The **tikzinclude** package[*]

Ferdinand Schwenk `me@nerdifand.de`
Benjamin Berg `benjamin@sipsolutions.net`

January 1, 2013

Abstract

This package addresses the problem of importing only one TikZ-image from a file holding multiple images (i.e. different versions of the same picture).

1 Introduction

Normaly I use one file per TikZ-image. This simplifies reusage of the images in different documents.

When drawing different versions of the same image, for example to highlight parts of the image or provide localized versions, it is not practical to put each of the version in a separate file. Doing this would increase the risk of version mismatch.

Because of this it can make sense to have all versions of one image or even different images inside the same source file. However, simply doing this makes it impossible to use the `\input` command as this would insert all images at the same time. This package solves the issue by allowing the user to only insert a single tikzpicture from a file.

2 Usage

To be able to select an image it necessary to name each drawing. This is done by assigning a figure name to the TikZ-Key `/tikzinclude/figure` at the beginning of the picture.

```
\begin{tikzpicture}[/tikzinclude/figure=foo]
  \node{foo};
\end{tikzpicture}
```

`\includetikzgraphics` After naming the images it is now possible to only include a specific image using the `\includetikzgraphics[`⟨*name*⟩`]{`⟨*imagefile*⟩`}` command.
⟨*name*⟩ is the name of the image that should be included. If ⟨*name*⟩ is provided then only the picture with the given name is included, all other pictures are dropped. If ⟨*name*⟩ is omitted all pictures in ⟨*imagefile*⟩ are included. This gives the same result as if `\input` was used.

---

[*] This document corresponds to **tikzinclude** v1.0, dated 2012/22/02.

3 Implementation

The Package is depending on `ifthen` and `etoolbox`

```
\RequirePackage{tikz}
\RequirePackage{ifthen}
\RequirePackage{etoolbox}
```

Provide a new if condition that states if tikzinclude should be active or if all pictures should be included.

```
\newif\if@tikzinclude@active\@tikzinclude@activefalse
```

Store the original definition of `\pgfsys@typesetpicturebox`

```
\let\@tikzinclude@typsetpicturebox\pgfsys@typesetpicturebox
```

`\@tikzinclude@picture@started` This internal macro sets `\par` to its original definition, if tikzinclude is used. Otherwise it should not have been changed.

```
\newcommand{\@tikzinclude@picture@started}{%
  \if@tikzinclude@active%
    \let\par\@tikzinclude@par%
  \fi%
}
```

`\@tikzinclude@picture@ended` This internal macro overwrites `\pgfsys@typesetpicturebox` with a macro that drops the next box entirely and then resets the definition of `\pgfsys@typesetpicturebox`. The overwrite only happens if the image should be suppressed.

```
\newcommand{\@tikzinclude@picture@ended}{%
  \if@tikzinclude@active%
    \ifthenelse{%
      \equal{\pgfkeysvalueof{/tikzinclude/figure}}{\pgfkeysvalueof{/tikzinclude/select}}%
    }%
    {}%
    {%
      \global\def\pgfsys@typesetpicturebox##1{%
        \global\let\pgfsys@typesetpicturebox\@tikzinclude@typsetpicturebox%
      }%
    }%
  \fi%
}
```

Some hooks need to be installed.

```
\BeforeBeginEnvironment{tikzpicture}{\if@tikzinclude@active%
  \whileboolexpr{test{\ifdimgreater{\lastskip}{0pt}}}{\unskip}\fi}%
  \AtBeginEnvironment{tikzpicture}{\@tikzinclude@picture@started%
}
\AtEndEnvironment{tikzpicture}{\@tikzinclude@picture@ended}
\AfterEndEnvironment{tikzpicture}{\ignorespaces}
```

Set the TikZ-Keys to empty values. This is necessary to suppress some TikZ-warnings

```
\pgfkeyssetvalue{/tikzinclude/figure}{}
\pgfkeyssetvalue{/tikzinclude/select}{}
```

`\includetikzgraphics` First it is checked if ⟨*name*⟩ is provided or not. If no name is given a simple `\input` is performed.

If ⟨*name*⟩ is given it needs to be assigned to `/tikzinclude/select`. Any whitespace in the image file needs to be ignored, but whitespace inside the images should be unchanged. Therefore the definition of `\par` is stored and `\par` is overridden outside of any TikZ-environment.

Then tikzinclude is activated and the image file is included using the `\input` command. After the picture is included some additional whitespace needs to be erased. To have a defined starting position and to avoid the deletion of whitespace added by the user a 0pt kerning is inserted prior to the inclusion of the image file. The whitespace that needs to be removed is caused by newlines at the end of TikZ-environments.

In the end all settings are restored.

```
\newcommand\includetikzgraphics[2][]{%
  \begingroup%
    \ifthenelse{\equal{#1}{}}%
    {%
      \input{#2}%
    }%
    {%
      \let\@tikzinclude@par\par%
      \def\par{}%
      \pgfkeyssetvalue{/tikzinclude/select}{#1}%
      \@tikzinclude@activetrue%
      \kern0pt\input{#2}%
      \whileboolexpr{test{\ifdimgreater{\lastskip}{0pt}}}{\unskip}%
      \@tikzinclude@activefalse%
      \let\par\@tikzinclude@par%
    }%
  \endgroup%
}
```