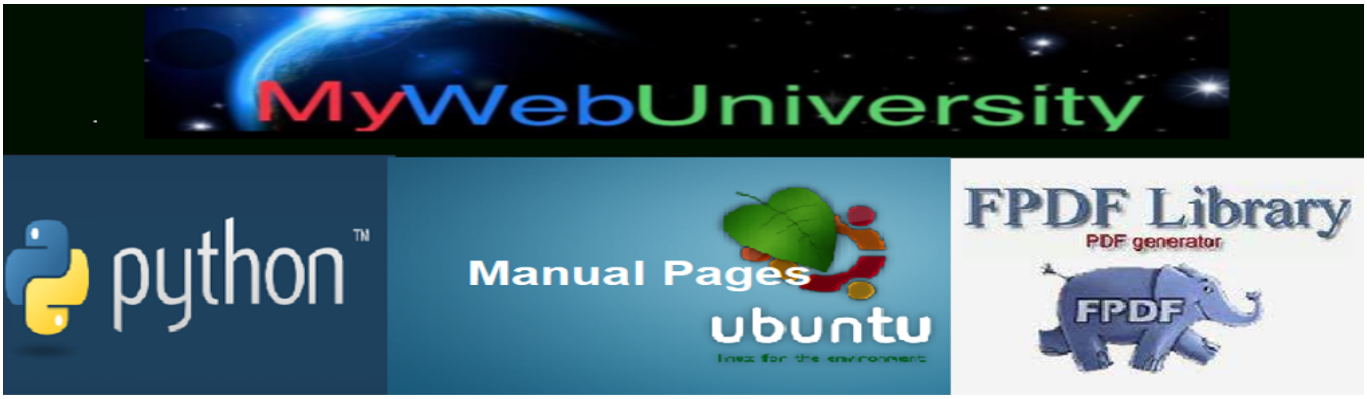


Rocky Enterprise Linux 9.2 Manual Pages on command 'life_cycle-cipher.7ssl'

$ man life_cycle-cipher.7ssl

NAME

life_cycle-cipher - The cipher algorithm life-cycle

DESCRIPTION

All symmetric ciphers (CIPHERs) go through a number of stages in their life-cycle:

start

This state represents the CIPHER before it has been allocated. It is the starting state for any life-cycle transitions.

newed

This state represents the CIPHER after it has been allocated.

initialised

These states represent the CIPHER when it is set up and capable of processing input.

There are three possible initialised states:

initialised using EVP_CipherInit

initialised for decryption using EVP_DecryptInit

initialised for encryption using EVP_EncryptInit

updated

These states represent the CIPHER when it is set up and capable of processing additional input or generating output. The three possible states directly correspond to those for initialised above. The three different streams should not be mixed.

