Using bagpipe.tex

1. Introduction

Bagpipe.tex is a macro-package for use on top of MusicTeX, or MusixTeX, which are macro-packages for use with TeX or LaTeX. TeX is a language designed for typesetting technical documents. It is public domain and is available for most platforms. Musi(c|x)TeX were written and are maintained by Daniel Taupin. They are available via anonymous ftp at the CTAN sites, and ftp.gmd.de . Bagpipe.tex itself is available via WWW from http://www.stanford.edu/w̃rinnes/bagpipe.tex/. Musi(c|x)TeX files are usually found in MSDOS|Windows format, and bagpipe.tex files in UNIX format. If you transfer them in binary mode (as you would for a compressed TAR or zip file), you may have to do some conversion to get the carriage returns and line feeds correct. TeX itself is not too fussy about such things, but a platform which doesn't see a line break in the files may cause TeX to choke (this happens on MACs if you don't convert the sources).

This write-up is for version 3.02. Version 3.02 of bagpipe.tex works with version 5.20 of MusicTeX and version T.89 of MusixTeX. Since it redefines a few Musi(c|x)TeX macros, it may not work with later versions without modification. It will not work with versions of MusicTeX before 4.7. Musi(c|x) TeX are designed to set complex multi-part scores. As a result, as delivered, typesetting bagpipe music is needlessly tedious. So, as suggested by Taupin in his write-up, I have written macros for commonly recurring complex codings. The result is that very few Musi(c|x)TeX macros appear directly in a score set with bagpipe.tex. Nevertheless, one should read Taupin's documentation to get an understanding of Musi(c|x)TeX's mode of operation and to find those macros which **are** used. I will not describe any Musi(c|x)TeX macros except when they interact with bagpipe.tex macros.

MusixTeX differs from MusicTeX in that it relies on a three pass system to adjust the spacing instead of glue (stretchable space). MusixTeX is also the version under active development and has many features which MusicTeX does not. Most of these are irrelevant for bagpipe music. As of version 3.02, MusixTeX is the default. To revert to MusicTeX, comment out the line:
\let\usemusixtex\relax% this sets the flag to use MusixTeX instead of MusicTeX, which is located near the beginning of the bagpipe.tex file.

2. Basic definitions, melody notes.

The (usually) single part and limited range of bagpipe music allows for simplification. There are two schemes for indicating pitch. The default is the new scheme

in which pitch values are designated by one of N, a, b, c, d, e, f, g, h.

In the following I will represent one of these by x. To specify a pitch one types x or **x**. In forming macro names G may be used in place of N and A may be used in place of h. These last alternates may also be used for pitches providing a leading \ is used. An alternative scheme may invoked by **oldpitch**. In this alternate scheme the pitch values are designated by one of g, a, b, c, d, e, f, G, or A. A a leading \ is required. The new scheme may be restored at any time with a **newpitch**. In the examples in the rest of this note I will use the new scheme as do the sample tunes.

To specify an isolated (not connected to a beam) melody note type **whx** for a whole note, **hx** for a half note, **qlx** for a quarter note, **cx** for an eighth note, **sx** for a sixteenth note, **tx** for a thirty-second note, and **sfx** for sixty-fourth note. Dotted melody notes are specified **hp x**, **qlp x**, **cp x**, **sp x**, **tp x**, **sfp**. Similarly, **hpp x**, **qlpp x**, and **cpp x** are used for double dotted notes. Beamed notes are specified as arguments of beam macros which will be described below. The arguments are **bx** for a plain note, **bxp** for a dotted note, and **bxpp** for a double dotted note. The time value is controlled by the beam macro.

