing two add up to 1/8, just as it says in the error report. I don't know this tune, but I would guess that the pattern of notes in this bar should follow that in the previous bar, so if you change the four bad bars to read |G2g2 efge| the error will disappear.

```
X:10
T:Bar check error on repeat (1)
T:The Swallow's Tail
R:reel
M:C|
K:Ador
gf|eA (3cBA eA (3cBA|edef gedB|G2 (3BAG dGBG|(3Bcd ef g2fg|
eA (3cBA eA (3cBA|edef g2fg|afge dBGB|1 AcBG A2gf:|2 AcBG A2Bd||
|:eaag abag|edef gedB|G2BG dGBG|(3Bcd ef g2fg|
eaag abag|edef g2fg|afge dBGB|1 AcBG A2Bd:|2 AcBG A2||
```

"Bar check error on repeat" means that the error was detected on the second time through the tune. The bar which is indicated is the anacrusis (the part-bar at the start); this is not checked on the first time through, since it can be of any length. On the second time through its length is added to that of the previous bar, which is already complete so the total is 1/4 note too long. The solution here is quite simple; we don't want the repeat to include the anacrusis, so place a colon after the first bar line to make the repeat start from the first complete bar. This kind of error is common both in abc and in printed music. In printed music it doesn't matter because a musician playing with the beat in his toe will automatically skip the anacrusis on the second time through without even noticing that anything is wrong. Computers, alas, are more literal and play exactly what's written, even when it's wrong.

```
X:11
T:Bar check error on repeat (2)
T:Connaughtman's Rambles, The
M:6/8
R:Jig
K:D
AG|FAA dAA|BAB dAG|FAA dfe|dBB BAG|FAA dAA|BAB def|gfe dfe|dBB B2:|
e|fbb faa|fef d2e|fbb faa|fed ede|fbb afe|dBc def|gfe dfe|dBB B2:|
```

Despite being a simpler tune, the situation here is a little more complex than in the previous example. As before, the error is caused by a mismatch between the length of the anacrusis and that of the last bar in the first part; however, this tune has anacruses on both parts, and just to make life really difficult, the two anacruses are of different lengths. In order to match the first anacrusis the last bar of the first part has to be four eighths long, but to match the second anacrusis on the second time through it has to be five eighths long (as it is at present). The only solution here is to use two alternate endings. To fix this, change the last bar of the first part to this:

```
 |1 dBB B:|2 dBB B2 ||
```

Now it matches in both directions, and BarFly reports no errors. If you are really fussy though, you may notice that the last bar of the second part doesn't match the anacrusis of the first part, so there's an extra beat when the whole tune is repeated. (Turn the Loop option on and leave it playing and you'll see what I mean.) The solution is just the same, except that the two alternate repeat bars are reversed in order.

```
X:12
T:Bar check error on every bar
T:J B Milne
```

```
C:Angus Fitchett
S:Rosa Michaelson, Edinburgh
O:Scotland
M:2/4
K:A
c/d/|e/f/e/d/ cA|EA c2|c/e/d/c/ BA|d3 B/c/|d/e/d/c/ BG|
EG B2|B/d/c/B/ AB|c3 c/d/|e/f/e/d/ cA|EA c2|c/e/d/c/ BA|
f3 f/g/|a/f/^d/=c/ A a/g/|f/e/c/A/ E g/f/|(3e/f/e/ ^d/e/ fg|a3|]
c/d/|ec fc|ec fc|e2 f/e/d/c/|d3 B/c/|dB eB|dB eB|
d2 e/d/c/B/|c3 c/d/|ec fc|ec fc|(3a/b/a/ g/a/ b/a/g/a/|
f3 f/g/|a/b/a/g/ f/a/g/f/|f/e/c/A/ E g/f/|(3e/f/e/ d/e/ fg|a3|]
```

