The pgfkeysearch Package
A Search Extension for pgfkeys
Version 1.5

Alceu Frigeri*

November 2025

Abstract

The command `\pgfkeysvalueof`, unlike `\pgfkeys` command, doesn't use the *.unknown* handler or offers the option to search for a key in other paths, and raises an error if the key isn't defined in the given path.

The following commands will recursively search for a key in a collection of paths.

Contents

1 Package Options

The default search behaviour assumes that all keys defined by a package or document are under a uniquely defined path, meaning, no root keys. For instance, given the path /A/B/C/D, the following commands will look, first, at /A/B/C/D/⟨key⟩, then /A/B/C/⟨key⟩, and so on, until /A/⟨key⟩, stopping at the first hit. This can be changed with the `root search` package option.

By default, the value stored in a *pgfkey* key will be recovered, but, with the option `key=macro` a macro will be returned, such that, later changes to the searched key (with *pgfkey*) will be reflected by the recovered macro, the recovered macro will, effectively, be an alias for it.

Lastly, if a key isn't found, the returning macro will always be cleared up (`settings=new`), but, with `settings=old`, the *expl3* commands (see 3) won't clear up the returning macro (old/original behaviour of this package).

root search (default: false) If set, the *path root* will also be included in the search, meaning it will look if /⟨*key*⟩, as last resort, is defined.

key (default: *value*. Possible values: *value* or *macro*. As said, the default behaviour is to recover the value stored in a *pgfkey*. With *key=macro*, a macro "pointing to" the *pgfkey* will be recovered.

settings (default: *new*). Possible values: *old* or *new*. If set to *old*, this will revert to the original *expl3* behaviour, whereas if the key wasn't found, the returning variable won't be cleared up (no assignment taking place).

> ***Note:*** With `root search` set, the root key (/⟨key⟩) will be look at for every path in the path list. For instance `\pgfkeysearch {/A/B/C,/X/Y,/Z/T}{key}`, /⟨key⟩ (at the root) will be tried up to three times.

*https://github.com/alceu-frigeri/pgfkeysearch

2 User Document Commands

Those commands are meant to be used at Document level. For packages, one is advised to use the ones defined at 3.

\pgfkeysearchsettings `\pgfkeysearchsettings {⟨options⟩}`

new: 2025/05/27

To change the search behaviour, middle document. ⟨options⟩ can be any package option (see 1).

\pgfkeygetvalueof `\pgfkeygetvalueof {⟨single-path⟩}{⟨key⟩}{⟨macro⟩}`
\pgfkeyget `\pgfkeyget {⟨single-path⟩}{⟨key⟩}{⟨macro⟩}`
\pgfkeygetvalueof*TF* `\pgfkeygetvalueofTF {⟨single-path⟩}{⟨key⟩}{⟨macro⟩}{⟨if-found⟩}{⟨if-not⟩}`
\pgfkeyget*TF* `\pgfkeygetTF {⟨single-path⟩}{⟨key⟩}{⟨macro⟩}{⟨if-found⟩}{⟨if-not⟩}`

new: 2025/11/03

Those are "non searching" variants (faster than the searching variants), whereas ⟨single-path⟩ is the single location/path to be looked at. ⟨key⟩ is the desired key, and ⟨macro⟩ is the macro/command that will receive (store) the key value (if one is found). ⟨macro⟩ will be set with the found (if any) value.

Note: If `key=value`, then the key value will be recovered. Otherwise, if `key=macro` then ⟨macro⟩ will "point to" (*pgfkey*) key.

Note: \pgfkeykeg and \pgfkeygetvalueof are aliases to each other. Same for \pgfkeyget*TF* and \pgfkeygetvalueof*TF*

Note: Those commands aren't expandable, though, once retrieved, the returning macro can be used in an expandable context.

Note: If ⟨key⟩ isn't found, ⟨macro⟩ will be empty, no warning or error will be raised.

\pgfkeysearch `\pgfkeysearch {⟨path-list⟩}{⟨key⟩}{⟨macro⟩}`
\pgfkeysearchvalueof `\pgfkeysearchvalueof {⟨path-list⟩}{⟨key⟩}{⟨macro⟩}`
\pgfkeysearch*TF* `\pgfkeysearchTF {⟨path-list⟩}{⟨key⟩}{⟨macro⟩}{⟨if-found⟩}{⟨if-not⟩}`
\pgfkeysearchvalueof*TF* `\pgfkeysearchvalueofTF {⟨path-list⟩}{⟨key⟩}{⟨macro⟩}{⟨if-found⟩}{⟨if-not⟩}`

updated: 2024/01/11

⟨path-list⟩ is a comma separated list (clist) of paths (can be a single one). ⟨key⟩ is the desired key, and ⟨macro⟩ is the macro/command that will receive (store) the key value (if one is found). ⟨key⟩ will be searched for in the many paths from ⟨path-list⟩ as described in 1. ⟨macro⟩ will be set with the found (if any) value. The branch versions will also execute either ⟨if-found⟩ or ⟨if-not⟩.

Note: If `key=value`, then the key value will be recovered. Otherwise, if `key=macro` then ⟨macro⟩ will "point to" (*pgfkey*) key.

Note: \pgfkeysearch and \pgfkeysearchvalueof are aliases to each other. Same with \pgfkeysearchvalueof*TF* and \pgfkeysearch*TF*.

Note: Those commands aren't expandable, though, once retrieved, the returning macro can be used in an expandable context.

