Installation Diagnostics with `diagnose.sty` version 0.2

Oliver Corff
`corff@zedat.fu-berlin.de`

April 10th, 2001

Contents

1 Introduction

Large LaTeX 2_ε packages with extensive functionality are ideally written with the support offered by the LaTeX 2_ε installation in mind as it does not make sense to reinvent the wheel or copy code from other packages. If a package makes use of other packages in the system there should be a means to test for the presence of the required package, or a required token.

Depending on the result of this test, several situations are possible. Either the calling package fails completely, or fails to provide the intended service in its full scope, or has to check for some alternative. In any case, an informative message should be written into the log file and perhaps also into the text document.

TeX and LaTeX 2_ε provide basic functionality for testing the existence of files and tokens; this package goes a step further and provides mechanisms for informing the software and the user about a detected condition.

Besides providing a simple mechanism for repeated references to the existence of a file or token, the other important application is for checking whether a complex application can be used immediately after installing or whether it requires additional software to be installed. For this purpose, LaTeX packages can now be accompanied by a simple text document which, when compiled, checks the user's environment and makes suggestions about missing software, if necessary. An example of this idea is given in the file `mls-diag.tex`.

2 The `diagnose` Package

The `diagnose` package is included with the usual

```
\usepackage{diagnose}
```

statement. There are no options at present. The four commands offered are:

1. `\PackageDiagnostics`
2. `\TokenDiagnostics`
3. `\TextMessageYes`
4. `\TextMessageNo`

2.1 `\PackageDiagnostics`

```
\PackageDiagnostics{} % Tested package
                   {} % Associated counter
                   {} % message if file available
                   {} % message if file not available
```

This command verifies whether a file is available within the TEX directory structure. It is based on the LATEX 2ε command `\IfFileExists{}` and offers the possibility to create a counter which is associated with the existence of the requested file. The counter can then be used throughout the software without additional calls of `\IfFileExists{}`. Two message texts can be given; one will be printed if the file exists, the other will be printed if the file is absent.

By default the message texts appear in the log file and the terminal output of the running caller program; additionally it is possible to have the messages appear in the output text of the document.

2.2 `\TokenDiagnostics`

```
\TokenDiagnostics{} % Tested token
                 {} % Associated counter
                 {} % message if token available
                 {} % message if token not available
```

This command verifies whether a token (a command sequence, etc.) is known to the system. Like `\PackageDiagnostics` it accepts an associated counter and two message texts as arguments.

2.3 `\TextMessageYes`

By default, messages generated by the two diagnostics commands appear only in terminal output and log files. If the messages shall also be displayed in the output text then the command

```
\TextMessageYes
```

is to be used. It is not followed by any argument. By default, message texts do not appear in the output text, only in log files and terminal output.

2.4 `\TextMessageNo`

If output of diagnostics messages to the main text had been enabled before, it can be disabled with the command

```
\TextMessageNo
```

which happens to be the default of the diagnostics package.

3 Examples

Two examples are given here. First the usage of the package diagnostics command is shown, and, in a separate file, the reader is invited to compile the accompanying document `mls-diag.tex` which shows a real-life application of `diagnose.sty`.

```
\PackageDiagnostics{rotating.sty} % Tested package
 {RotatingAvailable} % Associated counter
 {Vertical text possible!} % message if file available
 {rotating.sty not found, no vertical text!} % message if not

\ifnum\theRotatingAvailable=1
        You can use vertical text in your documents. No further
        measure is required.
\else
        Without the \texttt{rotating.sty} package vertical text
        cannot be typeset. If you need this option, then access
        CTAN and install \texttt{rotating.sty} on your system.
        Make sure you can process PostScript output.
\fi
```

The actual output of this command sequence will depend on the presence of `rotating.sty` on the system where this source file is compiled:

> You can use vertical text in your documents. No further measure is required.

You can also see that the log file and the terminal output contain the appropriate message.

4 Legal Status

This software is placed under the GPL. Usual disclaimers apply.

As this software tool is still under development, suggestions on how to improve the usability and versatility of this package are highly appreciated.