You might think that fixing a tune which generates a bar check error on every single bar would involve a whole lot of editing, but actually that's not the case. This symptom usually results from an error in the M: or L: fields. In the present case, there is no L: field, so the abc standard specifies a fallback value for default note length of 1/16. The tune has been transcribed with four default notes per bar, which adds up to 1/4, rather than the 2/4 specified in the metre field. All that's needed to fix the problem is to change the default note length to 1/8 so that the four default notes per bar add up to 2/4. Insert a new line above the K: field and type:

```
L:1/8
```

and all will be well.

```
X:13
T:Metre missing
T:Star of Munster, The
R:Reel
S:Old O'Neill's
K:G
ed|c2Ac B2GB|AGEF GEDF|EAAB cBcd|eaaf gfed|\
cBAc BAGB|AGEF GEDF|EAAB cded|cABG A2 z2||\
eaab ageg|agbg agef|{a}gfga gfef|geaf gfed|\
eaab ageg|agbg agef|~g2ge ~a2af|b2ba gfed||
```

This is an easy one. Some programs allow you to leave out the metre (M:) field, and default to 4/4 in its absence. BarFly doesn't do this because the abc standard says that the M: field is essential. If you really want to write non-metric music you can do so, but you must use a metre of 'none':

```
M:none
```

To fix the tune, just put in a metre field of 4/4 or C.

```
X: 14
T:Mismatched slurs
T:Lady Mary Ramsay
M:4/4
L:1/8
S:J.Winder MS,Wyresdale,Lancs.,1789
R:Reel
O:England
A:Lancashire
K:D
D2-D>B (A<F)F>B|A<FB<G A<FA<d|D2-D>B A>FF>A|(3Bcd) (3AGF) E2-E<e|
D2-D>B A<FF<B|A<FB<G A<FA<d|D2-D>B A>FF>A|(3Bcd) (3AGF) E2 E<g|
f<dd>f e<cc>e|d<BB>d A<FF>g|f<dd>f e<cc>A|(3Bcd) (3AGF) E2-Eg|
f<dd>f e<cc>e|d<BB>d A<FF>A|(3def) (3edc) (3Bcd) (3AGF)|(3GAB) (3AGF) E2 Ee||
```

The opening parenthesis followed by a number, e.g. (3 is used by abc to denote a tuplet. This is NOT matched by a closing parenthesis. This confuses beginners, especially if they are programmers or mathematicians, to whom closing parentheses is second nature! Parentheses used to denote slurs are closed, and BarFly will check that they match, also checking other paired symbols such as chord brackets [] and guitar chord quotes. In the above tune, the transcriber has carefully closed parens after every triplet, leading to a long list of mismatched slur errors, and a very messy display as the program draws all the half-slurs back to the beginning of each line. It's quite common to see the opposite error too, where the transcriber omits the number from an intended triplet, leading to an unmatched slur error and a bar check error on the same bar.

The best way to fix this tune is to use the Replace Text command again.
Place the insertion point above the tune,
Choose Replace Text
Enter) in the Search for box and leave the Replace with box empty.
Don't check the Replace All checkbox (or you will take out every closing parenthesis from here on).
Hit OK
Type Command-T repeatedly until all the closing parens have been deleted.

```
X:15
T:Plays at the wrong speed
T:Spanish Dance
M:2/4
L:1/16
Q:100
S:Harrison & Wall MS,Ashover,Derbyshire,1762-75
R:Reel
O:England
A:Derbyshire
K:D
DAFA DBGB | D2ef gff[Ae] | DAFA Defg | fedc {c} d4 :|!
|:ecAg fdAa | ecAg [d2f2]eA | afdA BcdG | F2E2 D4:|!
|:ABcd dcBA | efgf f2e2 |f2f2 e^GAB | cBA^G {G}A4:|!
|:e3c  A2d2 | BAGF {F}E4 | Begf ecdg | fedc {c}d4:|
```

