

The T_EXPower bundle

Frequently asked questions list^a

Stephan Lehmke

`mailto:Stephan.Lehmke@cs.uni-dortmund.de`

July 3, 2000

^aFAQ v0.0.4 of Jun 22, 2000 for T_EXPower v0.0.8f of Jun 27, 2000 (pre-alpha).

Contents

1	General	5
1.1	Where can I get the newest version of the T _E XPower FAQ ?	5
1.2	What is T _E XPower ?	6
1.3	Where can I obtain T _E XPower ?	7
1.4	Where can I discuss T _E XPower or ask for help ? . .	8
1.5	What alternatives are there to using T _E XPower ? . .	9
2	Usage	10
2.1	How do I design a presentation with T _E XPower ? . .	10
2.2	I find T _E XPower very complicated. How can I learn how to realize dynamic effects ?	11
2.3	Can I combine T _E XPower with PPower4 ?	12

2.4	I'm missing some of the classes and packages used in the demo and example files.	13
3	How do I. . .	17
3.1	How can I incrementally display a paragraph of text?	17
3.2	Instead of making text appear 'out of nowhere', I'd rather just change colors from 'dimmed' to normal.	21
3.3	<code>\dstep</code> and <code>\hidedimmed</code> work only with <code>texpower</code> 's standard colors. How can I dim my own colors? . .	24
4	Problems	26
4.1	I'm loading the <code>texpower</code> package, but dynamic features don't seem to work.	26
4.2	When I use the <code>colormath</code> option, my displayed formulae are not colored.	27

4.3	It seems I can't use <code>\vfill</code> in combination with <code>\pause</code>	28
4.4	When I use <code>L^AT_EX+dvips+distiller</code> , the result looks strange and 'hyper' features don't work. . . .	29
4.5	When using <code>\highlighttext</code> or <code>\hidetext</code> , I'm getting strange error messages.	30
4.6	Inside the argument of <code>\stepwise</code> , all counters seem to be freezed on all pages of the sequence generated. How can I use a self-defined counter which does not freeze?	32

1 General

1.1 Where can I get the newest version of the T_EXPower FAQ ?

You can download the latest version of the T_EXPower FAQ from the following URLs:

<http://Ls1-www.cs.uni-dortmund.de/~lehmke/texpower/doc/FAQ-display.pdf> Screen version

<http://Ls1-www.cs.uni-dortmund.de/~lehmke/texpower/doc/FAQ-printout.pdf> Printout version

1.2 What is T_EXPower ?

The T_EXPower bundle contains style and class files for creating dynamic online presentations with L^AT_EX.

The heart of the bundle is the package `texpower.sty` which implements some commands for presentation effects. This includes setting page transitions, color highlighting and displaying pages incrementally.

The document class `powersem.cls` is a wrapper for seminar which sets up everything for dynamic presentations.

1.3 Where can I obtain T_EXPower ?

The complete bundle, together with its documentation, can be found under the URL

<http://Ls1-www.cs.uni-dortmund.de/~lehmke/texpower>

For the current pre-alpha release, the license forbids any form of redistribution, so for the time being, if you want a new version of T_EXPower, you have to download it from there.

1.4 Where can I discuss T_EXPower or ask for help ?

Bug and problem reports should go to the author **Stephan Lehmke** by email.

Discussions about T_EXPower should take place on the mailing list **texpower@ls6.cs.uni-dortmund.de**. To subscribe, send an email message to the address **texpower-request@ls6.cs.uni-dortmund.de** with the subject **subscribe**.

The mailing list is publicly archived at

http:
//www.mail-archive.com/texpower@ls6.cs.uni-dortmund.de/

Until the first alpha release, I'd like to keep T_EXPower out of the 'general' T_EX mailing lists and newsgroups as much as possible, to avoid drawing too much premature attention.

