The **embedall** package*

Alberto Sartori – `alberto.sartori.as@gmail.com`
David Lichti – `dlichtistw@gmx.de`

May 18, 2019

1 Introduction

This package is designed to help you storing your projects without losing anything. It uses the `filehook` package and some custom patches to hook into several supported import mechanisms. These imported files are then attached to the PDF file using the `embedfile` package. In particular it can embed images, external TeX files, external code listings, CSV files, and, most importantly, the main TeX file itself.

2 Usage

To use the package, simply add `\usepackage{embedsources}` to your document's preamble. Without any option, the default behaviour is to attach all supported source file types to the finished document. See section 2.1 for options to change this behaviour. No furhter macros are needed, and no change has to be made to the rest of the source.

As of now, this package can hook into `\input` and `\include` using the `filehook` package. Furthermore, two hooks for `graphicx` and `csvsimple` are provided.

2.1 Package Options

If the package is loaded without any option, then all hooks are installed. In case this is not wanted, each hook can be enabled individually.

---

*This document corresponds to embedall v2.0, dated 2019/05/18.

main	Attach the main source file. This uses the `\AtBeginDocument` hook.
input	Attach files loaded with `\input`. This is done using `\AtBeginOfInputs` from the `filehook` package. It also applies to every macro relying on `\input` to load external content.
include	Attach files loaded with `\include`. This is done using the `\AtBeginOfIncludes` hook.
graphicx	Attach image files used with `\includegraphics`. This patches some macro in the `graphicx` package, so make sure that it is loaded prior to `embedsources`.
csvsimple	Attach data files used with the `csvsimple` package. Make sure that it is already loaded.

Specifying any of these options will disable all hooks that are not enabled explicitly.

There are some other options for very special cases.

all	Enable all file type hooks. This is the default and the option is mostly redundant. However, it may be used to reenable all hooks, after an explicit type option has disabled them.
notall	Do not enable all file type hooks. As for the the `all` option, this is mostly useless, since it disables all effects this package was intended for. However, some side effects may remain. Consider not loading the package at all.
compat	Enable compatibility mode to maintain some functionality from version 1.0. Use this if your document needs the old style `\embedinput` macro.

2.2 User Commands

`\embedfile[`⟨*options*⟩`]`
`{`⟨*filename*⟩`}`

Although not defined by this package, it is made available by loading `embedfile`. You may use it to manually embed sources or any file, that is not covered by the hooks described above. See the `embedfile` documentation for more details.

`\embedsource[`⟨*options*⟩`]`

This macro can be used to embed the current TeX file in case the automatic hook has been disabled. Use it inside the file to be loaded. The ⟨*options*⟩ are passed to the `\embedfile` macro.

`\embedinput[`⟨*options*⟩`]`
`{`⟨*filename*⟩`}`

This command is a substitute for `\input{<filename>}` in compatibility mode. It adds the feature of attaching the file to the PDF after inserting it in the TeX source. The ⟨*options*⟩ are ultimately passed to the `\embedfile` macro. As of version 2.0

with the automatic input file hook enabled, this is not needed anymore. Consider using `\embedsource` for manual embedding without the automatic hook.

3 Compatibility

This package heavily relies on `filehook`'s[1] file hooks and `embedfile`'s[2] file embedding. See the compatibility notes in their respective documentations.

Furthermore, hooking into the `graphicx` and `csvsimple` macros is done by patching some of their macros. If they have not been loaded when `embedall` is loaded, this will fail. If they are loaded or reloaded after `embedall`, these patches may be overwritten. To ensure good functionality, load this package after all packages providing import commands.

4 Implementation

4.1 Options

First, we need some booleans to store package options. The switch `embedall@all` acts on all types of hooks.

```
\newif\ifembedall@all
\embedall@alltrue
```

The following `embedall@<hook>` can be used to enable specific embedding hooks.

```
\newif\ifembedall@main
\embedall@mainfalse
\newif\ifembedall@input
\embedall@inputfalse
\newif\ifembedall@include
\embedall@includefalse
\newif\ifembedall@graphicx
\embedall@graphicxfalse
\newif\ifembedall@csvsimple
\embedall@csvsimplefalse
```

Compatibility mode is disabled by default.

---

[1] filehook package: `https://ctan.org/pkg/filehook`

[2] embedfile package: `https://ctan.org/pkg/embedfile`

```
\newif\ifembedall@compat
\embedall@compatfalse
```

Now, declare the actual package options.

```
\DeclareOption{all}{\embedall@alltrue}
\DeclareOption{notall}{\embedall@allfalse}
```

The following options correspond to the more specific embedding hooks mentioned above. Using any of these options will turn of general embedding off all source file types.

```
\DeclareOption{main}{\embedall@maintrue\embedall@allfalse}
\DeclareOption{input}{\embedall@inputtrue\embedall@allfalse}
\DeclareOption{include}{\embedall@includetrue\embedall@allfalse}
\DeclareOption{graphicx}{\embedall@graphicxtrue\embedall@allfalse}
\DeclareOption{csvsimple}{\embedall@csvsimpletrue\embedall@allfalse}
```

Request compatibility mode.

```
\DeclareOption{compat}{\embedall@compattrue}
```

Now, process these options.

```
\ProcessOptions\relax
```

Turn on all `embedall@<hook>` if `embedall@all` is `true`.

```
\ifembedall@all
    \embedall@maintrue
    \embedall@inputtrue
    \embedall@includetrue
    \embedall@graphicxtrue
    \embedall@csvsimpletrue
\fi
```

