Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'life_cycle-pkey.7ossl' command

$ man life_cycle-pkey.7ossl

LIFE_CYCLE-PKEY(7ossl) OpenSSL LIFE_CYCLE-PKEY(7ossl)

NAME

life_cycle-pkey - The PKEY algorithm life-cycle

DESCRIPTION

All public keys (PKEYs) go through a number of stages in their life-cycle:

start

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

newed

This state represents the PKEY after it has been allocated.

decapsulate

This state represents the PKEY when it is ready to perform a private key decapsulation opeartion.

decrypt

This state represents the PKEY when it is ready to decrypt some ciphertext.

derive

This state represents the PKEY when it is ready to derive a shared secret.

digest sign

This state represents the PKEY when it is ready to perform a private key signature operation.

encapsulate

This state represents the PKEY when it is ready to perform a public key encapsulation opeartion.

encrypt

This state represents the PKEY when it is ready to encrypt some plaintext.

key generation

This state represents the PKEY when it is ready to generate a new public/private key.

parameter generation

This state represents the PKEY when it is ready to generate key parameters.

verify

This state represents the PKEY when it is ready to verify a public key signature.

verify recover

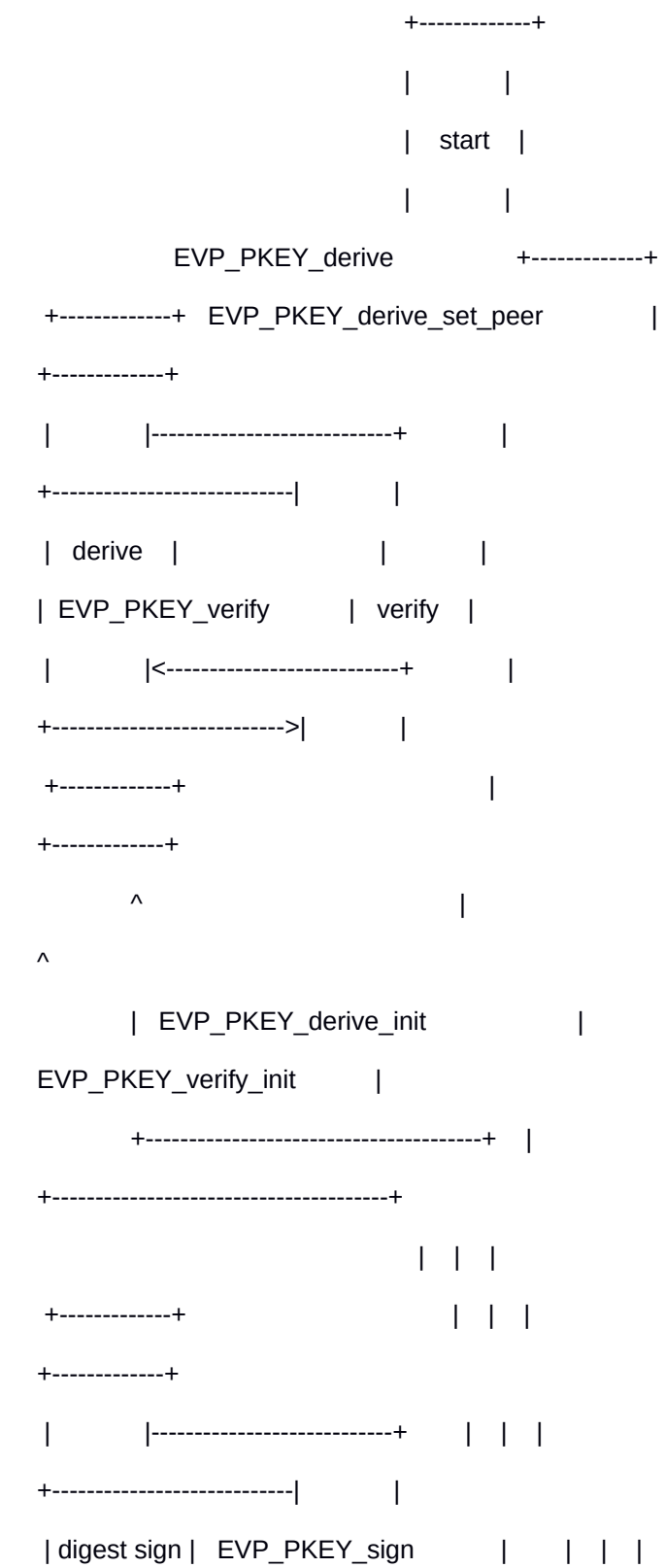
This state represents the PKEY when it is ready to recover a public key signature data.