Now I know that they play reels more slowly in England than they do here in Scotland, but this is ridiculous! It sounds more like a funeral march than a dance tune. The Q: field in abc determines the tempo, and an entry of Q:100 means to play it at 100 default notes per minute. The default note length is 1/16, so here it means play it at 100 sixteenths or 25 quarter notes per minute, which is just about right for a funeral. The presence of the exclamation marks at the ends of the lines may give you a clue to the explanation - it's our old friend abc2win again. That program comes with a DOS abc player called PlayQABC which notoriously misinterprets the Q field and plays everything at the wrong speed. This leads people to just enter a number which gives the right speed on their machine without thinking what it means. Furthermore, PlayQABC doesn't even seem to be consistent about speeds so I can't just tell you to multiply the entry in the Q: field by some factor to fix the problem. For an English reel I would guess about Q:300 would be about right, or if you want to be a bit more musical, write Q: 1/4 = 75 (i.e. play 75 quarter notes per minute).

```
X:16
T:One or more of the specified parts could not be found in the tune
T: \'Ciro Kolo
O: Croatia
M: C|
L: 1/8
P: A
K: A
|: "A"[AE][cA] [ec][ec] [ec][fd] [ec][cA] | "E7"[ec][ec] [dB][dB] "A"[cA][cA] [BG][AE] :| \
|: "A"[c2A] "E7"[ec][BG] "A"[c2A] "E7"[ec][BG] | "A"[cA][cA] "E7"[BG][BG] "A"[A4E] :|
P: D
K: D
|: "D"[dA][fd] [af][af] [af][bg] [af][fd] | "A7"[af][af] [ge][ge] "D"[fd][fd] [ec][dA] :| \
|: "D"[f2d] "A7"[af][ec] "D"[f2d] "A7"[af][ec] | "D"[fd][fd] "A7"[ec][ec] "D"[d4A] :|
P: G
K: G
|: "G"[GD][BG] [dB][dB] [dB][ec] [dB][BG] | "D7"[dB][dB] [cA][cA] "G"[BG][BG] [AF][GD] :| \
|: "G"[B2G] "D7"[dB][AF] "G"[B2G] "D7"[dB][AF] | "G"[BG][BG] "D7"[AF][AF] "G"[G4D] :|
```

