Proposal for New Feature(s) in MuseScore 2.0.4

1. Add Points/Pixels as Units in Page Layout Dialog
## 2. Normalize Units for Page Layout in all MuseScore

Justification

I am using MuseScore to create animated SVG exports. The exported files are designed to be placed directly in an HTML page or opened by a browser without an HTML wrapper at all.

- The native units of HTML and SVG are pixels

- HTML and SVG are constantly increasing in popularity and usage, globally.

- For my purposes, as I imagine for many future applications, I want to size my scores in pixels, not inches or millimeters. Inches and millimeters are for printing on physical paper. Many of my scores are designed specifically for display on a screen, not to be printed.

- Fortunately, MuseScore recently aligned its core units, Points, with Pixels. Ever since the 72dpi changes, points and pixels are equivalent in MuseScore. This means that adding Points and Pixels as a combined type of Page Layout unit makes great sense.

- Not only does it make sense to add the user functionality, but in doing so I have eliminated unnecessary unit conversions in the code, and simplified the usage of Page Layout units. Currently Page Layout units are stored internally in Inches, and then converted into all the other units on demand. Making points/pixels the core units means that Millimeters and Inches are treated the same way, as converted unit types for printing/exporting only. The result is fewer, simpler conversions.

- Thankfully, the file where this causes the most change has not been modified much since 2.3: pagesettings.cpp
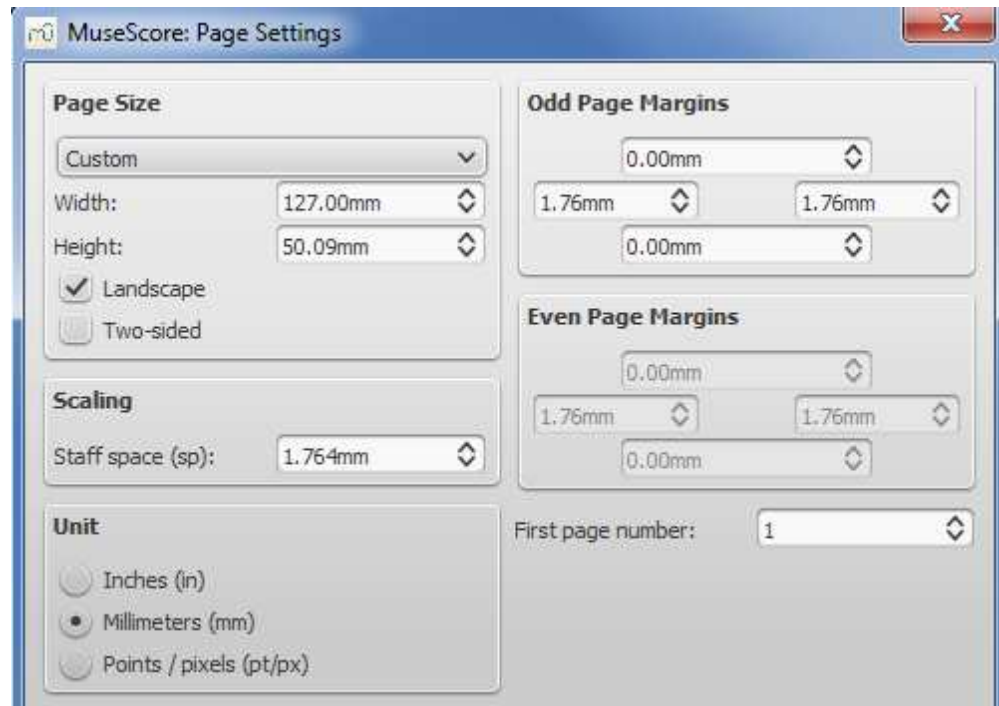
User Interface

The Page Layout dialog must change to add a third radio button to the units options.

BEFORE:
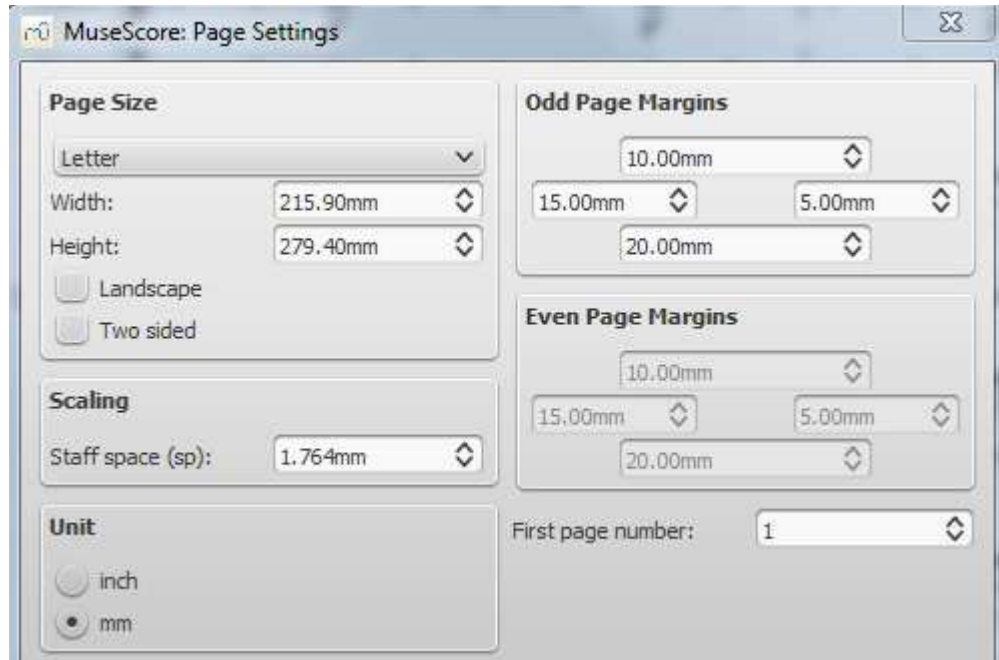
AFTER:

It also increases the maximum width and height of a page from 2,000 to 200,000. The reason for this is that the HTML/SVG canvas is infinitely wide, and pixels are very small. 2,000 is extremely limiting. 200,000 seems reasonable to me, but it could be increased even more. I see no harm in this.

