

The morse-simple package

Jim Hefferon

v 1.0, 2026-July-04

This package is for including small bits of morse code as strings, not sounds, in $\text{\LaTeX}$ documents. Here is a minimal working example.

```
\documentclass{article}
\usepackage{morse-simple}
\begin{document}
Morse for FN34 is \cwWord{FN34}.
\end{document}
```

and there is a sample morse code cheat card included in this package.

These are the most useful commands.

<i>Input</i>	<i>Output</i>
<code>\cwWord{Vermont}</code>	<code>----</code> <code>..</code> <code>----</code> <code>----</code> <code>----</code> <code>----</code> <code>----</code>
<code>\cwMessage{ur 599}</code>	<code>----</code> <code>----</code> <code>----</code> <code>----</code> <code>----</code> <code>----</code> <code>----</code>
<code>\morse{tu 73}</code>	<code>-</code> <code>----</code> <code>----</code> <code>----</code> <code>----</code> <code>----</code> <code>----</code>

If you like, you can just use the second one. It just calls the first after breaking the input along the spaces.

A note about the names: what you may know as ‘dots’ and ‘dashes’ are best called ‘dits’ and ‘dahs’. You can get a single dit or dah with `\cwDit` and `\cwDah`.

Package options

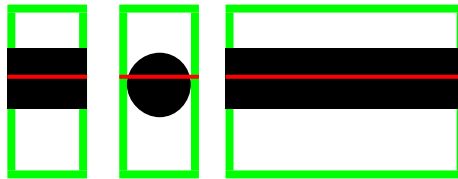
The main package option is that the default for dits is a short dash ‘`-`’ but you can instead choose a filled-in circle ‘`.`’ by calling the package as `\usepackage[ditdot]{morse-simple}`, giving `.-.-.` `.-` `.-.-.` `..` `...` for `\cwWord{Paris}`.

<i>Option</i>	<i>Type</i>	<i>Default</i>	<i>Affects</i>
<code>thickness=...</code>	Dimension	<code>0.35ex</code>	Thickness (vertical) of a dash.
<code>height=...</code>	Dimension	Same as math —	Distance of the middle of a dash above the baseline.
<code>ditwidth=...</code>	Dimension	<code>0.20em</code>	Width (horizontal) of a dit.
<code>dahwidth=...</code>	Dimension	<code>3 · ditwidth</code>	Width (horizontal) of a dah.
<code>intracharfactor=...</code>	Number	<code>1</code>	Multiply this by the ditwidth to get the space between dits and dahs inside of a character.
<code>intercharfactor=...</code>	Number	<code>3</code>	Multiply this by the ditwidth to get the space between characters.
<code>interwordfactor=...</code>	Number	<code>7</code>	Multiply this by the ditwidth to get the space between words.
<code>grayshade=...</code>	Number in <code>[0...1]</code>	<code>0.5</code>	Darkness of gray for <code>\morse{...}</code> (larger numbers are lighter).
<code>ditdot</code>	Toggle on	Off	Make dah’s be circles, not small dashes.
<code>dotscale=...</code>	Number	<code>1.5</code>	Factor by which periods are rescaled to make dot circles.
<code>dotraise=...</code>	Dimension	<code>1.70pt</code>	Height of rescaled dot.

Justifications for the defaults: the standard for sounds is that dahs are three times as long as dits. Similarly, the intracharacter space, that between a character's dits and dahs, is the same length as a dit, the intercharacter space is three times as long a dit, and the interword space is seven dits long. Visually, those all looked OK to me so I used them. (I have made these spacings be dimensions, not skips. In particular, I judged that no one would want to put several lines of morse where for the interword space they would wish to add some stretch or shrink. If I guessed wrong then sorry about that; instead you should use a ragged right environment or something similar.)

Most of the options are for fiddling with placements and sizes, perhaps to better suit your document's font. Because some of the option defaults involve font characteristics, you want to load this package after you do any font loading.

To help fiddle, there are commands to enlarge the dits and dahs. Here are the results of `\cwEnlargeDit` and `\cwEnlargeDah` (these take an optional magnification factor argument, with default 15), where the `ditdot` package option is not invoked. The middle one temporarily changes the dah to `\cwDitDot`, just to show what enlargement of that looks like. The green boxes run from the baseline to a height of 1 ex. The red line is at the height of a math minus sign. Here the dit and dah are at that height, which is the default.



This package

This package is not named 'morse' because there is already a package by that name.

The justification for a new package is that the existing one does things in a way that is different from how most $\TeX$ -ers work today: it uses a METAFONT font. Some more recent approaches propose `TikZ`, which is a big solution to a small problem. Instead this package uses rules (with circles for dits, if desired, that are scaled-up periods).

This is the first time I've done `expl3` programming. Bug reports welcome.