This is quite a subtle one. The transcriber has chosen to use the P: field to label the three parts of the tune with their keys, A, D and G. This is fine: you can use P: fields as simple labels provided that you don't have a P: field in the header, as this will be interpreted as an instruction to play the parts in the indicated order. He has also placed each P: field before the K: field, which is generally correct, as the key change should come at the beginning of each part, rather than at the end of the previous one. The problem comes with the first K: field, as this also has one very special function - it marks the end of the header. This means that the P: field above it is in the header rather than in the tune proper. The program will therefore interpret it as an instruction to play only part A. Since there is no P:A field in the tune proper it reports the error "One or more of the specified parts could not be found in the tune", and just plays the whole tune. The solution in this case is to move the P: A field down to below the first K: field.

```
X:17
T:Unequal notes in chords
T: Rachenica for Three
T: ABABCDECDECFGCFGCDEC
O: Bulgaria
Z: John Chambers <jc@eddie.mit.edu> http://eddie.mit.edu/~jc/music/
M: 7/8
L: 1/8
K: G
P: A
[| "G"[B2G] [B2G] [B2G][cA] | [d2B] [d2B] [d2B][ec] | "D7"[f4d] [e3c] | "G"[d7B] \
|  "G"[B2G] [B2G] [B2G][cA] | [d2B] [d2B] [d2B][ec] | "D7"[f4d] [e3c] | "G"[d7B] |]
P: B
|: "G" [d4B] "D7"[c3A] | "G"[B4G] "C"[e3c] | "G"[d2B] [d2B] [cA][BG][cA] | "G"[d7B] \
|  "G" [d4B] "D7"[c3A] | "G"[B4G] "C"[e3c] | "G"[d2B] [d2B] [cA][BG][cA] | "G"[B7G] :|
P: C
|: "G"[BG][cA] [BG][cA] [d2B][dB] | "D7"[dB][ec] [dB][cA] "G"[B3G] | "G"[BG][cA] [BG][cA] [d2B][dB] |
"D7"[dB][ec] [dB][cA] "G"[B3G] \
|  "G"[BG][cA] [BG][cA] [d2B][dB] | "D7"[dB][ec] [dB][ec] [f2d][ec] | "D7"[dB][ec] [ec][dB] [dB][cA][cA] |
"G"[B7G] :|
P: D
[| "G"[B2G] [c2A] [d3B] | [B2G] [c2A] [d3B] | "C"[e2c] [e4c][ec] | "D7"[dB][cA] [cA][BG] "G"[B3G] \
|  "G"[B2G] [c2A] [d3B] | [B2G] [c2A] [d3B] | "C"[e2c] [e4c][ec] | "D7"[dB][cA] [cA][BG] "G"[B3G] |]
P: E
|: "C"[e2c] [e2c] [e2c][ec] | [e2c] [e2c] [e2c][dB] | "D7"[d2B] [e2c] [f2d][ec] | "G"[e2c] [d2B] [d3B] \
|  "D7"[d2B] [e2c] [f2d][ec] | "G"[e2c] [d2B] [d3B] | [B2G] [c2A][d2B][dB] | "D7"[dB][cA] [cA][BG] "G"[B3G] :|
P: F
[| "G"[d2B] [e2c] [ec][dB][ec] | "D7"[f7d] | [dB][ec] [e2c] [ec][dB][cA] | "G"[d7B] \
| "G"[d2B] [e2c] [ec][dB][ec] | "D7"[f7d] | [dB][ec] [e2c] [ec][dB][cA] | "G"[d7B] |]
P: G
[| "G"[d2B] [e2c] [dB][cA][BG] | "D7"[c7A] | [d2B] [e2c] [ec][dB][cA] | "G"[B7G] \
| "G"[d2B] [e2c] [dB][cA][BG] | "D7"[c7A] | [d2B] [e2c] [ec][dB][cA] | "G"[B7G] |]
```

Here, only the first note of each chord has been given a length. Early versions of abc2ps required chords to be written this way, and some of the old tunes transcribed for it are still to be found on the net. It doesn't cause a serious problem, in fact the error-checking doesn't notice it since BarFly's player just assumes that all notes in a chord are the length of the first one. The display will look rather strange though. Unfortunately there's no quick fix for this, other than to replace all the [B2G] chords with [B2G2] etc.

The strange looking second title:

```
T: ABABCDECDECFGCFGCDEC
```

is another simple error. It's clearly intended as an instruction to play the parts in this order, and should have been in a P: field.

```
X:18
T:Displays staves in the wrong order
T: Candlemas Eve
S: Hymn 126 Arr. R. Herrick from an old church-gallery book
M:4/4
L:1/8
Q:1/8=400
N:from an old church-gallery book
H:The old church-gallery book was discovered by the Rev. L.J.T. Darwall.
H:The source has a 4-part harmony.
O:English
K:G
V: 1
% soprano
D2 |\
G2  G2  B2  G2  | E2  F2  G2  Bd  | c2  B2   A2  G2  | A6  Bc  |
d2  B2  G2  AB  | c2  A2  F2  GA  | B2  G2   E2  F2  | G6  Bc  |
d2  d2  d2  B2  | e2  c2  A2  Bd  | c2  B2   A2  G2  | d6  B2  |
e2  d2  c2  B2  | A2  G2  F2  GA  | B2  G2   E2  F2  | G6  z2  ||
V: 2
% alto
D2 |\
D2  C2  B,2 D2  | C2  C2  D2  D2  | G2  G2   E2  E2  | F6  G2  |
G2  F2  E2  D2  | C2  E2  D2  E2  | D2  D2   C2  C2  | D6  G2  |
G2  G2  G2  G2  | G2  G2  F2  D2  | G2  G2   E2  G2  | F6  D2  |
C2  D2  EF  G2  | E2  E2  D2  E2  | D2  B,2  C2  D2  | D6  z2  ||
V: 3
% tenor
D,2 |\
G,2 G,2 G,2 G,2 | G,2 A,2 B,2 B,2 | E2  D2   C2  B,2 | D6  D2  |
D2  D2  B,2 G,2 | E,2 A,2 A,2 C2  | G,2 G,2  G,2 A,2 | B,6  DC |
B,2 D2  B,2 D2  | C2  E2  D2  B,2 | C2  D2   C2  G,2 | A,6 G,2 |
G,2 G,2 C2  D2  | CD  CB, A,2 C2  | G,2 G,2  A,2 A,2 | B,6 z2  ||
V: 4
```

```
% bass up one octave
D2  |\
B,2 A,2 G,2 B,2 | C2  A,2 G,2 G2  | E2  G2   A2  E2  | D6  GA  |
B,2 D2  E2  E2  | A,2 C2  D2  C2  | B,2 A,B, C2  A,2 | G,6 G,2 |
G2  B2  G2  G2  | c2  C2  D2  G2  | E2  G2   C2  E2  | D6   G2 |
C2  B,2 A,2 G,2 | A,2 C2  D2  C2  | B,2 E2   A,2 D2  | G,6  z2 ||
```