finaled

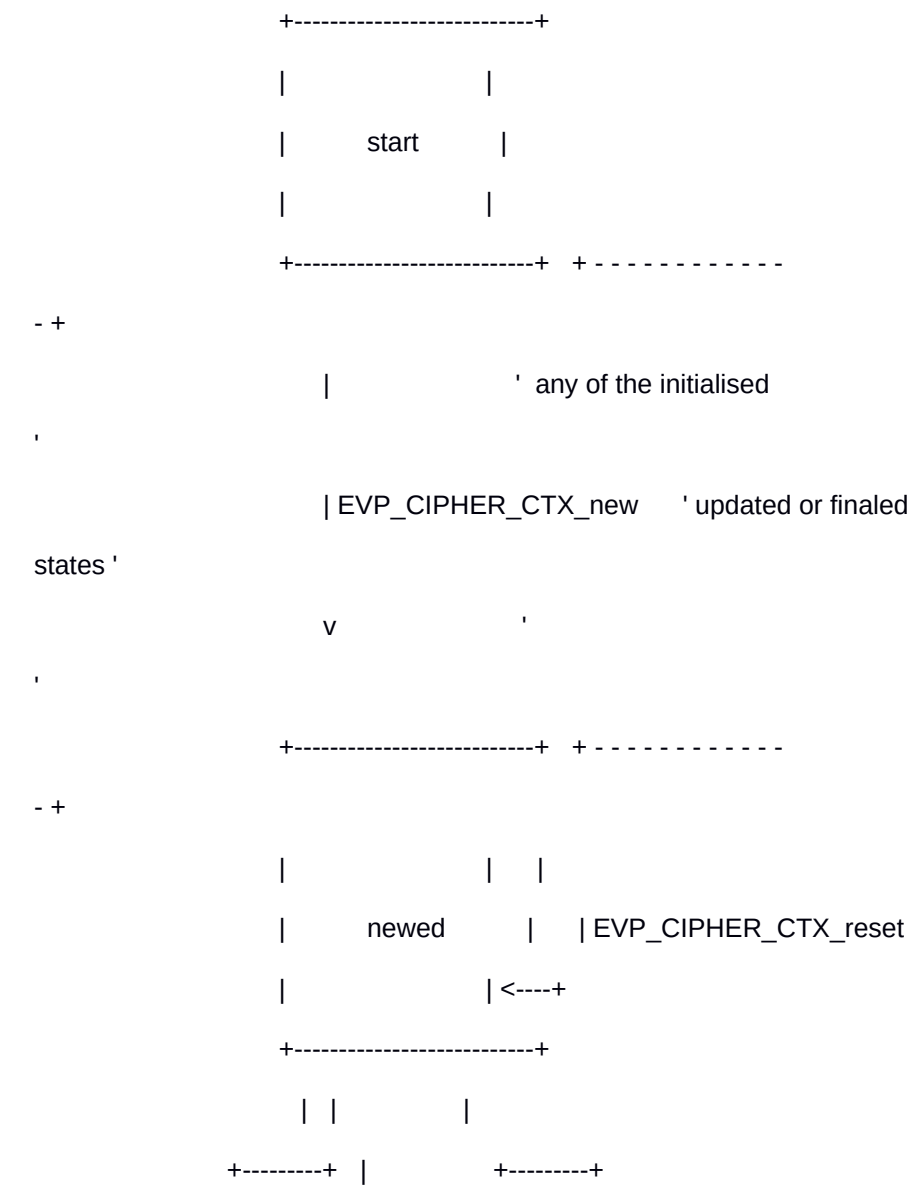
This state represents the CIPHER when it has generated output.