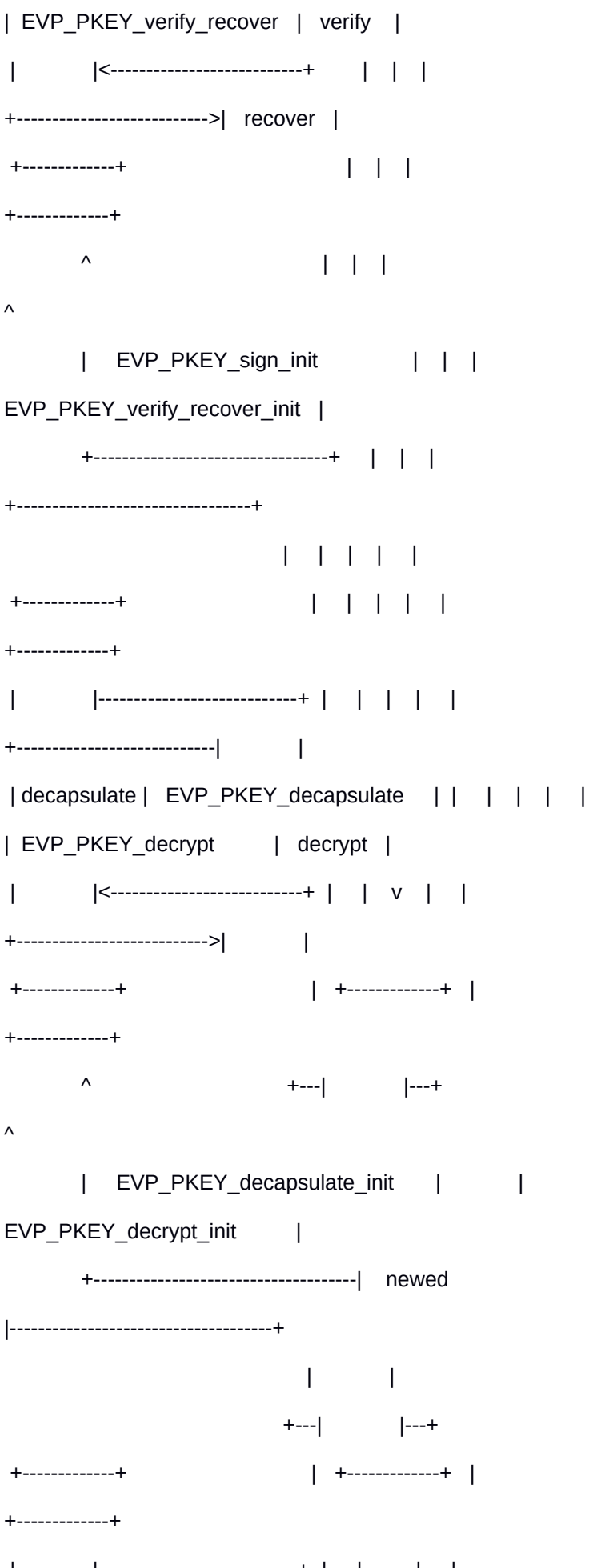
freed

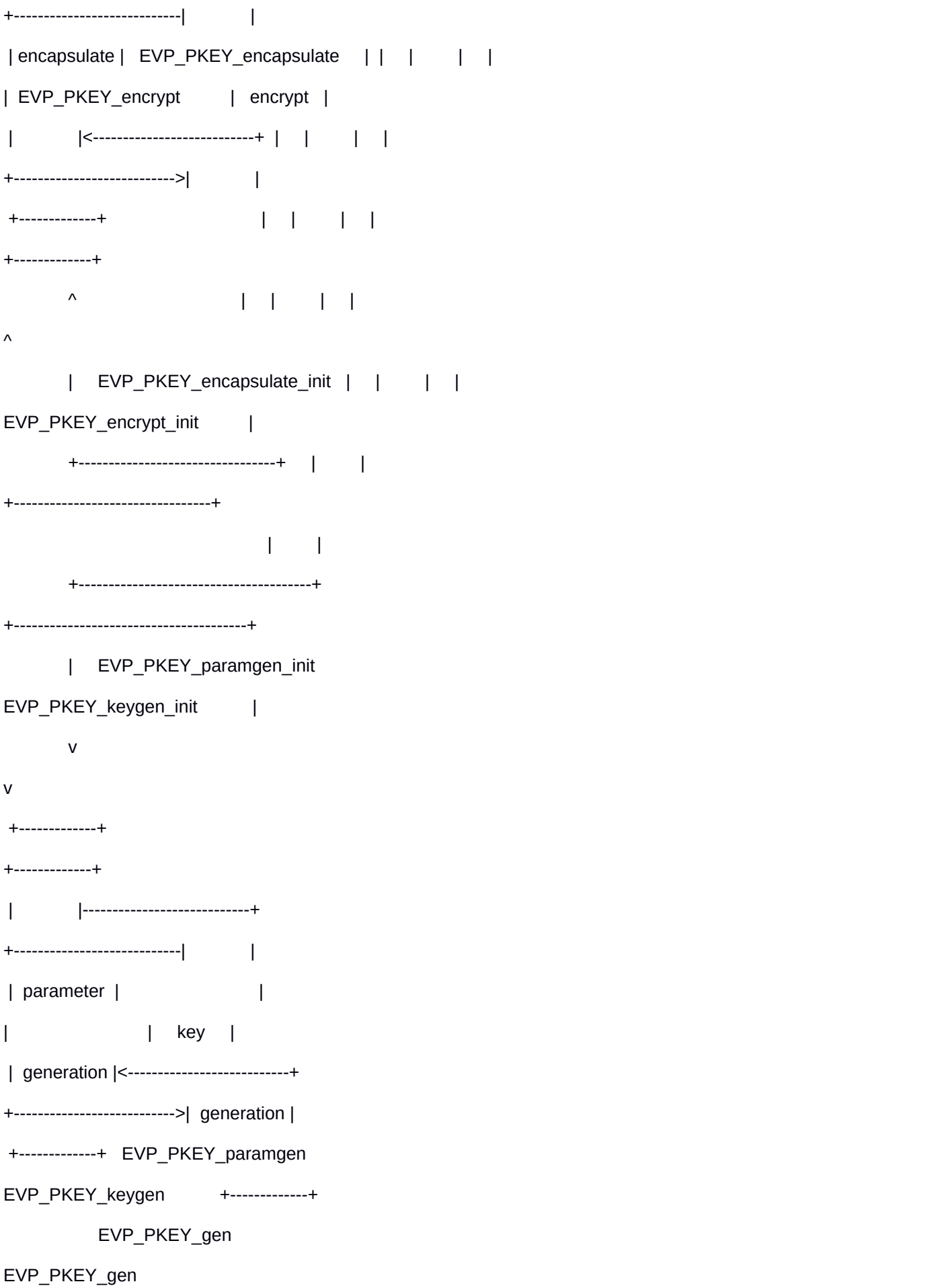
This state is entered when the PKEY is freed. It is the terminal state for all life-cycle transitions.

