

NAME

luatex, dviluatex, lua_hbtex, lua_jithbtex, lua_jittex, texlua, texluac, texlua_jit, texlua_jitc – TeX extended with Lua as an embedded scripting language

SYNOPSIS

luatex [**--lua=luafile**] [*options*] [**&format**] [*file* [*more-input*] | [**\more-input**]

DESCRIPTION

Run the Lua_{TEX} typesetter on *file*.[*tex*], usually creating *file*.pdf. If the file argument has no extension, ".tex" will be appended to it. See **tex**(1) for details of command-line parsing, with the exceptions described here.

Lua_{TEX} began as an extended version of pdf_{TEX} with **lua**(1) as an embedded scripting language, Unicode and OpenType/TrueType font support, the e-_{TEX} and Omega extensions, as well as an integrated MetaPost library. It can generate either pdf or dvi output. For more information about Lua_{TEX}, see <https://www.luatex.org>; or you can read the Lua_{TEX} manual using the texdoc utility (**texdoc luatex**).

All Lua_{TEX} text input and output is read natively as Unicode text, although filters make it possible to support any encoding. Caveat: just because any Unicode character can be recognized as input does not mean it can be typeset as output; you still have to load the necessary fonts.

If the program is called as **texlua** it acts as a Lua interpreter. If called as **texluac** it acts as a Lua bytecode compiler.

By default, **luatex** produces pdf output, and **dviluatex** produces dvi output; both executables behave identically otherwise, within the limits of the output format. Both variants can be configured to produce either dvi or pdf output.

In pdf mode, Lua_{TEX} can natively handle the pdf, jpg, jbig2, and png graphics formats, but not PostScript or Encapsulated PostScript (eps) graphics; see **pdf_{tex}**(1). As a special case, the PostScript files output by MetaPost can be handled directly via the *supp-pdf.tex* and *supp-pdf.mkii* macros.

The regular **luatex** programs embed Lua 5.3, developed by PUC-Rio; the **j**it variants such as **lua_jittex** embed LuaJIT for just-in-time compilation, developed by Mike Pall.

The **lua_hbtex** variant includes the HarfBuzz engine for glyph shaping. HarfBuzz must still be explicitly enabled at the _{TEX} level.

By default, Lua_{TEX} does not provide any primitives beyond those from _{TEX}3. All extended primitives must be enabled at runtime using **\directlua**; see the Lua_{TEX} manual and existing format-building files, e.g., luatex.ini.

OPTIONS

For the common _{TEX} command-line options and handling, see **tex**(1).

When the Lua_{TEX} executable starts, it looks for the **--lua** command-line option. If there is no **--lua** option, the command line is interpreted similarly to traditional pdf_{TEX} and Aleph. But if the option is present, Lua_{TEX} will enter an alternative mode of command-line parsing, unrelated to the standard Web2C programs. The presence of **--lua** thus makes all other options unreliable, because the Lua initialization file can disable the default file searching via Kpathsea and/or hook functions into various callbacks.

--lua=luafile

The Lua initialization file. The Lua_{TEX} executables embed a default initialization file (luatex-core.lua); if this option is used, *luafile* will be used instead. The Lua initialization file is responsible for setting up security restrictions, path searching behaviour, and other low-level settings, so you must be careful when using a custom file or else you will likely crash the engine. If this option is used, you must pass the same file both when dumping the format and when typesetting a document. This option is used by Con_{TEX}t.

The following two options alter the executable behaviour:

--luaonly

Start Lua_{TEX} as a Lua interpreter. In this mode, it will set Lua's *arg[0]* to the found script name, pushing preceding options in negative values and the rest of the command line in the positive values, just like the Lua interpreter. Lua_{TEX} will exit immediately after executing the specified Lua script.

--luaconly

Start Lua_{TEX} as a Lua bytecode compiler. In this mode, Lua_{TEX} is exactly like **luac** from the standalone Lua distribution, except that it does not have the **-l** switch, and that it accepts and ignores the **--luaconly** switch.

--safer Disable most file-modifying and program-executing Lua functions. This option is disabled by default since it breaks most documents.

--luadebug

Enables the standard Lua *debug* module, which is disabled by default for security reasons. This option is implicitly enabled when unrestricted shell-escape is enabled.

--[no-]socket

Enables the Lua *socket* module, which allows _{TEX} documents to make arbitrary network requests, and is disabled by default for security reasons. This option is implicitly enabled when unrestricted shell escape is enabled.

ENVIRONMENT

See **tex**(1) and **pdf_{tex}**(1).

BUGS

See **tex**(1).

SEE ALSO

aleph(1), **et_{tex}**(1), **lua**(1), **m_{post}**(1), **pdf_{tex}**(1), **tex**(1).

Lua_{TEX} home page: <https://luatex.org>

Package page on CTAN: <https://ctan.org/pkg/luatex>

Lua_{TEX} reference manual: <https://mirror.ctan.org/systems/doc/luatex/luatex.pdf>

Web2c manual: <https://tug.org/web2c>

Kpathsea manual: <https://tug.org/kpathsea>

Numerous articles have been published in *TUGboat* about Lua_{TEX}:

<https://tug.org/TUGboat/Contents>

AUTHORS

{TEX} was created by Donald E. Knuth. The primary authors of the Lua{TEX} enhancements are Taco Hoekwater, Hartmut Henkel, Hans Hagen, and Luigi Scarso, with help from Martin

Schröder, Karel Skoupý, and Hàn The Thành.

Bug reports: dev-luatex@ntg.nl

Public discussion list for Lua_{TEX}-specific issues: <https://lists.tug.org/luatex>

Public discussion list for all things _{TEX}: <https://lists.tug.org/texhax>