NAME

PKCS12_newpass - change the password of a PKCS12 structure

SYNOPSIS

```
#include <openssl/pkcs12.h>

int PKCS12_newpass(PKCS12 *p12, const char *oldpass, const char *newpass);
```

DESCRIPTION

PKCS12_newpass() changes the password of a PKCS12 structure.

p12 is a pointer to a PKCS12 structure. **oldpass** is the existing password and **newpass** is the new password.

Each of **oldpass** and **newpass** is independently interpreted as a string in the UTF-8 encoding. If it is not valid UTF-8, it is assumed to be ISO8859-1 instead.

In particular, this means that passwords in the locale character set (or code page on Windows) must potentially be converted to UTF-8 before use. This may include passwords from local text files, or input from the terminal or command line. Refer to the documentation of **UI_OpenSSL**(3), for example.

If the PKCS#12 structure does not have a password, then you must use the empty string "" for **oldpass**. Using NULL for **oldpass** will result in a **PKCS12_newpass()** failure.

If the wrong password is used for **oldpass** then the function will fail, with a MAC verification error. In rare cases the PKCS12 structure does not contain a MAC: in this case it will usually fail with a decryption padding error.

RETURN VALUES

PKCS12_newpass() returns 1 on success or 0 on failure. Applications can retrieve the most recent error from **PKCS12_newpass()** with **ERR_get_error()**.

EXAMPLES

This example loads a PKCS#12 file, changes its password and writes out the result to a new file.

```
#include <stdio.h>
#include <stdlib.h>
#include <openssl/pem.h>
#include <openssl/err.h>
#include <openssl/pkcs12.h>
```

```
int main(int argc, char **argv)
{
    FILE *fp;
    PKCS12 *p12;

    if (argc != 5) {
        fprintf(stderr, "Usage: pkread p12file password newpass opfile\n");
        return 1;
    }
    if ((fp = fopen(argv[1], "rb")) == NULL) {
        fprintf(stderr, "Error opening file %s\n", argv[1]);
        return 1;
    }
    p12 = d2i_PKCS12_fp(fp, NULL);
    fclose(fp);
    if (p12 == NULL) {
        fprintf(stderr, "Error reading PKCS#12 file\n");
        ERR_print_errors_fp(stderr);
        return 1;
    }
    if (PKCS12_newpass(p12, argv[2], argv[3]) == 0) {
        fprintf(stderr, "Error changing password\n");
        ERR_print_errors_fp(stderr);
        PKCS12_free(p12);
        return 1;
    }
    if ((fp = fopen(argv[4], "wb")) == NULL) {
        fprintf(stderr, "Error opening file %s\n", argv[4]);
        PKCS12_free(p12);
        return 1;
    }
    i2d_PKCS12_fp(fp, p12);
    PKCS12_free(p12);
    fclose(fp);
    return 0;
}
```

BUGS

The password format is a NULL terminated ASCII string which is converted to Unicode form internally. As a result some passwords cannot be supplied to this function.

SEE ALSO

PKCS12_create(3), **ERR_get_error**(3), **passphrase-encoding**(7)

COPYRIGHT