1.5 What alternatives are there to using T_EXPower ?

The most prominent alternative to T_EXPower is the Pdf Presentation Post Processor PPower4, the homepage of which is

[http://www-sp.iti.informatik.tu-darmstadt.de/
software/ppower4/](http://www-sp.iti.informatik.tu-darmstadt.de/software/ppower4/)

Another alternative is the Utopia PDF Presentations Bundle, which provides a complete presentation design environment. Its home page is

<http://www.utopiatype.com.au/products/ubundle.html>

Comparisons of different presentation packages can be found on the home page of Prof. D. P. Story:

http://www.math.uakron.edu/~dpstory/pdf_demos.html

and in the talk held by Ross Moore at the California Institute of Technology on 8th May 2000:

<http://www.cds.caltech.edu/caltex/2000/>

2 Usage

2.1 How do I design a presentation with T_EXPower ?

It should be stressed that T_EXPower is **not** (currently) a complete presentation package. It just adds dynamic presentation effects (and some other gimmicks specifically interesting for dynamic presentations) and should always be combined with a document class dedicated to designing presentations (or a package like `pdfslide`).

There are demos in the `doc` directory for most popular presentation-making document classes and packages.

2.2 I find T_EXPower very complicated. How can I learn how to realize dynamic effects ?

As always with T_EX, you should first make up your mind what kind of effect you desire, and what L^AT_EX structures will be involved.

Then you should check the examples in the `doc` directory for anything similar to what you want. If you find anything suitable, read the corresponding code. There are some inline comments to explain what's going on. Print out the `manual` for documentation of the T_EXPower commands.

Further 'recepies' can be found in section 3.

If you don't find anything suitable you can modify to your needs, and can't figure out from the documentation how to achieve your aims, please report to [Stephan Lehmke](#) by email. If you've found an application for T_EXPower not covered by the examples, a new example should be created.

2.3 Can I combine T_EXPower with PPower4 ?

There is no problem postprocessing documents in which T_EXPower is used. This can be useful, for instance, for realising structured backgrounds with the `background` package from the PPower4 bundle.

If there are presentation effects for which you'd like to use PPower4's implementation of the `\pause` command, then just load PPower4's `pause` package. PPower4's definition of `\pause` will override `texpower`'s. Then you can combine PPower4's `\pause` functionality with T_EXPower's `\stepwise` functionality, for maximum expressive power.

2.4 I'm missing some of the classes and packages used in the demo and example files.

First of all, it has to be said that T_EXPower makes use of some ‘modern’ features which have been introduced into the T_EX System quite recently and are evolving swiftly. The core of the `texpower` package, namely the commands `\pause` and `\stepwise` is implemented in ‘pure’ L^AT_EX and should be largely independent of any fancy extensions, but to get most out of T_EXPower’s presentation features and process the more advanced examples, it is recommended to have a moderately new T_EX distribution installed (rule of thumb: not older than one year).

But even if your distribution is quite new, it might not contain some of the classes and packages used by the demos and examples. Here’s a list of (hopefully all of) the packages and classes used (which are not part of core L^AT_EX) and their availability:

Package	used in	available from
hyperref	most	CTAN, e.g. ftp://ftp.dante.de/tex-archive/macros/latex/contrib/supported/hyperref
url	most	CTAN, e.g. ftp://ftp.dante.de/tex-archive/macros/latex/contrib/other/misc
soul	many	CTAN, e.g. ftp://ftp.dante.de/tex-archive/macros/latex/contrib/supported/soul
pstricks	fulldemo, picexample	CTAN, e.g. ftp://ftp.dante.de/tex-archive/graphics/pstricks
xr-hyper	manual	CTAN, e.g. ftp://ftp.dante.de/tex-archive/macros/latex/contrib/supported/hyperref
fancyvrb	FAQ	CTAN, e.g. ftp://ftp.dante.de/tex-archive/macros/latex/contrib/supported/fancyvrb
pdfscreen	pdfscrdemo	CTAN, e.g. ftp://ftp.dante.de/tex-archive/macros/latex/contrib/supported/pdfscreen
pdfslide	pdfslidemo	CTAN, e.g. ftp://ftp.dante.de/tex-archive/macros/latex/contrib/supported/pdfslide

