

The pgfkeysearch Package

A Search Extension for pgfkeys

Version 1.4

Alceu Frigeri*

May 2025

Abstract

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

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

Contents

1	Package Options	1
2	User Document Commands	1
2.1	Example	2
3	Expl3 Commands	3

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 following package option:

`root search` If set, the *path root* will also be included in the search, meaning it will look if `/<key>`, as last resort, is defined.

Note: If 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>` will be tried up to three times.

`\pgfkeysearchsettings` `\pgfkeysearchsettings {<options>}`

new: 2025/05/27

To change the search behaviour, middle document, including or not the path root. `<options>` are any valid package option (for now just `root search`).

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.

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

<code>\pgfkeysearchvalueof</code>	<code>\pgfkeysearchvalueof {<path-list>} {<key>} {<macro>}</code>
<code>\pgfkeysearch</code>	<code>\pgfkeysearch {<path-list>} {<key>} {<macro>}</code>

update: 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.

Note: `\pgfkeysearch` and `\pgfkeysearchvalueof` are aliases to each other.

Note: These commands aren't expandable, that's the reason to store the key value in a macro and not just place the found value in the input stream.

Note: If `<key>` isn't found, `<macro>` will be empty, no warning or error will be raised.

<code>\pgfkeysearchvalueofTF</code>	<code>\pgfkeysearchvalueofTF {<path-list>} {<key>} {<macro>} {<if-found>} {<if-not>}</code>
<code>\pgfkeysearchTF</code>	<code>\pgfkeysearchTF {<path-list>} {<key>} {<macro>} {<if-found>} {<if-not>}</code>

update: 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 was found). These branch versions will also execute either `<if-found>` or `<if-not>`.

Note: `\pgfkeysearchvalueofTF` and `\pgfkeysearchTF` are aliases to each other.

Note: These commands aren't expandable, that's the reason to store the key value in a macro and not just place the found value in the input stream.

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 as:

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

and finally used as:

```
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

`\pgfkeysearch_settings:n` `\pgfkeysearch_settings:n` {<options>}

new: 2025/05/27

To change the search behaviour, middle document, including or not the path root. <options> are any valid package option (for now just *root search*, see 1, notice the space...).

`\pgfkeysearch_keysearch:nnNTF` `\pgfkeysearch_keysearch:nnNTF` {<single-path>} {<key>} {<tl-var>} {<if-found>} {<if-not>}

update: 2025/05/26

<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, no assignment will be made (to <tl-var>), unlike the user's document one (defined at 2). No warning or error will be raised.

Note: The old signature `\pgfkeysearch_keysearch:nnnTF` is deprecated, and will raise a warning if used.

`\pgfkeysearch_multipath_keysearch:nnNTF` `\pgfkeysearch_multipath_keysearch:nnNTF` {<path-list>} {<key>} {<tl-var>} {<if-found>} {<if-not>}

update: 2025/05/26

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, no assignment will be made (to <tl-var>), unlike the user's document one (defined at 2). No warning or error will be raised.

Note: `\pgfkeysearchvalueof`, `\pgfkeysearch`, `\pgfkeysearchvalueofTF` and `\pgfkeysearchTF` are just wrappers to `\pgfkeysearch_multipath_keysearch:nnNTF`.

Note: The old signature `\pgfkeysearch_multipath_keysearch:nnnTF` is deprecated, and will raise a warning if used.