NAME

crypt_checksalt — validate a crypt setting string

LIBRARY

Crypt Library (libcrypt, −lcrypt)

SYNOPSIS

```
#include <crypt.h>

int
crypt_checksalt(const char *setting);
```

DESCRIPTION

crypt_checksalt checks the *setting* string against the system configuration and reports whether the hashing method and parameters it specifies are acceptable. It is intended to be used by programs such as login(1) to determine whether the user's passphrase should be re-hashed using the currently preferred hashing method.

RETURN VALUES

The return value is 0 if there is nothing wrong with this setting. Otherwise, it is one of the following constants:

CRYPT_SALT_OK
: *setting* is a fully correct setting string. This constant is guaranteed to equal 0.

CRYPT_SALT_INVALID
: *setting* is not a valid setting string; either it specifies a hashing method that is not known to this version of libxcrypt, or it specifies invalid parameters for the method.

CRYPT_SALT_METHOD_DISABLED (Not implemented, yet)
: *setting* specifies a hashing method that is no longer allowed to be used at all; **crypt** will fail if passed this *setting*. Manual intervention will be required to reactivate the user's account.

CRYPT_SALT_METHOD_LEGACY
: *setting* specifies a hashing method that is no longer considered strong enough for use with new passphrases. **crypt** will still authenticate a passphrase against this setting, but if authentication succeeds, the passphrase should be re-hashed using the currently preferred method.

CRYPT_SALT_TOO_CHEAP (Not implemented, yet)
: *setting* specifies cost parameters that are considered too cheap for use with new passphrases. **crypt** will still authenticate a passphrase against this setting, but if authentication succeeds, the passphrase should be re-hashed using the currently preferred method.

FEATURE TEST MACROS

<crypt.h> will define the macro CRYPT_CHECKSALT_AVAILABLE if **crypt_checksalt** is available in the current version of libxcrypt.

BUGS

Since full configurability is not yet implemented, the current implementation will only ever return **CRYPT_SALT_OK (0)** or **CRYPT_SALT_INVALID** when in voked.

PORTABILITY NOTES

The function **crypt_checksalt** is not part of any standard. It was added to libxcrypt in version 4.3.0.

ATTRIBUTES

For an explanation of the terms used in this section, see `attributes`(7).

Interface	Attribute	Value
`crypt_checksalt`	Thread safety	MT-Safe

SEE ALSO

`crypt`(3), `crypt_gensalt`(3), `crypt`(5)