3. Embellishments.

The big breakthrough in MusicTeX was to make possible arbitrary grace note sequences. I have coded macros for all the common and many not so common bagpipe grace notes and embellishments. These macros specify only the grace notes proper and not the following melody notes. Simple grace notes and strikes are entered as **grx**. Doublings are **dblx**, half doublings **hdblx**, thumb doublings **tdblx**, and slurred doublings **sdblx**. The throw on D is **thrwd** and on f is **thrwf**, the grip is **grip**, and the grip from d is **dgrip**. The **thrwf** is the same as the dare **dare**. There is also a grip-throw on D (with an extra low G), **gripthrwd**. The half throw on d is **hthrwd** and the throw on d from low G is **Nthrwd**. The grip from low G is **Ngrip**. The note sequences specified by these grips are often called by other names in particular contexts such as throws or leumluaths. I have not defined many such synonyms. When coding, look at what grace notes are actually there, not at the surrounding melody notes. The taorluath is **taor**, the taorluath from g is **gtaor**, and from d **dtaor**. The half taorluath from low A is **ahtaor**. The birl is **birl**, the strong birl **sbirl**, the weak birl **wbirl**, and the thumb birl **tbirl**. The slurs take x values a to f only. They are the slur **slurx**, half slur **hslurx**, and the thumb slur **tslurx**. In all of the slurs, "slap" may be used in place of "slur". Similarly shakes are **shkx** for x from a to g. There are also half shakes **hskx** and thumb shakes **tshkx**. Note that for a, b, c, and e shakes and slurs are the same. I have not defined such a synonym. There are slurred doublings **sdblx**, slurred half doublings **shdblx**, and slurred thumb doublings **stdblx**. There are the shaken doubling and thumb doubling on d **shkdbld**, **shtdbld**.

Some less commonly used embellishments are the double grace notes **ydgrx** where y goes from d to t (t instead h), the double strikes **dstrx**, strong double strikes **gdstrx**, thumb double strikes **tdbstrx**, and the half double strikes **hdbstrx**. There is also the special case of the light half doulbe strike on D **lhdbstrd**. In exactly the same pattern as the double strikes there are triple strikes **tstrx**, etc. For x=b,c,d,e there are the pele's, sometimes called hornpipe shakes **pelx**, **tpelx**, **hpelx**, **lpeld**, **ltpeld**, and **lhpeld**. For d and e these are the same as the slurred doublings. For x values of a through d there are catches **catchx**, strong catches **scatchx**, and thumb catches **tcatchx**.

Note that some named movements such as tachums are not represented here because they consist of a combination of grace and melody notes. They need to be broken down for type setting.

Piobaireachd embellishments include the crunluaths **crun**, **crunf**, **ahcrun**, **dcrun**, **Nhcrun**, crunluath an machs **crunmb**, **crunmc**, **crunmd**. The crunluath breabachs are **crunbr**, **dcrunbr**, **ahcrunbr**. **chelalho**, and the darodos **darodo** and **Ndarodo**. "darodo" may be replaced by "bubbly". The **crunmd** is the same as the edre on d **edred**. There also the dre **dre**, the edre **edre**, and edre's on b and c **edrex**. Other dre's and dare's are **gedre**, **gdare**, **tedre**, bsltdare, **Nedre**, **aedre**, **hdre**, **hedale**, **hchechere**, **hedari**, **tchechere**.

The piobaireachd throws are **enbain**, **otro**, **odro**, **adela**, and similarly with leading g or t for the strong or thumb variations. There are also **dbstf** and **dbsth**. Special piobaireachd throws on D are **tra**, and **trac**. Themal grace notes are coded **themx** and echo notes **echox**. The cadences are coded **cadxx** or **cadxxx**. The retarded variations are **fcadxx** or **fcadxx**.

If your favorite embellishment is not (?!!) included, note that any set notes may turned into grace notes using them as the argument of **ggr**{}. For the standard thirty second note embellishments, you can use the **multigr** and **smultigr** macros. See the bagpipe.tex source for many examples.

Versions 2.00 and later offer a different means of generating grace note spacing. This option is invoked by **newgrace** or **multipart**. A byproduct is that the afterruleskip can be be restored to closer to the MusicTeX standard value making typesetting somewhat easier. The old (default) scheme may be restored with the macro **oldgrace**. Good practice would dictate that you put one or the other at the beginning of each piece to prevent problems if you later mix sources or the default changes in a future version. I recommend the new mechanism.

4. Beamed note pairs.

Because of there being only two notes, the location and slopes of beams for note pairs may be calculated from the note pitches. This allows for comparatively simple entry. Here I describe a group of macros for specifying these note pairs. They all

have the form **pd...d** where there two to four d's. d represents either a note length (c, s, or t) , p indicating that the previous note is dotted, or g for a grace note. For each d (except p) the macro requires an appropriate argument. Note lengths require a pitch value x and a g requires any grace note macro, e.g. **grx**. An example of use is **pcps ab** which would set a dotted eighth note at pitch a beamed to a sixteenth note at pitch b. Another is **psgcp b****grd c** which sets a sixteenth note at pitch b followed by a d grace note and then a dotted eighth note at pitch c. The b and the c are connected by a beam. The complete list of macros can be determined by looking into bagpipe.tex. All reasonable combinations are defined.