4.2 Dependencies

Load packages for file embedding and file hooks.

```
\RequirePackage{embedfile}
\RequirePackage{filehook}
\RequirePackage{currfile}
\RequirePackage{etoolbox}
\ifembedall@compat
    \RequirePackage{letltxmacro}
\fi
```

4.3 Macros

`\embedsource` We define a shorthand to embed the current source file.

```
\newcommand\embedsource[1][]{\embedall@embed[#1]{\currfilename}}
```

`\embedall@filelist` Keep a list of files that have already been attached.

```
\def\embedall@filelist{}
```

`\embedall@embed` This is done to avoid embedding the same file multiple times.

```
\newcommand{\embedall@embed}[2][]{%
    \xifinlist{#2}{\embedall@filelist}{}{%
        \newlinechar=`\^^J%
        \message{^^J^^Jembedall: Attaching file '#2'.^^J^^J}%
        \embedfile[#1]{#2}%
        \listxadd{\embedall@filelist}{#2}%
    }%
}
```

4.4 Compatibility Commands

The following macros are only defined in compatibility mode to maintain backward compatibility.

```
\ifembedall@compat
```

`\embedall@includegraphics` This macro is the modified version of `\includegraphics` command from the `graphicx` package with included the `\embedfile` command. First the original `\includegraphics` command is stored inside the not-user-accessible `\embedall@latex@includegraphics`.

```
    \LetLtxMacro\embedall@latex@includegraphics\includegraphics
```

`\LetLtxMacro` is used (instead of simply `\let`) because some packages modify the `\includegraphics` command. For this reason you can not use the * variant of this command but you have to declare the `clip` option instead.

Then the `\embedall@includegraphics` is defined

```
    \newcommand\embedall@includegraphics[2][]{%
        \embedall@embed[desc=image]{#2}
        \embedall@latex@includegraphics[#1]{#2}%
    }
```

`\embedall@listinputlisting` This macro is the modified version of `\listinputlisting` command from the `listings` package with included the `\embedfile` command. First the original `\listinputlisting` command is stored inside the not-user-accessible `\embedall@latex@listinputlisting`.

```
    \LetLtxMacro\embedall@latex@lstinputlisting\lstinputlisting
```

Then the `\embedall@listinputlisting` is defined

```
    \newcommand\embedall@lstinputlisting[2][]{%
        \embedfile[desc=listing file]{#2}
        \embedall@latex@lstinputlisting[#1]{#2}%
    }
```

`\embedinput` This command can be used as a substitute for `\input` to manually embed the source if the automatic file hook is not enabled. Otherwise, it will just call `\input` without further side effects. The additional optional parameter sets the input file description in the generated PDF.

```
    \ifembedall@input
        \DeclareRobustCommand\embedinput[2][]{%
            \input{#2}%
        }
    \else
        \DeclareRobustCommand\embedinput[2][input file]{%
            \input{#2}%
            \embedall@embed[desc={#1}]{#2}%
        }
    \fi
\fi
```

4.5 Installing Hooks

The actual work happens here. First, the main source file is attached.

```
\ifembedall@main
    \ifembedall@compat
        \IfFileExists{./\jobname.tex}{%
            \embedall@embed[desc=main source file]{\jobname.tex}
        }{}
    \else
        \AtBeginDocument{\embedsource[desc=main source file]}
    \fi
\fi
```

These macros are provided by the `embedfile` and `currfile` packages.

Then, the hooks for `\input` and `\include` are installed, given the respective switch was enabled.

```
\ifembedall@input
    \AtBeginOfInputs{\embedsource[desc=input file]}
\fi
\ifembedall@include
    \AtBeginOfIncludes{\embedsource[desc=include file]}
\fi
```

The macros `\AtBeginOfInputs` and `\AtBeginOfIncludes` are provided by the `filehook` package.

For the `graphicx` hook, we need to patch the `\Gin@setfile` macro to call `\embedfile` after finishing its own work. At this point, the full file name including the file name extension will be available as the third argument passed to the macro.

```
\ifembedall@graphicx
    \ifcsname Gin@setfile\endcsname
        \let\embedall@Gin@setfile\Gin@setfile
        \def\Gin@setfile#1#2#3{%
            \embedall@Gin@setfile{#1}{#2}{#3}%
            \embedall@embed[image file]{#3}%
        }
```

Issue a warning if `\Gin@setfile` does not exists. This could happen in the unlikely case that the internal workings of `graphicx` have changed, or, more likely, because the package was not loaded.

```
    \else
        \ifembedall@all\else
            \PackageWarning{embedall}{Patching of graphicx requested, but package not loaded. C
        \fi
    \fi
\fi
```

The `csvsimple` hook is very similar to the `graphicx` hook. It patches into the `\csvloop` macro which is used by all other user macros to do the actual work.

```
\ifembedall@csvsimple
    \ifcsname csvloop\endcsname
        \let\embedall@csvloop\csvloop
        \def\csvloop#1{%
            \embedall@csvloop{#1}%
            \embedall@embed[CSV file]{\csv@input@filename}%
        }
    \else
        \ifembedall@all\else
            \PackageWarning{embedall}{Patching of csvsimple requested explicitly, but package n
        \fi
    \fi
\fi
```

The `\csv@input@filename` macro is used in the `csvsimple` macro to store the input file name.