

**NAME**

`re_comp`, `re_exec` - BSD regex functions

**SYNOPSIS**

```
#define _REGEX_RE_COMP
#include <sys/types.h>
#include <regex.h>

char *re_comp(const char *regex);

int re_exec(const char *string);
```

**DESCRIPTION**

`re_comp()` is used to compile the null-terminated regular expression pointed to by *regex*. The compiled pattern occupies a static area, the pattern buffer, which is overwritten by subsequent use of `re_comp()`. If *regex* is NULL, no operation is performed and the pattern buffer's contents are not altered.

`re_exec()` is used to assess whether the null-terminated string pointed to by *string* matches the previously compiled *regex*.

**RETURN VALUE**

`re_comp()` returns NULL on successful compilation of *regex* otherwise it returns a pointer to an appropriate error message.

`re_exec()` returns 1 for a successful match, zero for failure.

**ATTRIBUTES**

For an explanation of the terms used in this section, see [attributes\(7\)](#).

Interface	Attribute	Value
<code>re_comp()</code> , <code>re_exec()</code>	Thread safety	MT-Unsafe

**CONFORMING TO**

4.3BSD.

**NOTES**

These functions are obsolete; the functions documented in [regcomp\(3\)](#) should be used instead.

**SEE ALSO**

[regcomp\(3\)](#), [regex\(7\)](#), GNU regex manual

**COLOPHON**

This page is part of release 4.10 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.