5. More general beamed note groups.

Combinatorics being what they are, the above approach becomes impractical for more than two notes. Also there are a few cases where more flexibility is required even for pairs (e.g. sometimes the auto beam code puts the beam on top of the grace note !). For these reasons I have defined more general macros. These have the form **bd...d** where there from two to four d's. The d's represent time values (c, s, or t). These control the beam structure. The first three arguments specify the beam: starting pitch, ending pitch, and length in note spacings. Often the pitches will be those of the starting and ending note but may differ for many reasons as you will soon find out. The length is an integer from 1 to 9. An additional argument is required for each d. In their simplest form these arguments are beamed note specifiers **bx**, or **bxp**. If there is a grace note between two melody notes within the beamed group, it is specified by replacing the argument for the first melody note by {**bx****g**} or {**bxp****g**} where **g** represents any grace note macro. An example which might be found in a jig is **bccc cc3**{**bc****grd**}{**bc****gre**}**bc**. Again the complete list of macros can be found in the code. It is intended that all possibilities are defined up to 4 'd's. Longer beamed groups can be specified if the number beams for successive notes (not including the last) are the same. This is done by including multiple **bx** macros within a pair of braces.

A note for TeX neophytes: macro names consist of letters only. Thus they are terminated by spaces, \'s, braces and numbers. If a macro requires arguments the processor will take the following "tokens" until a sufficient number have been found to feed to the macro. The braces serve to group macros so that they will be counted as one argument.

bgrx was a work around for a bug which has been squashed. It has been removed.

6. Multiple parts

Version 2.00 introduces the ability to set multiple parts (parts as in the melody plus seconds). The standard Musi(c|x)TeX instrument and staff definitions will work

(if you invoke **noautoglue**, see below), but if all your parts are on bagpipe, you can use **multipart**{n} where n is the number of parts. Note that **multipart**{**1**} is a simple way of setting the options for the newgrace note scheme and no autoglue. If you are content to line up only the beginning of the beats, you can use all bagpipe.tex macros. Otherwise you are on your own for beams. Start a bar with **notes**, set the first beat of the bottom part, enter a | then the corresponding beat of the next part. Proceed through all the parts and then terminate with a **etn**. Then enter the music for the bottom part of the next beat and continue on. Terminate the last beat of the bar with an **enotes**. A bar in a two part, two beat per bar piece would go like:

notes...|...**etn**...|...**enotes****xbarre**

If you do go your own way, you may have pending beams in both parts. In this case you must specify different beam numbers for the parts. Some of the macros have these beam numbers built in. 0 is used by the beam macros and 1 is used by the grace note macros, so avoid using these in your own beam number assignments.

|'s and &'s mean different things within the music environment than without. Thus the typical pattern of defining bars before **debutmorceau** will not work. This can be corrected two ways. The bar definitions can be sandwiched by a **catcodesmusic** ... **endcatcodesmusic** pair, or the defintions can be put immediately after the **debutmorceau**. If you do the latter, be very careful that there is no extraneous white space in the defintions. All definitions should be immediately followed by a **relax**.

An **enotes** (or **etn**, **enn**, **ttn**) terminates a column of notes. Thus none of these can used in the interior of an intended note column. If you are building columns by the beat, then there can be none of these (and no glue) within the beat. This situation may arise when taking existing music for which the parts have been set separately and merging them into one multipart setting.

7. Miscellaneous macros.

Pick-up notes require some special bookkeeping to keep the bar count correct. I have provided the following macro **pickup**{any pick-up notes}. This enters the specified notes, decrements the bar counter, and puts in a bar. Unfortunately this puts the bar number on the pick-up bar if this is not the first bar in the tune. Ending the previous line with **suspmorceau** and using **reppickup** in the place of **reprmorceau****pickup** will fix this problem. It decrements the bar counter before the new line is started, thus suppressing the bar number on the pickup bar. An additional complication arises if the new line begins with a repeat symbol. In this case use **repreppickup** in place of **reprmorceau****leftrepeat** **pickup**. The details of how these macros behave differ in the autoglue and noautoglue schemes.

psk, and **fsk** provide for small horizontal spaces (they stand for point skip and flag skip). They are related to the note width for open music and to the note spacing

for tight music.