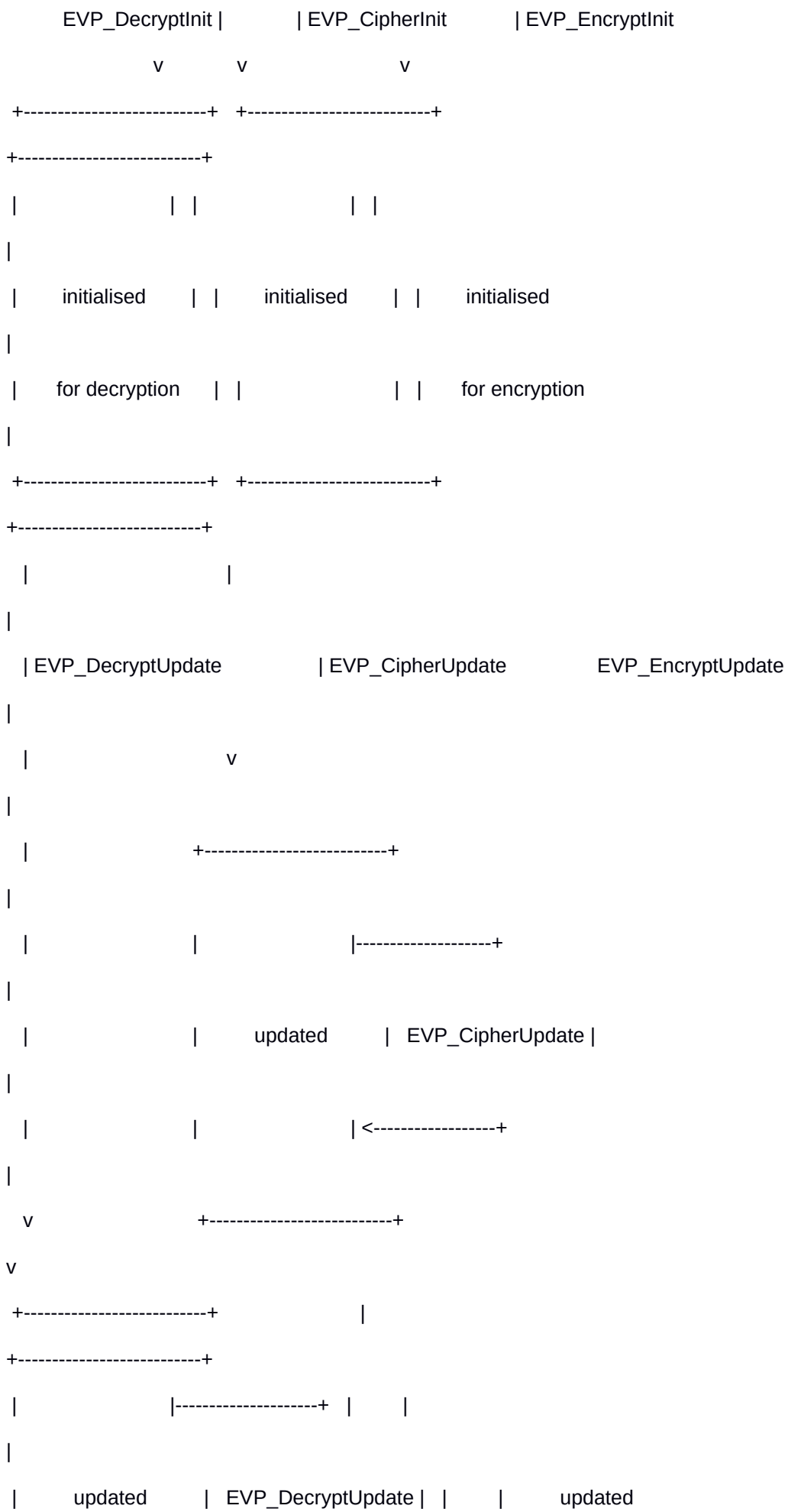
freed

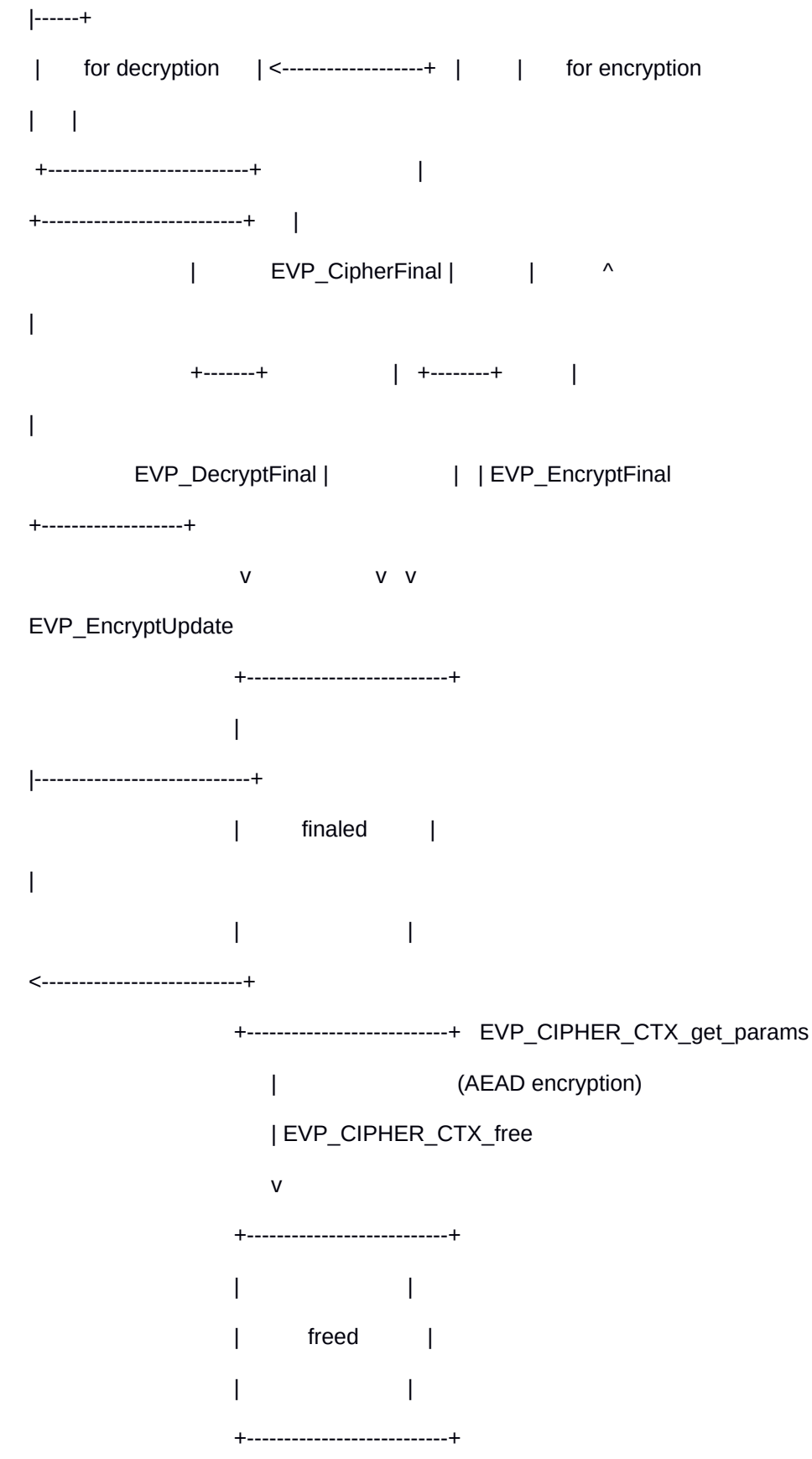
This state is entered when the CIPHER is freed. It is the terminal state for all life-cycle transitions.

