



TeX Environment Variable Expansion

A search path can come from many sources. In priority order (meaning Kpathsea will use whichever it finds first):

- A user-set environment variable, e.g., **TEXINPUTS**.
- A program-specific configuration file, e.g., an `S / a : / b` line in `Dvips config.ps`.
- A line in a Kpathsea configuration file `texmf.cnf`, e.g., `TEXINPUTS=/c:/d`. See section below.
- The compile-time default (specified in `kpathsea/paths.h`).

In any case, once the path specification to use is determined, its evaluation is independent of its source. These sources may also be combined via default expansion. See the next section.

Default Expansion

If the highest-priority search path (in the list in the previous section) contains an "extra colon" (i.e., leading, trailing, or doubled), Kpathsea inserts the next-highest-priority search path that is set at that point. If that search path has an extra colon, the same happens with the next-highest.

For example, given

```
% setenv TEXINPUTS /home/karl:
```

and a `TEXINPUTS` value from `texmf.cnf` of

```
.: $TEXMF//tex
```

then the final value used for searching will be:

```
/home/karl:.: $TEXMF//tex
```

You can trace this by debugging "paths."

Minor technical point: Since it would be useless to insert the default value in more than one place, Kpathsea changes only one extra `‘.’` and leaves any others in place (where they will eventually be effectively equivalent to `‘.’`, i.e., the current directory). It checks first for a leading `‘.’`, then a trailing `‘.’`, then a doubled `‘.’`.

Variable Expansion

`$foo` or `${foo}` in a path element is replaced by (1) the value of an environment variable `foo` (if it is set); (2) the value of `foo` from `texmf.cnf` (if any such exists); (3) the empty string.

If the character after the `‘$’` is alphanumeric or `‘_’`, the variable name consists of all consecutive such characters. If the character after the `‘$’` is a `‘{’`, the variable name consists of everything up to the next `‘}’` (braces are not balanced!). Otherwise, Kpathsea gives a warning and ignores the `‘$’` and its following character.

Remember:

- you need to quote the `‘$’`s and braces as necessary for your shell.
- shell variable values cannot be seen by Kpathsea.

For example, given

```
% setenv TEXMF /home/tex
% setenv TEXINPUTS
.: $TEXMF: ${TEXMF}new
```

the final `‘TEXINPUTS` path is the three directories:

```
.: /home/tex: /home/texnew
```

You can trace this by debugging "paths."

Tilde Expansion

A leading `~` or `~USER` in a path element is replaced by the current or `USER`'s home directory, respectively. If `USER` is invalid, or the home directory cannot be determined, Kpathsea uses `‘.’` instead.

For example,

```
% setenv TEXINPUTS ~/mymacros:
```

will prepend a directory `mymacros` in your home directory to the default path.

Subdirectory Expansion

A `‘/’` in a path element following a directory `D` is replaced by all subdirectories of `D`: first those subdirectories directly under `D`, then the subsubdirectories under those, and so on. At each level, the order in which the directories are searched

is unspecified. (It's "directory order", and definitely not alphabetical.)

If you specify any filename components after the '/', only subdirectories which contain those components are included. For example, '/a/b' would expand into directories '/a/1/b', '/a/2/b', '/a/1/1/b', and so on, but not '/a/b/c' or '/a/1'.

Getting Help

If you have any questions regarding these settings, please call the Customer Support Center at (213) 740-5555 or e-mail *consult@usc.edu*.