Package	used in	available from
pp4slide	pp4sldemo	CTAN, e.g. ftp://ftp.dante.de/tex-archive/support/ppower4/pp4sty.zip
ifmslide	ifmslideo	CTAN, e.g. ftp://ftp.dante.de/tex-archive/macros/latex/contrib/supported/ifmslide

Class	used in	available from
seminar	most	CTAN, e.g. ftp://ftp.dante.de/tex-archive/macros/latex/contrib/other/seminar
scrartcl	most	CTAN, e.g. ftp://ftp.dante.de/tex-archive/macros/latex/contrib/supported/koma-script
foils	foilsdemo, pp4sldemo	CTAN, e.g. ftp://ftp.dante.de/tex-archive/nonfree/macros/latex/contrib/supported/foiltex

The T_EXPower bundle is in some sense ‘intertwined’ with some packages and files which develop very fast at the moment, because it builds on features provided by those. Consequently, the development of T_EXPower has to be ‘synchronised’ with the development of these packages. To obtain the best results with T_EXPower, it might be advisable to download the newest version of the following files from their home page (which is likely to be newer than the newest version on CTAN):

File/Package	available from
--------------	----------------

<code>hyperref</code>	<code>http:</code> <code>//www.tug.org/applications/hyperref/hyperref.zip</code>
-----------------------	---

<code>pdftex.def</code>	<code>http://www.tug.org/applications/pdftex/pdftex.def</code>
-------------------------	--

3 How do I...

3.1 How can I incrementally display a paragraph of text ?

The easiest solution is to use `\parstepwise`, but if the arguments of `\step` are long, you'll get problems with line breaks, as `\parstepwise` forces `\step` to put its argument in a box.

You can use `\hidetext` like this:

```
1 \stepwise[\let\hidestepcontents=\hidetext]
2 {\step{Line breaks} \step{work in here.}}
```

yields



But note that `\hidetext`, being implemented using the `soul` package, is quite fragile (compare 4.5).

3 How do I...

3.1 How can I incrementally display a paragraph of text ?

The easiest solution is to use `\parstepwise`, but if the arguments of `\step` are long, you'll get problems with line breaks, as `\parstepwise` forces `\step` to put its argument in a box.

You can use `\hidetext` like this:

```
1 \stepwise[\let\hidestepcontents=\hidetext]  
2 {\step{Line breaks} \step{work in here.}}
```

yields

Line breaks

But note that `\hidetext`, being implemented using the `soul` package, is quite fragile (compare 4.5).

3 How do I...

3.1 How can I incrementally display a paragraph of text?

The easiest solution is to use `\parstepwise`, but if the arguments of `\step` are long, you'll get problems with line breaks, as `\parstepwise` forces `\step` to put its argument in a box.

You can use `\hidetext` like this:

```
1 \stepwise[\let\hidestepcontents=\hidetext]
2 {\step{Line breaks} \step{work in here.}}
```

yields

Line breaks work
in here.

But note that `\hidetext`, being implemented using the `soul` package, is quite fragile (compare 4.5).

If you're not using structured backgrounds, `\hidevanish` is another alternative which can be used exactly like `\hidetext`, but is much more robust (note that this will fail whenever your text should appear in front of different background colors, for any reason).

In the argument of `\hidevanish`, which uses `\textcolor`, paragraph breaks are not allowed. Using `\vstep` is a little less restrictive:

```
1 \stepwise
2 {%
3   {\vstep Line and paragraph breaks
4   \vstep work in here.\par Yeah!}%
5 }
```

yields



If you're not using structured backgrounds, `\hidevanish` is another alternative which can be used exactly like `\hidetext`, but is much more robust (note that this will fail whenever your text should appear in front of different background colors, for any reason).

In the argument of `\hidevanish`, which uses `\textcolor`, paragraph breaks are not allowed. Using `\vstep` is a little less restrictive:

```
1 \stepwise
2 {%
3   {\vstep Line and paragraph breaks
4   \vstep work in here.\par Yeah!}%
5 }
```

yields

Line and paragraph breaks

If you're not using structured backgrounds, `\hidevanish` is another alternative which can be used exactly like `\hidetext`, but is much more robust (note that this will fail whenever your text should appear in front of different background colors, for any reason).