This is not a "bad tune". The problem here is that we are dealing with multivoice abc, and multiple voices are not covered by the abc standard. Each program that implements multivoice has its own way of doing things, and you will often find transcriptions which don't work correctly in BarFly. In the present case, the tune has been entered in four voices, and each V: field is followed by multiple lines of music, like this:

```
K:G
V:1
<several lines of abc>
V:2
<several lines of abc>
V:3
<several lines of abc>
V:4
<several lines of abc>
```

BarFly can play tunes entered in this way with no problems, but in order to display the music it needs the lines of abc to be entered interleaved with one another so that they are in the same order as the staves will be drawn in the music:

```
K:G
V:1
<first staff line>
V:2
<second staff line>
V:3
<third staff line>
V:4
<fourth staff line>
%
V:1
<fifth staff line>
V:2
<sixth staff line>
V:3
<seventh staff line>
V:4
<eighth staff line>
etc.
```

Fortunately, you don't have to do all that editing by hand. You can use the Format tune command from the Edit menu to do the job for you. Place the insertion point anywhere in the tune and choose Format tune. In the dialog, set 6 bars per line and a maximum of 48 notes per line. Click OK and the program will open a new window containing the rearranged tune.

```
X:19
```

```
T:Key change doesn't work
T:Example
M:3/4
L:1/8
K:C
[V:1] g2-|(g2fe) (fd|eG) (eG) (ec|dG)(dG)(dc|eG)(eG)(ec|fG)fG fd|egegab|c'bc'gfd|c4:|
[V:2] e2-|(e2dc) (dB|cE) (cEcA)|BDBDBD|cEcEcE|dFdFdF|cece fd|egecdG|e4:|
K:G
[V:1] d'2-|d'2c'b c'a|bd bd bg|ad ad ag|bd bd bg|c'dc'd c'a|bdbdbg|ac'dc'bd|g4:|
[V:2] [B2-D2-]|[B2D2](ag) (af|gB) (bd)(bg|fA)(fA)fA|gBgBgB|acacac|gBgBgB|faca gd|B4:|
```

This is an example tune sent to me by a user (and slightly edited by me to simplify it). The problem is that the key change from C to G only seems to affect the second voice, and when the tune is displayed there is a key signature only in the second voice, so with one extra symbol in that voice the bars don't line up when Align Voices is turned on. The explanation is that the voices in multivoice abc are treated as separate tunes sharing a common header, and once the tune proper starts, there's no place to put a field that affects both voices. There's nothing to stop you writing a tune with voices in different keys or metres, and if you want both voices to change key at the same time you must put in two key changes, one in each voice.

