

Idea Vault

Tomasz M. Czarkowski

March 28, 2025

1 Why

Assume, You are writing something large. For example, like me, a TT RPG system.

Often You will find Yourself wanting to declare some concept (idea, if You indulge me) and later print it all over the place, without needing to copy and paste **it**

For really complicated projects, You may want to classify some ideas using tags, or make them depend on each other.

This is what the Idea Vault is used for.

2 Dependencies

Lua \LaTeX — this is Lua \LaTeX package and will not work without Lua.

Additionally, Idea Vault requires some packages (most of them for styling the printed idea).

- `bookmark`
- `xargs`
- `luacode`
- `xpatch`
- `mdframed`
- `needspace`

3 Usage

For idea creation there is `\createIdea`, while for printing them there are a few more:

- `\printIdea` – prints one idea
- `\printAllWithPrefix` – prints all ideas in given prefix
- `\printAllWithTag` – prints all ideas in given prefix that have given tag

3.1 `\createIdea`

`\createIdea`[prefix]{name}{content}[dependencies][tags][weight]

- **prefix**: prefix to locate the idea in. By default, equals to `default`.
- **name**: name of the idea. If such name exists in prefix, it will be overwritten
- **content**: valid LaTeX code for the content of idea. It will be evaluated at invocation time.
- **dependencies**: each dependency is Lua pair of strings, multiple dependencies are separated with comma.
- **tags**: each tag is Lua string, multiple tags are separated with comma.
- **weight**: a number used for sorting ideas when multiple are printed at the same time. By default, equal to `-1`. Special value of `"auto"` makes it number weight sequentially, starting from 1 and growing by 1 per step — it's useful mostly for preserving order of ideas on mass printing.

Note: due to how `xkeyval` treats `{}`, for ideas with exactly one dependency, You need to add one space after the dependency, while for multi dependency there is no need.

Example:

```
\createIdea[somePrefix]{Not So Great Idea}{
\par Boring content
}[]["tag2"][1]
```

```
\createIdea[somePrefix]{Great Idea}{
  \par Long content
  \par Very Interesting
}{{"somePrefix", "Not So Great Idea"} }["tag1"][13]
```

3.2 `\printIdea`

```
\printIdea[prefix] [style]{name}
```

- `prefix`: prefix to locate the idea in. By default, equals to `default`.
- `style`: style for printing the idea. For options, see later.
- `name`: name of the idea.

Example:

```
\printIdea[somePrefix] [lcbf]{Great Idea}
```

3.3 `\printAllWithPrefix`

```
\printIdea[prefix] [style]
```

- `prefix`: prefix to print.
- `style`: style for printing the idea. For options, see later.

Example:

```
\printAllWithPrefix[somePrefix] [lcbf]
```

3.4 `\printAllWithTag`

```
\printIdea[prefix] [style]{tag}
```

- `prefix`: prefix to locate the ideas in.
- `style`: style for printing the idea. For options, see later.
- `tag`: tag to print

Example:

```
\printAllWithTag[somePrefix] [lcbf]{tag1}
```

4 Style

Printing style is decided by one letter flags.

- **f**: Frame whole idea
- **c**: Center whole idea
- **b**: Create bookmark leading to the idea
- **l**: Use Large font for the name
- **L**: Use LARGE font for the name
- **h**: Use huge font for the name
- **H**: Use Huge font for the name
- **s**: Reserve some space using needspace to avoid awkward pagebreaks
- **e**: emph the name
- **q**: Do not print the name
- **p**: New page before printing the idea (useful when doing serial printing, for example via `printAllWithTag`)
- **P**: New page after printing the idea (useful when doing serial printing, for example via `printAllWithTag`)

5 Debugging

By default, this package avoids spamming too much to stdout, but sometimes it would be useful to see what idea creation/printing is triggering some problems.

5.1 `\debugEnable`

`\debugEnable`

Enables debug output — remember, it may be quite overwhelming if there are a lot of ideas.

5.2 `\debugDisable`

`\debugDisable`

Disables debug output. Debug output is disabled by default, so it's only use is to revert the effect of enabling it.

6 Example

This is short example of code, and it's result. Extended example can be found in file `example.tex`.

6.1 Code

```
\createIdea[somePrefix]{Not So Great Idea}{
\par Boring content
}["tag2"][1]

\createIdea[somePrefix]{Great Idea}{
\par Long content
\par Very Interesting
}["somePrefix", "Not So Great Idea"] ["tag1"][13]

\createIdea[somePrefix2]{Third Idea}{
\par Third try is even longer
\par But not very
\par Only 3 lines
}["somePrefix", "Not So Great Idea"], {"somePrefix", "Great Idea"} [] [21]

\par Here is a Great Idea:
\printIdea[somePrefix][lcf]{Great Idea}
\vspace{1em}
\par Here are all ideas in prefix \texttt{somePrefix}:
\printAllWithPrefix[somePrefix][lcf]
\vspace{1em}
\par Here are all ideas in prefix \texttt{somePrefix} with tag \texttt{tag1}:
\printAllWithTag[somePrefix][lcf]{tag1}
\vspace{1em}
\par And here is Third Idea, presenting nested dependencies:
\printIdea[somePrefix2][lcf]{Third Idea}
```

6.2 Result

Here is a Great Idea:

Great Idea
Long content Very Interesting
Not So Great Idea
Boring content

Here are all ideas in prefix `somePrefix`:

Not So Great Idea
Boring content

Great Idea
Long content Very Interesting
Not So Great Idea
Boring content

Here are all ideas in prefix `somePrefix` with tag `tag1`:

Great Idea
Long content Very Interesting
Not So Great Idea
Boring content

And here is Third Idea, presenting nested dependencies:

Third Idea

Third try is even longer
But not very
Only 3 lines

Great Idea

Long content
Very Interesting

Not So Great Idea

Boring content

Not So Great Idea

Boring content