In the argument of `\hidevanish`, which uses `\textcolor`, paragraph breaks are not allowed. Using `\vstep` is a little less restrictive:

```
1 \stepwise
2 {%
3   {\vstep Line and paragraph breaks
4   \vstep work in here.\par Yeah!}%
5 }
```

yields

Line and paragraph breaks work
in here.
Yeah!

To facilitate the decision, here's a side-by-side comparison of the pros and cons:

`\parstepwise`:

- + robust
- + works with structured backgrounds
- no automatic line breaks in `\step`'s argument
- no paragraph breaks in `\step`'s argument

`\hidetext`:

- very fragile
- + works with structured backgrounds
- + allows automatic line breaks in `\step`'s argument
- no paragraph breaks in `\step`'s argument

`\hidevanish:`

- + robust
- fails with structured backgrounds
- + allows automatic line breaks in `\step`'s argument
- no paragraph breaks in `\step`'s argument

`\vstep:`

- + very robust
- fails with structured backgrounds
- + allows automatic line breaks
- + allows paragraph breaks

3.2 Instead of making text appear ‘out of nowhere’, I’d rather just change colors from ‘dimmed’ to normal.

There are some analogies between this item and 3.1.

If you’re using `texpower`’s standard colors, probably `\hidedimmed` does what you want:

```
1 \stepwise[\let\hidestepcontents=\hidedimmed]
2 {%
3   \step{This works with} \step{\emph{all}}
4   \step{\highlighttext{highlighting} commands.}%
5 }
```

yields

This works with all high-
lighting commands.

3.2 Instead of making text appear ‘out of nowhere’, I’d rather just change colors from ‘dimmed’ to normal.

There are some analogies between this item and 3.1.

If you’re using `texpower`’s standard colors, probably `\hidedimmed` does what you want:

```
1 \stepwise[\let\hidestepcontents=\hidedimmed]
2 {%
3   \step{This works with} \step{\emph{all}}
4   \step{\highlighttext{highlighting} commands.}%
5 }
```

yields

This works with all high-
lighting commands.

3.2 Instead of making text appear ‘out of nowhere’, I’d rather just change colors from ‘dimmed’ to normal.

There are some analogies between this item and 3.1.

If you’re using `texpower`’s standard colors, probably `\hidedimmed` does what you want:

```
1 \stepwise[\let\hidestepcontents=\hidedimmed]
2 {%
3   \step{This works with} \step{\emph{all}}
4   \step{\highlighttext{highlighting} commands.}%
5 }
```

yields

This works with all high-
lighting commands.

3.2 Instead of making text appear ‘out of nowhere’, I’d rather just change colors from ‘dimmed’ to normal.

There are some analogies between this item and 3.1.

If you’re using `texpower`’s standard colors, probably `\hidedimmed` does what you want:

```
1 \stepwise[\let\hidestepcontents=\hidedimmed]
2 {%
3   \step{This works with} \step{\emph{all}}
4   \step{\highlighttext{highlighting} commands.}%
5 }
```

yields

This works with all high-
lighting commands.

In the argument of `\hidedimmed`, which uses `\textcolor`, paragraph breaks are not allowed. Using `\dstep` is a little less restrictive. The following achieves the same result as above:

```
1 \stepwise
2 {%
3   \dstep This works with \dstep \emph{all}
4   \dstep \highlighttext{highlighting} commands.%
5 }
```

If the dimmed colors look too fancy to you, you can also use `\vstep` for this purpose, setting `\vanishcolor` to some ‘dimmed’ color:

```
1 \stepwise[\renewcommand{\vanishcolor}{inactivecolor}]
2 {%
3   \vstep This works with \vstep \emph{all}
4   \vstep \highlighttext{highlighting} commands.%
5 }
```

yields

This works with all 
 commands.

