

Convert Spaces To Units

Bob Zawalich

A new plugin, Convert Spaces To Units, is available for free download at <http://www.sibelius.com/download/plugins/index.html?plugin=260>.

This is a plugin that not many will want to use. When I offered it to Daniel he said "That's quite the nerdy plug-in ;^)".

Nonetheless, I have found it useful and so am offering it for any nerdy types who might find it useful.

Sibelius stores certain sizes internally in units of "spaces", where a space is defined as 1/4 of the height of the staff, as defined in Document Properties.

The X and Y offsets in the General pane of the Property window are in spaces, and font sizes are internally stored as spaces, which is why your 12 point font shows up in the text pane as 11.9 points (There is a bit of rounding error going from 77-32nds of a space to 12 points). Storing sizes as spaces lets them automatically scale up when the staff size changes.

So say you want to move a text item over an inch. What do you put in the X offset? You can find out that the staff size is .28" and so a space is .07" and you can divide 1 by .07 and thus you would need about 14.3 spaces to do that.

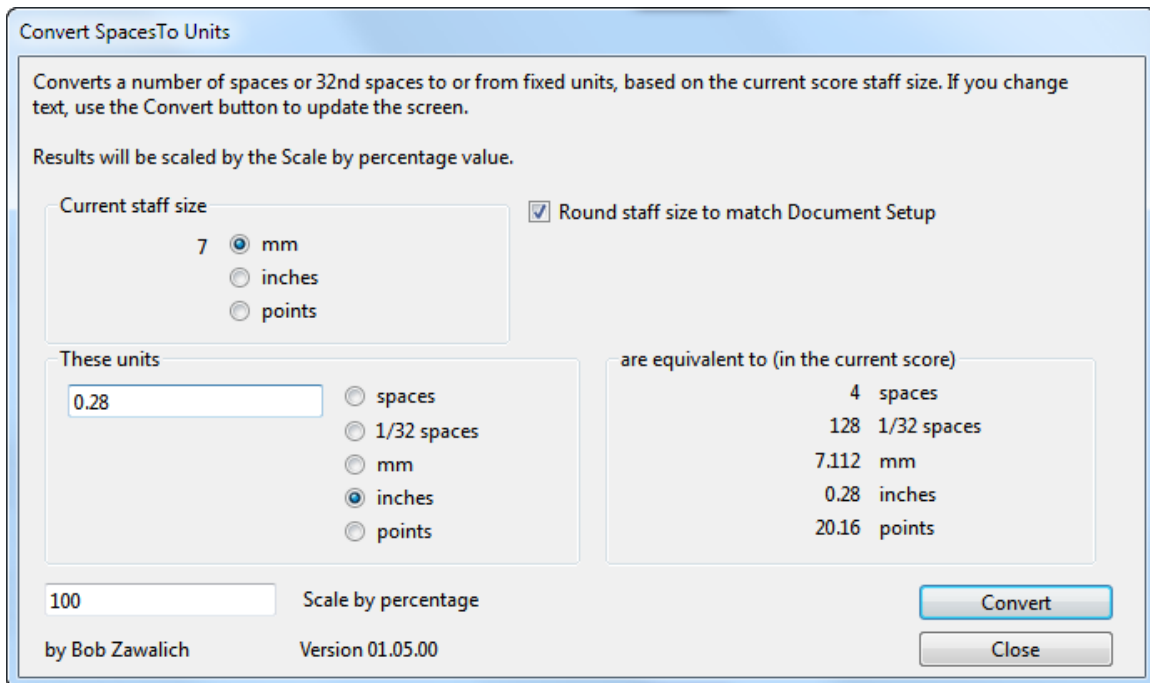
Or you could run the plugin and ask how many spaces in an inch.

In another obscure example, there is a wildcard modifier "s", defined as:

`\s123\` Size change to 123 (units are 1/32nds of a space, not points)

So fine, what if you want to change a font size to 12 points. Now let's see, how many 32nds of a space is that? Run the plugin and find out. Or maybe you prefer to find the staff size in points, divide by 4, divide again by 32 and then decide maybe it is time to go out for a grande nonfat latte...