Note: If ⟨key⟩ isn't found, ⟨macro⟩ will be empty, no warning or error will be raised.

2.1 Example

Given the following pgfkeys:

```
\pgfkeys{%
   /tikz/A/.cd,
   keyA/.initial={keyA at /tikz/A},
   keyB/.initial={keyB at /tikz/A},
   %
   B/.cd,
   keyA/.initial={keyA at /tikz/A/B},
   keyC/.initial={keyC at /tikz/A/B},
   %
   C/.cd,
   keyX/.initial={keyX at /tikz/A/B/C}
 }
```

Key values can be retrieved and used as:

```
\pgfkeysearch{/tikz/X,/tikz/A/B/C}{keyA}{\VALkeyA}
\pgfkeysearch{/tikz/X/Y,/tikz/A/B/C}{keyB}{\VALkeyB}
\pgfkeysearch{/tikz/X/Y,/tikz/Y/Y,/tikz/A/B/C}{keyC}{\VALkeyC}
\pgfkeysearch{/tikz/X/Y,/tikz/Y/Y,/tikz/A/B/C}{keyX}{\VALkeyX}
```

```
I got for keyA: \textbf{\VALkeyA} \par
I got for keyB: \textbf{\VALkeyB} \par
I got for keyC: \textbf{\VALkeyC} \par
I got for keyX: \textbf{\VALkeyX} \par
```

I got for keyA: **keyA at /tikz/A/B**
I got for keyB: **keyB at /tikz/A**
I got for keyC: **keyC at /tikz/A/B**
I got for keyX: **keyX at /tikz/A/B/C**

3 Expl3 Commands

Deprecation: A warning will be raised if either `\pgfkeysearch_keysearch:nnnTF` or `\pgfkeysearch_multipath_keysearch:nnnTF` are used.

`\pgfkeysearch_settings:n` `\pgfkeysearch_settings:n {⟨options⟩}`

new: 2025/05/27

To change the search behaviour, middle document. ⟨options⟩ can be any package option (see 1).

`\pgfkeysearch_keyget:nnN` `\pgfkeysearch_keyget:nnN {⟨single-path⟩} {⟨key⟩} {⟨tl-var⟩}`
`\pgfkeysearch_keyget:nnNTF` `\pgfkeysearch_keyget:nnNTF {⟨single-path⟩} {⟨key⟩} {⟨tl-var⟩} {⟨if-found⟩} {⟨if-not⟩}`

new: 2025/11/03

⟨key⟩ is the desired key, and ⟨tl-var⟩ is a token list variable that will receive the key value, if one is found. ⟨key⟩ will be looked at ⟨single-path⟩ *only*.
`\pgfkeysearch_keyget:` is faster than the search variants.

Note: If ⟨key⟩ isn't found ⟨tl-var⟩ will be cleared up (new default). But, with option `settings=old` ⟨tl-var⟩ will preserve whatever value it had, no assignment will be made. In both cases, no warning or error will be raised.

Note: If `key=value`, then the key value will be recovered. Otherwise, if `key=macro` then ⟨macro⟩ will "point to" (*pgfkey*) key.

`\pgfkeysearch_keysearch:nnN` `\pgfkeysearch_keysearch:nnN {⟨single-path⟩} {⟨key⟩} {⟨tl-var⟩}`
`\pgfkeysearch_keysearch:nnNTF` `\pgfkeysearch_keysearch:nnNTF {⟨single-path⟩} {⟨key⟩} {⟨tl-var⟩} {⟨if-found⟩} {⟨if-not⟩}`

updated: 2025/05/26
updated: 2025/11/03

⟨key⟩ is the desired key, and ⟨tl-var⟩ is a token list variable that will receive the key value, if one is found. ⟨key⟩ will be searched for in ⟨single-path⟩ as described in 1.
`\pgfkeysearch_keysearch:nnNTF` is slightly faster than the more generic multi-path version.

Note: If ⟨key⟩ isn't found ⟨tl-var⟩ will be cleared up (new default). But, with option `settings=old` ⟨tl-var⟩ will preserve whatever value it had, no assignment will be made. In both cases, no warning or error will be raised.

Note: If `key=value`, then the key value will be recovered. Otherwise, if `key=macro` then ⟨macro⟩ will "point to" (*pgfkey*) key.

`\pgfkeysearch_multipath_keysearch:nnN` `\pgfkeysearch_multipath_keysearch:nnN {⟨path-list⟩} {⟨key⟩} {⟨tl-var⟩}`
`\pgfkeysearch_multipath_keysearch:nnNTF` `\pgfkeysearch_multipath_keysearch:nnNTF {⟨path-list⟩} {⟨key⟩} {⟨tl-var⟩} {⟨if-found⟩} {⟨if-not⟩}`

updated: 2025/05/26
updated: 2025/11/03

Given a comma separated ⟨path-list⟩, this will call `\pgfkeysearch_keysearch:nnNTF` for each path in ⟨path-list⟩, until ⟨key⟩ is found.

Note: If ⟨key⟩ isn't found ⟨tl-var⟩ will be cleared up (new default). But, with option `settings=old` ⟨tl-var⟩ will preserve whatever value it had, no assignment will be made. In both cases, no warning or error will be raised.

Note: The document level commands (in 2) are just wrappers to this command.

Note: If `key=value`, then the key value will be recovered. Otherwise, if `key=macro` then ⟨macro⟩ will "point to" (*pgfkey*) key.