If the dimmed colors look too fancy to you, you can also use `\vstep` for this purpose, setting `\vanishcolor` to some ‘dimmed’ color:

```
1 \stepwise[\renewcommand{\vanishcolor}{inactivecolor}]
2 {%
3   \vstep This works with \vstep \emph{all}
4   \vstep \highlighttext{highlighting} commands.%
5 }
```

yields

This works with all 
 commands.

If the dimmed colors look too fancy to you, you can also use `\vstep` for this purpose, setting `\vanishcolor` to some ‘dimmed’ color:

```
1 \stepwise[\renewcommand{\vanishcolor}{inactivecolor}]
2 {%
3   \vstep This works with \vstep \emph{all}
4   \vstep \highlighttext{highlighting} commands.%
5 }
```

yields

This works with all 
 commands.

If the dimmed colors look too fancy to you, you can also use `\vstep` for this purpose, setting `\vanishcolor` to some ‘dimmed’ color:

```
1 \stepwise[\renewcommand{\vanishcolor}{inactivecolor}]
2 {%
3   \vstep This works with \vstep \emph{all}
4   \vstep \highlighttext{highlighting} commands.%
5 }
```

yields

This works with all high-
lighting commands.

Achieving the same with `\hidevanish` is left as an exercise to the reader.

3.3 `\dstep` and `\hidedimmed` work only with `texpower`'s standard colors. How can I dim my own colors?

`texpower` maintains a list of colors which will be affected by `\dimcolors` (which is behind `\dstep` and `\hidedimmed`).

You can add your own colors to this list by issuing `\addTPcolor{mycolor}`. Then you only have to define another color `dmycolor` which will be replaced for `mycolor` automatically when `\dimcolors` is executed.

For instance:

```
1 \definecolor{mycolor}{rgb}{1,0.5,0}%  
2 \definecolor{dmycolor}{rgb}{0.9,0.8,0.6}%  
3 \addTPcolor{mycolor}  
4 \stepwise  
5 {\dstep My \emph{own} \dstep \textcolor{mycolor}{color}.}
```

yields My own color.

For instance:

```
1 \definecolor{mycolor}{rgb}{1,0.5,0}%  
2 \definecolor{dmycolor}{rgb}{0.9,0.8,0.6}%  
3 \addTPcolor{mycolor}  
4 \stepwise  
5 {\dstep My \emph{own} \dstep \textcolor{mycolor}{color}.}
```

yields My own color.

For instance:

```
1 \definecolor{mycolor}{rgb}{1,0.5,0}%  
2 \definecolor{dmycolor}{rgb}{0.9,0.8,0.6}%  
3 \addTPcolor{mycolor}  
4 \stepwise  
5 {\dstep My \emph{own} \dstep \textcolor{mycolor}{color}.}
```

yields My own color.

Note that if you ever wish to use `\enhancecolors` or `\highlightenhanced`, you'll also need an enhanced version of your new color named `emycolor`.

If you wish to use one of the commands `\whitebackground`, `\lightbackground`, `\darkbackground`, or `\blackbackground`, you'll need even more variants of your new color. In this case, you'll better define it in the file `tpsettings.cfg` (which contains an example).

4 Problems

4.1 I'm loading the `texpower` package, but dynamic features don't seem to work.

Remember that you have to turn on dynamic features explicitly by giving the `display` option either to `texpower` or as a global option. Otherwise, a printout version of your document is produced.

4.2 When I use the `colormath` option, my displayed formulae are not colored.

Don't use the $\text{T}_\text{E}\text{X}$ environment `$$...$$` for displayed formulae if you want to profit from math coloring.

`texpower` supports $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$'s environments `\[...\]`, `displaymath`, `equation`, `eqnarray`, and `eqnarray*`. It also works with the diverse displayed math environments from the `amsmath` package.

Replacing `$$...$$` everywhere by `\[...\]` should solve this problem.

4.3 It seems I can't use `\vfill` in combination with `\pause`.

This is a problem indeed, as \LaTeX never gets to see anything after `\pause` when the first part of the sequence is produced. You can use `\vfill` with `\stepwise` if you

1. use a configuration where `\step` leaves blank space (to ensure proper vertical spacing);
2. put all `\vfills` into the argument of `\stepwise`, outside the argument of any `\step`.