You might think that this would be the way to do it:

```
X:20
T:Key change still doesn't work
T:Example
M:3/4
L:1/8
K:C
[V:1] g2-|(g2fe) (fd|eG) (eG) (ec|dG)(dG)(dc|eG)(eG)(ec|fG)fG fd|egegab|c'bc'gfd|c4:|
[V:2] e2-|(e2dc) (dB|cE) (cEcA)|BDBDBD|cEcEcE|dFdFdF|cece fd|egecdG|e4:|
K:G
[V:1] d'2-|d'2c'b c'a|bd bd bg|ad ad ag|bd bd bg|c'dc'd c'a|bdbdbg|ac'dc'bd|g4:|
K:G
[V:2] [B2-D2-]|[B2D2](ag) (af|gB) (bd)(bg|fA)(fA)fA|gBgBgB|acacac|gBgBgB|faca gd|B4:|
```

but that doesn't work either. The reason is that the key changes have been placed before the V: fields, so that they are located not at the start of the part to which they apply, but at the end of the previous line of music, which belongs to a different voice. So the first key field is at the end of the first part of V:2 (and appears at the beginning of the second part of V:2), while the second key field is at the end of the second part of V:1 (and doesn't appear at all, because the tune is finished). To make this work properly, we have to place the K: fields after the V: fields at the start of the part like this:

```
[V:1] g2-|(g2fe) (fd|eG) (eG) (ec|dG)(dG)(dc|eG)(eG)(ec|fG)fG fd|egegab|c'bc'gfd|c4:|
[V:2] e2-|(e2dc) (dB|cE) (cEcA)|BDBDBD|cEcEcE|dFdFdF|cece fd|egecdG|e4:|
%
[V:1] [K:G] d'2-|d'2c'b c'a|bd bd bg|ad ad ag|bd bd bg|c'dc'd c'a|bdbdbg|ac'dc'bd|g4:|
[V:2] [K:G] [B2-D2-]|[B2D2](ag) (af|gB) (bd)(bg|fA)(fA)fA|gBgBgB|acacac|gBgBgB|faca gd|B4:|
```

or if you prefer the in-tune fields on separate lines, like this:

```
V:1
 g2-|(g2fe) (fd|eG) (eG) (ec|dG)(dG)(dc|eG)(eG)(ec|fG)fG fd|egegab|c'bc'gfd|c4:|
V:2
 e2-|(e2dc) (dB|cE) (cEcA)|BDBDBD|cEcEcE|dFdFdF|cece fd|egecdG|e4:|
%
V:1
K:G
 d'2-|d'2c'b c'a|bd bd bg|ad ad ag|bd bd bg|c'dc'd c'a|bdbdbg|ac'dc'bd|g4:|
V:2
K:G
 [B2-D2-]|[B2D2](ag) (af|gB) (bd)(bg|fA)(fA)fA|gBgBgB|acacac|gBgBgB|faca gd|B4:|
```

