

**NAME**

`des_read_password`, `des_read_2passwords`, `des_read_pw_string`, `des_read_pw` - Compatibility user interface functions

**SYNOPSIS**

```
#include <openssl/des_old.h>
```

```
int des_read_password(DES_cblock *key, const char *prompt, int verify);
int des_read_2passwords(DES_cblock *key1, DES_cblock *key2,
const char *prompt, int verify);
```

```
int des_read_pw_string(char *buf, int length, const char *prompt, int verify);
int des_read_pw(char *buf, char *buff, int size, const char *prompt, int verify);
```

**DESCRIPTION**

The DES library contained a few routines to prompt for passwords. These aren't necessarily dependent on DES, and have therefore become part of the UI compatibility library.

`des_read_pw()` writes the string specified by *prompt* to standard output turns echo off and reads an input string from the terminal. The string is returned in *buf*, which must have space for at least *size* bytes. If *verify* is set, the user is asked for the password twice and unless the two copies match, an error is returned. The second password is stored in *buff*, which must therefore also be at least *size* bytes. A return code of -1 indicates a system error, 1 failure due to user interaction, and 0 is success. All other functions described here use `des_read_pw()` to do the work.

`des_read_pw_string()` is a variant of `des_read_pw()` that provides a buffer for you if *verify* is set.

`des_read_password()` calls `des_read_pw()` and converts the password to a DES key by calling `DES_string_to_key()`; `des_read_2password()` operates in the same way as `des_read_password()` except that it generates two keys by using the `DES_string_to_2key()` function.

**NOTES**

`des_read_pw_string()` is available in the MIT Kerberos library as well, and is also available under the name `EVP_read_pw_string()`.

**SEE ALSO**

[ui\(3\)](#), [ui\\_create\(3\)](#)

**AUTHOR**

Richard Levitte ([richard@levitte.org](mailto:richard@levitte.org)) for the OpenSSL project (<http://www.openssl.org>).