Downtext and **downtext** put text under the staff. These can be used for piobaireached shorthand and large scale repeat notations such as *Da Capo al Segno*.

etn is the same as **enotes****temps****notes** which inserts some "glue". Glue stretches when there is space left on the line. I often use this after quarter notes and on beat boundaries in general. By default, beamed groups have one of these built in, so don't put one in explicitly. There is also implicit glue before a bar line which can double the glue at this point. To compensate one can put **etn** when the beat ends in a beamed group and **ttn** when it does not. **ttn** is **etn****etn**. Best, turn off this feature with **noautoglue** and then use **etn** consistently between beats. No autoglue also turns **ttn** into **etn** so one gets about the same effect autoglue or no. **autoglue** turns autoglue back on. Note that autoglue is not compatible with multiple parts and **multipart** turns it off.

normalwidth, **medwidewidth**, **widewidth** set up margins for various width layouts and odd and even pages. **normalheight**, **medtall**, and **tall** set vertical size and margins.

Afour and **USletter** set the paper size. US letter is the default. Note that music optimized for one paper size will not look as good (or may not fit at all) on the other.

today inserts the date. **USdate** and **Eurodate** specify the format. The US order (month day, year) is the default.

landscape is implementation dependent. It will probably work if you use dvips by Tom Rokicki. **realwidewidth** is landscape with small margins.

8. Miscellaneous usage notes.

This section will refer often to Musi(c|x)TeX macros. You need to have read the Musi(c|x)TeX documentation to follow the discussion.

bagpipe.tex **inputs** the following MusicTeX files: muscinft.tex, musictex.tex, and musicvbm.tex or the follwing MusixTeX files:, musix.tex and musixcpt.tex. It sets Musi(c|x)TeX for single instruments and single staff. It adds teenynotes to the font definitions.

I usually use **meterfrac** for the time signature. You may like to use **meterC** (commonly used for strathspeys) or **allabreve** (commonly used for reels).

I prefer to keep complete control of line breaks. Therefore I use **xbarre**, and **alaligne** instead of **barre**. I recommend using **autolines** except when the music is tight. In this case the bagpipe.tex command **setelemq**{f} will set the elementary note spacing to f times the quarter note width. f=1.2 is the smallest value that looks decent.

TeX treats any number of consecutive spaces, carriage returns, and line feeds as "white space". White space which terminates a macro name is eaten and has no

further effect. Other white space will insert a space and some "glue" in the document. Because Musi(c|x)TeX does not know about this space, any pending beams or slurs will be messed up. Therefore it is best, within the music, to avoid all white space that is not needed to terminate a macro name. To accomplish this, lines which do not end with a macro name should be terminated with a **relax** command.

Musi(c|x)TeX does not like to tie notes that are closer than 2.3 quarternote-head widths. If the music is tightly set, extra space must be inserted between tied notes using **sk**, **fsk**, or **psk**. A dot on the first tied note counts as space. Expanded glue does not. If the tied note is the last one in a beamed group, the **psk** should be grouped with that final note by using braces.

Musi(c|x)TeX appears to have a roundoff problem when constructing sloped beams with more than three notes. As a result there are sometimes little bumps and for certain note spacings near **setelemq**{1.7} small gaps appear. The problems are worst at low resolution such as with a previewer. Work-arounds are: avoid the gap producing note spacing, print at as high a resolution as is available, don't use excessive beam slopes.

In previous versions bagpipe.tex and by default in this version (i.e. unless you invoke **newgrace** or **multipart**), the after bar spacing is quite small to keep the distance from the bar to the first grace notes from being too large. If the bar began with a theme note, this resulted in too small a spacing, thus requiring that the bar begin with a **sk** or **psk** to alleviate this. This value is determined by **stdafterruleskip**. The Musi(c|x)TeX default is 4**Internote**. The default bagpipe.tex value is **Internote**. **newgrace** uses a different mechanism to set grace note spacing and does not suffer to the same degree. The **newgrace** value of **stdafterruleskip** is 3**Internote**. A similar problem can still occur between any pair of theme notes which do not have an embellishment in between. The solution is similar.

