The iftex package

The LaTeX Project Team*

2026/11/17 v1.0h TeX engine tests

Contents

1 Introduction **1**

2 Loading the package **2**
2.1 Loading the package in plain TeX 2
2.2 Loading the package in LaTeX 2
2.3 Loading the package in iniTeX 2

3 Engine test conditionals **2**

4 Requiring specific engines **4**

5 Output mode conditional **4**

6 Additional packages **5**

7 Compatibility with older copies of scrbase **5**

1 Introduction

This original iftex was written as part of the bidi collection (by the Persian TeX Group / Vafa Khalighi) and provided checks for whether a document was being processed with PDFTeX, or XeTeX, or LuaTeX. This version recodes the package and incorporates similar tests from the ifetex package by Martin Scharrer, the ifxetex package by Will Robertson, the ifluatex and ifvtex packages from Heiko Oberdiek and parts of ifptex by Takayuki Yato.

For each TeX variant engine supported two commands are provided:

- a conditional, `\iffootex` that is true if the (footex) engine (or a compatible extension) is being used.

*https://github.com/latex3/iftex

For compatibility with earlier packages which did not all use the same naming convention all these conditionals are provided in two forms, a lowercase name `\iffootex` and a mixed case name `\iffooTeX`.

- a command `RequireFooTeX` which checks that footex is being used, and stops the run with an error message if a different engine is detected.

2 Loading the package

The package can be loaded in the usual way in both Plain TEX and LATEX.

2.1 Loading the package in plain TEX

```
\input iftex.sty
```

2.2 Loading the package in LATEX

```
\usepackage{iftex}
```

2.3 Loading the package in iniTEX

The package assumes no existing macros and may be loaded during format setup in a format without the plain TEX or LATEX format being loaded. From an initial iniTEX setup the package may be loaded as for plain TEX.

3 Engine test conditionals

All the conditionals defined here are used in the same way:

```
\ifluatex
  luatex specific code
\else
  code for other engines
\fi
```

`\ifetex`, `\ifeTeX`
: True if an eTEX enabled format is in use. (This is necessarily true in all LATEX variants.)

`\ifpdftex`, `\ifPDFTeX`
: True if PDFTEX is in use (whether writing PDF or DVI), so this is true for documents processed with both the latex and pdflatex commands.

`\ifxetex`, `\ifXeTeX`
: True if XeTEX is in use.

`\ifluatex`, `\ifLuaTeX`
: True if LuaTeX and extensions such as LuaHBTeX or LuaMetaTeX are in use.

`\ifluahbtex`, `\ifLuaHBTeX`
: True if the luaharftex Lua module is available. This will be true in luahbtex and may be true in luatex if a binary Lua luaharftex module has been compiled and is available in Lua's search path.

`\ifluametatex`, `\ifLuaMetaTeX`
: True if LuaMetaTeX is in use.

`\ifptex`, `\ifpTeX`
: True if any of the pTeX variants are in use. (Note that the ptex command uses the uptex engine in current releases.)

`\ifuptex`, `\ifupTeX`
: True if any of the upTeX variants are in use. (`\ifetex` could be used in addition to distinguish uptex and euptex.)

`\ifptexng`, `\ifpTeXng`
: True if pTeX-ng (Asiatic pTeX) is in use.

`\ifvtex`, `\ifVTeX`
: True if VTeX is in use.

`\ifalephtex`, `\ifAlephTeX`
: True if Aleph is in use. (The aleph-based LaTeX command is lamed.)

`\iftutex`, `\ifTUTeX`
: This is not strictly an engine variant, but it is true if `\Umathchardef` is available, which essentially means that it is true for LuaTeX and XeTeX, allowing constructs such as

```
\iftutex
  \usepackage{fontspec}
  \setmainfont{TeX Gyre Termes}
  \usepackage{unicode-math}
  \setmathfont{Stix Two Math}
\else
  \usepackage{newtxtext,newtxmath}
\fi
```

`\iftexpadtex`, `\ifTexpadTeX`
: True if TexpadTeX is in use. Please note that TexpadTeX can run in two modes, one which uses Unicode and native fonts internally (similar to XeTeX and LuaTeX), and one which uses 8-bit codepages internally (similar to PDFTeX). This can be determined using `\iftutex`.

`\ifhint`, `\ifHINT`
: True if HiTEX (HINT output format) is in use.

`\ifprote`, `\ifProte`
: True if a Prote engine is in use. (This means it has TEX primitives, eTEX primitives, and the additional primitives required by LATEX.)

4 Requiring specific engines

For each supported engine, the package provides a command `\Require...` which checks that the document is being processed with a suitable engine, and stops with an error message if not.

`\RequireeTeX`

`\RequirePDFTeX`

`\RequireXeTeX`

`\RequireLuaTeX`

`\RequireLuaHBTeX`

`\RequireLuaMetaTeX`

`\RequirepTeX`

`\RequireupTeX`

`\RequirepTeXng`

`\RequireVTeX`

`\RequireAlephTeX`

`\RequireTUTeX`

`\RequireTexpadTeX`

`\RequireHINT`

`\RequireProte`

5 Output mode conditional

This package also provides an `\ifpdf` conditional that is true if the format is set up to output in PDF mode rather than DVI. This is equivalent to the test in the existing ifpdf package.

Unlike the engine tests above this is defined as if by `\newif` with user-documented commands `\pdftrue` and `\pdffalse` that can change the boolean

value. These would be needed to reset the boolean if the output mode is reset (for example by setting `\pdfoutput=0` in PDFLaTeX).

Unlike the original ifpdf package, the version here also detects PDF output mode if running in VTeX.

6 Additional packages

This extended iftex is designed to replace the original iftex and also the packages ifetex, ifluatex, ifvtex, ifxetex, ifpdf.

This collection includes small packages with these names that include the main iftex package, and in some cases define additional commands for increased compatibility. These packages should mean that authors do not need to change existing documents, although it is recommended that new documents use the iftex package directly.

Note that while this package provides basic support for detecting pTeX (Japanese TeX) variants and is broadly compatible with the ifptex package, the ifptex package has many more detailed tests for pTeX variants and this package does *not* replace the ifptex (or ifxptex) packages, which are maintained by their original authors and recommended for Japanese documents that need fine control over the Japanese TeX system in use.

7 Compatibility with older copies of scrbase

Prior to version 3.28 (released in 2019) the scrbase package (which is automatically included in the popular KOMA-Script classes) defined `\ifpdftex` and `\ifVTeX` with a different syntax unless the package option `internalonly` was used. This package detects if the old scrbase definitions are in effect and if so does not redefine them, for compatibility with existing documents.

Current versions of scrbase avoid this conflict.