

# Package ‘devEMF’

June 30, 2020

**Type** Package

**Title** EMF Graphics Output Device

**Version** 3.8

**Date** 2020-06-29

**Author** Philip Johnson

**Maintainer** Philip Johnson <plfj@umd.edu>

**Depends** R (>= 2.10.1)

**SystemRequirements** Xft or zlib (only needed for platforms other than OSX and Windows)

**Description** Output graphics to EMF+/EMF.

**License** GPL-3

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2020-06-29 22:00:09 UTC

## R topics documented:

emf . . . . .	1
<b>Index</b>	<b>4</b>

---

emf	<i>Enhanced Metafile Graphics Device</i>
-----	--

---

## Description

‘emf’ starts the graphics device driver for producing enhanced metafile (emf+/emf) graphics, which can imported natively as vector graphics in both LibreOffice and Microsoft Office.

**Usage**

```
emf(file = "Rplot.emf", width = 7, height = 7,
    bg = "transparent", fg = "black", pointsize = 12,
    family = "Helvetica", coordDPI = 300, custom.lty=emfPlus,
    emfPlus=TRUE, emfPlusFont = FALSE, emfPlusRaster = FALSE)
```

**Arguments**

file	character string giving the name of file.
width	width of plot in inches.
height	height of plot in inches.
bg	plot background color. If "transparent", no background color is painted.
fg	initial foreground color to be used.
pointsize	default font point size to be used.
family	default font family to be used (note fonts cannot be embedded in EMF/EMF+ files, use non-standard fonts with caution).
coordDPI	DPI used for coordinate system (while EMF is a vector format, it (and R) use discrete coordinate systems).
custom.lty	logical: if false, approximate R line types with EMF standard line types (any program that imports EMF should support this). If true, use exact (custom) R line type, which is only supported by LibreOffice if using EMF+.
emfPlus	logical: whether to use EMF+ records or not
emfPlusFont	logical: if using EMF+, should text use EMF+ or EMF records?
emfPlusRaster	logical: if using EMF+, should raster images use EMF+ or EMF records?

**Details**

The standard office suites support very few vector graphics formats for import. Enhanced Metafiles (EMFs) do tend to be supported, which is the purpose of this device. EMF+ is a newer format that is encapsulated within an EMF file and allows options such as partial transparency.

The EMF/EMF+ specifications do not allow for multiple pages; attempting to create multiple pages will result in a warning message and the output file will contain the different pages superimposed.

Also, EMF/EMF+ do not allow for embedding fonts. Thus you must be careful that your desired font is present both on the system that you use when you create the emf file (i.e., when you run R) and on the system that you use to import/view the resulting file. Sticking to the standard Adobe PostScript font families is probably a good idea.

EMF/EMF+ supports Unicode characters, and this package tries to maintain support as well. However, font metric information is system dependent and so support may be spotty in practice. Basic font metrics for the standard Adobe PostScript font families are built into this package.

Only EMF+ allows partial transparency (i.e., the only useful type –  $0.0 < \alpha < 1.0$ ); attempting to use a transparent color when `emfPlus = FALSE` will result in a warning message and the output will be completely transparent (invisible).

LibreOffice support for EMF+ remains incomplete as of version 5, and, cannot handle EMF+ records with rotated text or raster images (hence the options to turn off EMF+ and use EMF instead for these types of records). devEMF defaults to EMF for these records to maintain compatibility, but quality is higher if EMF+ records are used.

### Known limitations

- EMF (as opposed to EMF+) raster rendering does not support interpolation control.
- EMF (as opposed to EMF+) does not support an alpha channel.
- Path rendering is not implemented for EMF (as opposed to EMF+).

### Author(s)

Philip Johnson

### See Also

[Devices](#)

### Examples

```
require(devEMF)
## Not run:
# open file "bar.emf" for graphics output
emf("bar.emf")
# produce the desired graph(s)
plot(1,1)
dev.off() #turn off device and finalize file

## End(Not run)
```

# Index

\* **device**

emf, [1](#)

devEMF (emf), [1](#)

Devices, [3](#)

emf, [1](#)