For instance:

```
1 \parstepwise
2 {\step{One.}\vfill\step{Two.}\vfill\step{Three.}}
```

yields



4.3 It seems I can't use `\vfill` in combination with `\pause`.

This is a problem indeed, as \LaTeX never gets to see anything after `\pause` when the first part of the sequence is produced. You can use `\vfill` with `\stepwise` if you

1. use a configuration where `\step` leaves blank space (to ensure proper vertical spacing);
2. put all `\vfills` into the argument of `\stepwise`, outside the argument of any `\step`.

For instance:

```
1 \parstepwise
2 {\step{One.}\vfill\step{Two.}\vfill\step{Three.}}
```

yields

One.

4.3 It seems I can't use `\vfill` in combination with `\pause`.

This is a problem indeed, as \LaTeX never gets to see anything after `\pause` when the first part of the sequence is produced. You can use `\vfill` with `\stepwise` if you

1. use a configuration where `\step` leaves blank space (to ensure proper vertical spacing);
2. put all `\vfills` into the argument of `\stepwise`, outside the argument of any `\step`.

For instance:

```
1 \parstepwise
2 {\step{One.}\vfill\step{Two.}\vfill\step{Three.}}
```

yields

One.

Two.

4.3 It seems I can't use `\vfill` in combination with `\pause`.

This is a problem indeed, as \LaTeX never gets to see anything after `\pause` when the first part of the sequence is produced. You can use `\vfill` with `\stepwise` if you

1. use a configuration where `\step` leaves blank space (to ensure proper vertical spacing);
2. put all `\vfills` into the argument of `\stepwise`, outside the argument of any `\step`.

For instance:

```
1 \parstepwise
2 {\step{One.}\vfill\step{Two.}\vfill\step{Three.}}
```

yields

One.
Two.
Three.

4.4 When I use $\text{\LaTeX}+\text{dvips}+\text{distiller}$, the result looks strange and ‘hyper’ features don’t work.

Check the `log` file of your document. If it contains the line

```
*hyperref using default driver hypertex*
```

then the default hyperref driver for your system is not suited for processing by `dvips+distiller`.

Either you set another default driver (for instance, in the file `hyperref.cfg`), or you use the option `dvips` in your document as a global option or an option to `\usepackage{hyperref}`. See the documentation of the `hyperref` package for details.

4.5 When using `\highlighttext` or `\hidetext`, I'm getting strange error messages.

Note that both these commands are implemented using the `soul` package. `soul` has some rather severe restrictions concerning what is allowed to appear in the argument of commands using it. Consult the documentation of `soul` for a detailed description of these restrictions.

The most prominent one is that almost no \LaTeX command is allowed in the argument of a command implemented using `soul`. For instance, to use an emphasis or highlighting command like `\emph`, you have to use a sequence of `\highlighttext` commands, putting `\emph` 'outside'. Expect glitches in display quality though.

```
\highlighttext{This }\emph{\highlighttext{annoying }}%  
\highlighttext{behaviour} yields This annoying  
behaviour.
```

Another restriction is that accents are separated from the characters they belong to and break. You have to enclose the complete accented character with braces or use an appropriate input encoding, typing accented characters ‘as one’.

`\highlighttext{S{"u}\ss es}` yields Süßes

`\highlighttext{Süßes}` yields Süßes

4.6 Inside the argument of `\stepwise`, all counters seem to be freezed on all pages of the sequence generated. How can I use a self-defined counter which does not freeze ?

Freezing counters is a desirable behaviour in general, for instance to stop equation numbers from going astray.

But `texpower` maintains a list of counters which are not freezed, containing for instance the counter `step`.

If you need a counter for special effects while the incremental sequence is generated (for instance: generating a sequence of MetaPost figures with the `emp` and `feynmp` packages), use

```
1 \releasecounter{mycounter}
```

to release the counter `mycounter`.