State Transition Diagram

The usual life-cycle of a CIPHER is illustrated:




EVP_DecryptInit
EVP_CipherInit
EVP_EncryptInit
initialised for decryption
initialised
initialised for encryption
EVP_DecryptUpdate
EVP_CipherUpdate
EVP_EncryptUpdate
updated
EVP_CipherUpdate
updated
EVP_DecryptUpdate
updated



Formal State Transitions

This section defines all of the legal state transitions. This is the canonical list.

Function Call ---------- Current State

---

start newed initialised updated finaled

initialised updated initialised updated freed

decryption

decryption encryption encryption

EVP_CIPHER_CTX_new newed

EVP_CipherInit initialised initialised initialised initialised

initialised initialised initialised initialised

EVP_DecryptInit initialised initialised initialised initialised

initialised initialised initialised initialised

decryption decryption decryption decryption

decryption decryption decryption decryption

EVP_EncryptInit initialised initialised initialised initialised

initialised initialised initialised initialised

encryption encryption encryption encryption

encryption encryption encryption encryption

EVP_CipherUpdate updated updated

EVP_DecryptUpdate

updated updated

decryption

decryption

EVP_EncryptUpdate

updated updated

encryption

encryption

EVP_CipherFinal finaled

EVP_DecryptFinal

finaled

EVP_EncryptFinal

finaled

EVP_CIPHER_CTX_free freed freed freed freed freed

freed freed freed freed

EVP_CIPHER_CTX_reset newed newed newed newed

newed newed newed newed

EVP_CIPHER_CTX_get_params newed initialised updated

initialised updated initialised updated

decryption

decryption encryption encryption

EVP_CIPHER_CTX_set_params newed initialised updated

initialised updated initialised updated

decryption

decryption encryption encryption

EVP_CIPHER_CTX_gettable_params newed initialised updated

initialised updated initialised updated

decryption

decryption encryption encryption

EVP_CIPHER_CTX_settable_params newed initialised updated

initialised updated initialised updated

decryption

decryption encryption encryption

NOTES

At some point the EVP layer will begin enforcing the transitions described herein.

SEE ALSO

provider-cipher(7), EVP_EncryptInit(3)

COPYRIGHT