In some tunes, it is difficult to squeeze four bars on a line. It may be better to just give up and use more paper. If you wish to persist, here are some techniques to save space. Set the width to **widewidth**. Set the **setelemq** to a number like 1.1. Use landscape or **realwidewidth**. Eliminate beginning of line bars unless there is a repeat on other than the first part. Use the reppickup macros. Use **musicsize**=16.

Bagpipe.tex can be used by putting a **input bagpipe** at the beginning of your file. MusixTeX is the default. To use MusicTeX, the bagipe.tex file must be modified as described above. Alternatively, you can generate and use a bagpipe.fmt or a bagpipex.fmt file. The file bagpipe.ini may be used as the argument of initex to generate bagpipe.fmt (after the above modification to bagpipe.tex) and bagpipex.ini used to generate bagpipex.fmt.

9. MusixTex

The MusixTeX option is now the default. You may not shift between MusixTeX

and MusicTeX in the same job. Be sure to read the fine MusixTeX manual. I give here a brief summary of the operation of MusixTeX and how it affects bagpipe.tex. On the first pass of MusixTeX over tune.tex, MusixTeX (via bagpipe.tex) generates a file tune.mx1 which contains the length of each line. The program musixflx is invoked to generate tune.mx2, which contains spatial scale factors for each line. A second pass of MusixTeX uses tune.mx2 (if available) to generate the final output. In principle, all the fussing needed to get good spacing with MusicTeX is done for you. For this to work, MusixTeX needs to know about ALL the space in the music. There must be NO extraneous white space. See the MusixTeX for descriptions of the likely symptoms and cures for extraneous space problems.

This scheme can result in music which looks significantly different than with MusicTeX. In MusicTeX, only the glue is adjusted to fill out a line. In MusixTeX all spacing is adjusted. Glue is forbidden as are hard (not adjusted by musixflx) spaces. Putting glue on the beats and keeping the music within a beat relatively compact, gives results more in accord with usual bagpipe music setting practice than does MusixTeX. On the other hand, the MusixTeX scheme permits better looking ties because they don't have to stretch.

Another area of different results peculiar to bagpipe music is the spacing of complex embellishments. Stretching the spaces within an embellishment can look ugly. This was a problem with the old grace note scheme that was cured with the new grace note scheme in MusicTeX. The fixed space used by the new grace scheme is not permitted by MusixTeX. For the old grace note scheme I just let them stretch. In the new grace note scheme I put the extra space before the embellishment and keep the embellishment itself compact. This may cause ugliness if the stretch is too large or too small.

The musixtex option in bagpipe.tex **inputs** musixtex.tex and musixcpt.tex. The latter contains macros for compatibility with MusicTeX. Most of these are synonyms for macros whose names changed (usually to English from French) in MusixTeX. For a few such as **autolines**, the actual function has changed due to the fundamental difference between the packages. The default MusicTeX version has a number of synonyms for MusixTeX macros as an aid to backward compatibility for tunes coded in MusixTeX. These cover only a few of the most likely to be used macros and of course functions unique to MusixTeX cannot be handled by MusicTeX. If you care about other people using your settings, I recommend using the MusicTeX versions so that code is compatible with older bagpipe.tex versions. The synonyms are:
en = **enotes**,
instrumentnumber = **nbinstruments**,
startpiece = **debutmorceau**,
bar = **barre**,
itie = **iten**,

\ttie = **\tten**,
\contpiece = **\reprmorceau**,
\stoppiece = **\suspmorceau**,
\Stoppiece = **\finmorceau**,
\Endpiece = **\finmorceau**,
\leftrepeat = **\leftrepeatsymbol**,
\rightrepeat = **\rightrepeatsymbol**,
\setstaffs1#2 = **\nbporteesi**#2.

10. Concluding comments.

The best way to understand this guide is to preview and study the examples. The "example" entitled quickref.tex is a sample sheet. It is often convenient to define a macro for each bar of music. Many definitions will then consist of previous bars, and the music proper will be mostly a list of bar macros. Using the template files and a good cut and paste type editor will save a lot of time. I have found that with practice it takes between one half and one hour to type-set a complex four part 6/8 march. This includes proofing and adjusting. A simple two part 3/4 march can be done in less than fifteen minutes. Send any comments, suggestions, or bug reports to walt@slac.stanford.edu .

Good luck and happy piping,
Walt Innes