So yeah it is obscure, but if you ever need it you will be glad it is around.



<http://www.sibeliusblog.com/tutorials/using-font-sizes-in-wildcards/>

Converting a font size in points to spaces

The `\sheight\` wildcard rather inconveniently specifies the font size in units of 32nds of a space, rather than in points, which is what Sibelius uses in the Ribbon. We want to figure out how to convert a font size in points to a size in 32nds of spaces.

A space is a “relative unit”, defined by Sibelius as 1/4 of the current staff size. So if you are using the common 7 mm staff size, a space will be 7/4 mm, or 1.75 mm. This is a relative unit, because if you change the staff size to 6 mm, a space will automatically convert to 1.5 mm; in a 10 mm staff, a space will be 10/4, or 2.5 mm.

As long as you don’t change the staff size, you can treat a space as a fixed unit, but it allows the score to scale appropriately when you change the staff size.

When you use the `\sheight\` wildcard, you are typically going to think of the font size in points, rather than spaces. If you want the font size to be 19.6 points, how do you convert that to spaces to use in the wildcard?

You can do this by hand, and I will describe the process below. I think it is much easier to do this using the downloadable plug-in [Convert Spaces To Units](#), which may be downloaded directly through Sibelius 7 or 7.5 at **File > Plug-ins > Install Plug-ins > Developers’ Tools**. Users may also install it manually in Sibelius 6, 7 or 7.5 by visiting

the [plug-in download page](#) and following the [usual manual installation procedure](#), or by using the [Install New Plug-in](#) plug-in.

When you run it, you will see this dialog. Notice that it knows the staff size for the current score, in this case 7 mm:

Convert SpacesTo Units

Converts a number of spaces or 32nd spaces to or from fixed units, based on the current score staff size. If you change text, use the Convert button to update the screen.

Results will be scaled by the Scale by percentage value.

Current staff size

7 ☒ mm
☐ inches
☐ points

☒ Round staff size to match Document Setup

These units

19.6 ☐ spaces
☐ 1/32 spaces
☐ mm
☐ inches
☒ points

are equivalent to (in the current score)

3.92 spaces
125.44 1/32 spaces
6.9144 mm
0.2722 inches
19.6 points

100 Scale by percentage

Convert

Close

by Bob Zawalich Version 01.05.00

We want to convert a point size (19.6, in this example) to the units used by the `\s\` wildcard, which are in 32nds of a space. Type **19.6** into the “These units” box, the click on the “points” radio button, and you will see the equivalent value in a number of units, and we can find that at this staff size, 19.6 points is equivalent to 125.44 32nds of a space.

So we can specify

`\s125.44\Text at 19.6 points,`

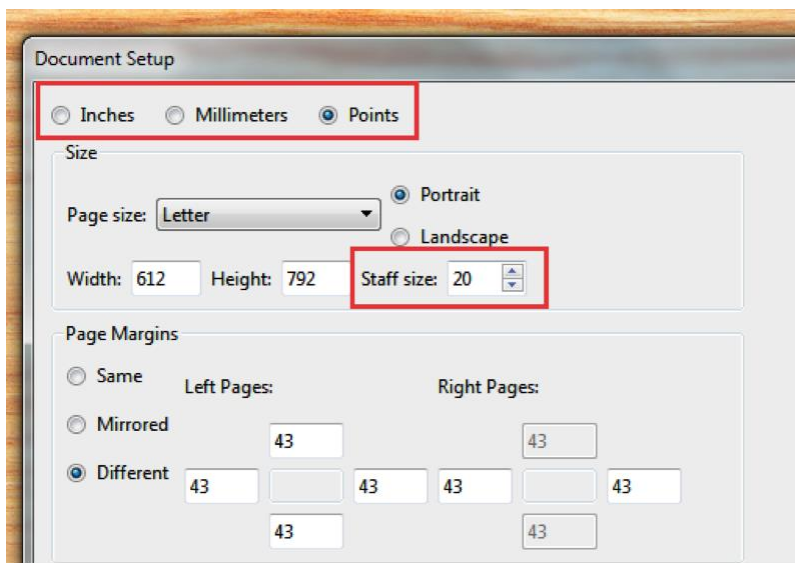
and “Text at 19.6 points” should appear at 19.6 points. When I try this in Windows 7 and Sibelius 7.5, Sibelius tells me the font size is 19.4 points. You might need to make some adjustments to work around some of Sibelius’ internal rounding, but we are at least in the ballpark.