I have also improved the capitalization and naming of the units radio buttons.

Source Code

Page Layout units are converted in a variety of ways in MuseScore. These source code changes simplify that and reduce the number and complexity of unit conversions, by using Points/Pixels as the core units, and only converting to Inches or Millimeters when necessary, which is only when printing or exporting the score.

libmscore/mscore.h

In the spirit of clarity, I have also change the names of some core constants in libmscore/mscore.h, at line 71.

- The constant INCH becomes MMPI, representing millimeters per inch, because that is a clearer and more consistent representation of the constant value.

- The new constant IPMM, inches per millimeter, is useful, and helps remove literal constants in the code

- DPI no longer equals a raw literal 72, it equals PPI, which I believe is important to understand in this context - it's not just the number 72, it's that PPI and DPI are equivalent.

- The new enum Units and the array unitSuffixes are ways of unifying lookups to these values and eliminating literals in the code. I'm not sure if it's constructed correctly, so I'd be happy to see a better way to declare these. These also integrate staff spaces as a unit of measurement too, which is a good thing, IMO. They generate a lot of warnings now because they are in mscore.h. Maybe I should put them in page.h.

So there are now 4 types of units: in, mm, pt/px, and sp. Even if you cannot select Staff Spaces as the Page Layout units, it unifies these in the code, which is helpful.

libmscore/page.h

Two new constants:

- SCALE_XML, for the scaling that exportxml does, at 144dpi. Line 84.
- SIZE_CUSTOM, for replacing literal value zero when referencing the paperSizes array. line 181.

libmscore/page.cpp

Some of these changes are simply to use the new constant names from mscore.h. Some of these are necessary now that Points/Pixels are replacing Inches are the core units in Page Layout.

- PageFormat::PageFormat - reducing reliance on raw literal values and integrating new constants. Also a question: Why is there no printableHeight? What makes width special?
- Removed 4 now unnecessary conversions to Points/Pixels in tm(), bm(), lm() and rm().

libmscore/pagesettings.cpp

Lots of changes. Some of them are to reorganize the code. After adding this new feature, the file is 36 lines smaller (5.6%).

libmscore/pagesettings.h

- A couple of new constants

- bool mmUnit becomes Ms::Units _units. There are more than two types of units now.

- Declarations for 3 new conversion functions in pagesettings.cpp. These are part of the reorganization and consolidation of code in that file.

libmscore/pagesettings.ui

Changes described previously in User Interface section.

libmscore/elementlayout.cpp

Search/replace INCH to MMPI x 6

libmscore/layout.cpp

Removed a now unnecessary conversion.

mscore/mscore.cpp

I removed a single line, a comment. That may seem silly, but that comment redefines the DPMM constant, and I touched that entire group of constants. DPMM is clearly defined in mscore.h.

libmscore/score.cpp

Score::cropPage() - Is this function ever used? I updated it for new constants and such regardless.

Removed unnecessary conversions in loWidth() and hiWidth().

libmscore/scorefile.cpp

Search/replace INCH to MMPI x 2

libmscore/style.cpp

Search/replace INCH to MMPI x 1, which changes to MM macro to be same as in page.cpp.

mscore/exportxml.cpp

These changes simplify the formulas for converting the Page Layout to the 144ppi/dpi scale used here.

Question/Concern: The new code makes getsTenthsFromInches() obsolete. But there is some #ifdef 0 code that still uses it, lines 5080-5083. What's the story with that "commented-out" code?

mscore/file.cpp

Search/replace changes:

- QPrinter::Inch to QPrinter::Point x 3

- QPageSize::Inch to QPageSize::Point x 1

- * DPI to empty string (removing now unnecessary conversions) x 3

mscore/fotomode.cpp

Search/replace INCH to MMPI x 2

mscore/importxmlpass1.cpp

These changes simplify the formulas for converting the Page Layout to the 144ppi/dpi scale used here.

mscore/magbox.cpp

Removed 7 now unnecessary conversions in getMag(). Note that calls to getMag() are dividing by DPI when the pass in arguments. I left this alone.

mscore/musescore.cpp

2 small changes in main() to use new constant and QPrinter::Point and MMPI.

I have a concern about line 4904 and the raw 2.0.0 literal there. Does that relate to SPATIUM?

mscore/preferences.cpp

What's is line 316 doing? I left that conversion alone. Changes made are:

- QPrinter::Inch to QPrinter::Point

- Raw literal 25.4 to MMPI

Also note the hardcoded margins in millimeters in printShortcutsClicked(). I left those alone too.

mscore/scoreview.cpp

No changes, but I'm concerned about line 1857, which calls the modified Page::tm(), bm(), lm(), rm().

mscore/textprop.cpp

Simple edits to use new constants.

mtest/testutils

QPrinter::Inch to QPrinter::Point