Introduction

LuaTeX modifies the package loading behavior for Lua package, as mentioned in Section 3.2 of the LuaTeX manual:

LuaTeX is able to use the kpathsea library to find require()d modules. For this purpose, `package.searchers[2]` is replaced by a different loader function, that decides at runtime whether to use kpathsea or the built-in core Lua function. It uses kpathsea when that is already initialized at that point in time, otherwise it reverts to using the normal `package.path` loader.

What this means is that once kpathsea is initialized, the normal package loading behavior is no longer available.

This package allows Lua package loading behavior inside LuaTeX to use both kpathsea and the default loading mechanism. If the module is not available in kpathsea, LuaTeX will try to load it from `package.path`. This functionality is very useful, for instance when you need to use LuaRocks modules inside LuaTeX.

1 Installation

Install this package like any other TeX or LaTeX package.

2 Dependencies

This package depends on `ifluatex` CTAN package.

3 Usage

3.1 Plain TeX

In Plain TeX, you can import the module as follows:

\input luapackageloader.sty

3.2 LaTeX

In LaTeX, add the `luapackageloader` packages to your document:

\usepackage{luapackageloader}
3.3 Loading Packages

Once you have imported `luapackageloader`, it adds a table called `luapackageloader` under the global namespace, and automatically overrides the default LuaTeX package searchers. It should be possible to load a Lua package (including LuaRocks modules, as long as `package.path` and `package.cpath` is setup properly in your environment) as follows:

\input{luapackageloader.sty}
\directlua{
local serpent = require("serpent") % luarocks install serpent
}

3.4 Reverting to default loading behavior

The package loading can be reverted back to the default LuaTeX behavior using the `restore_kpse_searchers()`

\directlua{
luapackageloader.resetore_kpse_searchers()
}

3.5 Manually setting up to load Lua packages

This package exposes the function `add_lua_searchers()`, to manually setup the package searchers again, if it was reverted for some reason, after loading the package.

\input{luapackageloader.sty}
\directlua{
% Lets revert the package searchers to their default loading behavior.
luapackageloader.restore_kpse_searchers()
% Some code here...
% Now lets setup the package searchers to use Lua searchers again.
luapackageloader.add_lua_searchers()
}