When you type into the “These units” box, calculation does not happen until you click somewhere else. So after typing, always click on one of the radio buttons for units (even if it is not changing) so the plug-in will perform the calculation.

Converting points to spaces by hand

If you bring up the **Layout > Document Setup** dialog, you can change the units to points and read the staff size. The standard 7mm staff is equivalent to 20 points. So a space is 5 points ($20/4$), and 1 point is .2 spaces.

Remember that a point is .2 spaces only when the staff size is 7 mm = 20 points = .28 inches.



Since 1 point is .2 spaces, 19.6 points would be 19.6 times .2, which is 3.92 spaces. The wildcard specifies a number in 32nds of a space, so we need to multiply this by 32 to get 125.44, which is what the plug-in figured out! So you can certainly do the calculation manually, but it is a bit tedious.

Updated August 24, 2015 with correct information about resetting the font size to the default for the text style.

Staff Text Style

Name:

Based on:

Font Border Indents Horizontal Posn Vertical Posn Repeat

Basic Formats

☐ Font:

☒ Style:

☐ Underline

☒ Size in score: (rel. to 7mm staff) (absolute)

☐ Keep absolute

☒ Size in parts: (rel. to 7mm staff) (absolute)

☐ Keep absolute

Floating point

From Wikipedia, the free encyclopedia

This article is about the method of representing a number. For the album by John McLaughlin, see [Floating Point](#).



An early electromechanical programmable computer, the [Z3](#), included floating-point arithmetic (replica on display at [Deutsches Museum](#) in [Munich](#)).

In [computing](#), **floating point** is the formulaic representation that approximates a [real number](#) so as to support a [trade-off](#) between range and [precision](#). A number is, in general, represented approximately to a fixed number of [significant digits](#) (the [significand](#)) and scaled using an [exponent](#); the base for the scaling is normally two, ten, or sixteen. A number that can be represented exactly is of the following form:

$$\text{significand} \times \text{base}^{\text{exponent}},$$

where significand $\in \mathbb{Z}$, base $\in \mathbb{N}$, and exponent $\in \mathbb{Z}$.

For example:

$$1.2345 = \underbrace{12345}_{\text{significand}} \times \underbrace{10^{-4}}_{\text{base}}^{\text{exponent}}$$

The term *floating point* refers to the fact that a number's [radix point](#) (*decimal point*, or, more commonly in computers, *binary point*) can "float"; that is, it can be placed anywhere relative to the significant digits of the number. This position is indicated as the exponent component, and thus the floating-point representation can be thought of as a kind of [scientific notation](#).

A floating-point system can be used to represent, with a fixed number of digits, numbers of different [orders of magnitude](#): e.g. the distance between galaxies or the diameter of an atomic nucleus can be expressed with the same unit of length. The result of this dynamic range is that the numbers that can be represented are not uniformly spaced; the difference between two consecutive representable numbers grows with the chosen scale.^[1]

Over the years, a variety of floating-point representations have been used in computers. However, since the 1990s, the most commonly encountered representation is that defined by the [IEEE 754](#) Standard.

The speed of floating-point operations, commonly measured in terms of [FLOPS](#), is an important characteristic of a computer system, especially for applications that involve intensive mathematical calculations.