You can see the same problem with any in-tune field in multivoice abc, so if you need to change key, metre, rhythm or tempo you must put the field in all of the voices, otherwise chaos will ensue.

```
X:21
T:Lots of leger lines!
T: Sonata I
C: J.S. Bach
M: C
Q:100 %"Adagio"
L: 1/8
V:1 clef=treble name="Violino I"      sname="Vl. I"
V:2 clef=treble name="Violino II"     sname="Vl. II"  space=+10
V:3 clef=bass   name="Violoncello\\<Vla. da Gamba>" sname="Vc."
K:C
P:Adagio
V:1
 g8-|gf/e/ {e}f>g (a/f/d/f/) (A//=B//A//B//TB3//A///B///)|
V:2
 z8 | z8 |
V:3
 z cec gGBG | Aa- a/_b/a/g/ f3 g/f/ |
%
V:1
 c/gf/ E/ed/ c/c'b/ A/ag/ | ^f/e/d- d/(c/B/A/) G/(e/c/e/) Aa| d2-d/g/_b/a/ a3 g/=f/|
V:2
 c8- | cB/A/ {A}B>c (e/c/A/c/) (E//^F//E//F//TF3//E///F///) | G/(D/G/A/) _B/G/g/e/ ^cA d2-|
V:3
 edcB AG^FE | D^FGg c3d/c/| _BG g2-gf/e/ f>g|
%
V:1
 Te/>d/e- e/(a/f/d/) TB/>A/B- B/(A//B//c/)d/| TG/>F/G- G/(c/A/F/) DTd/>c/ d/g/B/d/|Tc/>B/c2 B A/ag/ fA|
V:2
 dT=c/>=B/ c>f d>e AF-| FTE/>D/ Ec'/a/ Tf/>e/f- f/d/g/f/ | Te/>d/e/g/- g/fe/ d^c d/FE/|
V:3
 aA a2-a^g a>b| c'c c'2-c'b/>a/ b2-| b>e ag fed=c-|
%
V:1
 G/gf/ E/ed/ c/c'b/ A/ag/|^f/a/g/f/ g2-g>e' f2-|f/e^d/ e2-e>c' =d2|
V:2
 DG, G/gf/ E/ed/ =c/c'B/|ad'- d'/c'/d'/e'/ Ta/>g/a- a/c'/b/^f/|(g//a//g//a//Ta3//g///a///) b/a/b/c'/ T^f/>g/f- f/a/g/d/|
V:3
 cB/G/ cC zgac- |cb/>a/ b>c' d'c'd'd| e^fga babB|
%
P:Alla breve
```

```
V:1
L:1/4
 c4|B4|_B4|A2zd|=BGc2-|cd/e/ f/a/g/f/|e/d/c2B|e3^f|gG/A/Bc|
V:2
L:1/4
 zC/D/EF|GDG2-|GF/E/DE|FCF2-|FE/D/EF/G/|AFDB|cgG2-|GA/B/ c/e/d/c/|Bd2e|
V:3
L:1/4
 z4|z4|z4|z4|z4|z4|z4|z4|g4|
```

There appears to be something wrong with the Cello voice here; it's written on the bass staff, but all the notes are high up above the staff, and using lots of leger lines. Furthermore, when you play it that voice plays two octaves higher than it should. This tune has been transcribed for one of the abc2ps family of programs, and those programs interpret clefs other than the treble clef differently from BarFly. BarFly keeps the meaning of note symbols as described in the abc standard the same regardless of what clef is in force, so the symbol C always means middle C, and is drawn one leger line below the treble staff or one leger line above the bass staff. The abc2ps family changes the definition so that when a bass clef is in operation, users will enter the music two octaves higher than BarFly users (this avoids using lots of commas when typing bass lines).

You could transpose the cello part down by two octaves, but that's going to involve a lot of work. Fortunately there is a much simpler solution. BarFly will let you change the vertical position where the notes are drawn on the staff using the "middle" directive. By default, the middle line of the bass staff corresponds to the note of D, but here we want it to correspond to the note of d. The middle directive can be placed in the K: field (it would then affect all of the voices) or in the appropriate V: field in the header. In this case we want to put it in V:3 to affect only that voice. So, if we add "middle = d" or just "m=d" to the V:3 field in the header the music will display correctly.

The middle directive only affects display; to fix the tune to play properly we need to use another directive "transpose". This will cause the voice to be transposed by the number of semitones given, so if we add "transpose = -24" or just "t=-24" to the V:3 field in the header the tune will play correctly too. Those V: fields in the header are full of stuff that BarFly doesn't understand (and will ignore), so we might as well tidy them up and place some definitions that BarFly will understand. If we change the three V: header fields as follows, BarFly will play the music using the indicated instruments, and in the correct octave, and will also display the tune correctly:

```
V:1 program 1 40 %Violin
V:2 program 1 40 %Violin
V:3 bass program 1 42 m=d t=-24 %Cello displayed and played two octaves lower than written.
```

(Mind you, I find Quicktime's implementation of the violin family unspeakably horrid, and almost any other combination of instruments will sound better.)

Back to the table of contents:

file://:Table of Contents