State Transition Diagram

The usual life-cycle of a PKEY object is illustrated:




| EVP_PKEY_verify_recover | verify |
| |<----------+ | | |
+---------->| recover |
+----------+ | | |
+----------+
^ | | |
^
| EVP_PKEY_sign_init | | |
EVP_PKEY_verify_recover_init |
+----------+ | | |
+----------+
| | | | |
+----------+ | | | | |
+----------+
| |----------+ | | | | |
+----------|
decapsulate	EVP_PKEY_decapsulate					
EVP_PKEY_decrypt	decrypt
	<----------+		v	
+----------->| |
+----------+ | +----------+ |
+----------+
^ +---| |---+
^
| EVP_PKEY_decapsulate_init | |
EVP_PKEY_decrypt_init |
+----------| newed
|----------+
| |
+---| |---+
+----------+ | +----------+ |
+----------+
| |----------+ | | | |


encapsulate
EVP_PKEY_encapsulate
EVP_PKEY_encrypt
encrypt
EVP_PKEY_encapsulate_init
EVP_PKEY_encrypt_init
EVP_PKEY_paramgen_init
EVP_PKEY_keygen_init
parameter
generation
key
generation
EVP_PKEY_paramgen
EVP_PKEY_keygen
EVP_PKEY_gen
EVP_PKEY_gen

+ - - - - - +

+-----------+

' ' EVP_PKEY_CTX_free |

|

' any state '------------------->|

freed |

' ' |

|

+ - - - - - +

+-----------+

Formal State Transitions

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

canonical list.

Function Call

Current State

----

start newed digest verify

verify encrypt decrypt derive encapsulate

decapsulate parameter key freed

sign

recover

generation generation

EVP_PKEY_CTX_new newed

EVP_PKEY_CTX_new_id newed

EVP_PKEY_CTX_new_from_name newed

EVP_PKEY_CTX_new_from_pkey newed

EVP_PKEY_sign_init digest digest digest

digest digest digest digest digest digest

digest digest

sign sign sign

sign sign sign sign sign sign

sign sign

EVP_PKEY_sign digest

sign

EVP_PKEY_verify_init verify verify verify

verify verify verify verify verify verify

verify verify

EVP_PKEY_verify verify

EVP_PKEY_verify_recover_init verify verify verify

verify verify verify verify verify verify

verify verify

recover recover

recover recover recover recover recover

recover recover recover recover

EVP_PKEY_verify_recover

verify

recover

EVP_PKEY_encrypt_init encrypt encrypt

encrypt encrypt encrypt encrypt encrypt

encrypt encrypt encrypt encrypt

EVP_PKEY_encrypt

encrypt

EVP_PKEY_decrypt_init decrypt decrypt

decrypt decrypt decrypt decrypt decrypt

decrypt decrypt decrypt decrypt

EVP_PKEY_decrypt

decrypt

EVP_PKEY_derive_init derive derive derive

derive derive derive derive derive derive

derive derive

EVP_PKEY_derive_set_peer

derive

EVP_PKEY_derive

derive

EVP_PKEY_encapsulate_init encapsulate encapsulate
encapsulate encapsulate encapsulate encapsulate encapsulate
encapsulate encapsulate encapsulate encapsulate
EVP_PKEY_encapsulate
encapsulate
EVP_PKEY_decapsulate_init decapsulate decapsulate
decapsulate decapsulate decapsulate decapsulate decapsulate
decapsulate decapsulate decapsulate decapsulate
EVP_PKEY_decapsulate
decapsulate
EVP_PKEY_paramgen_init parameter parameter
parameter parameter parameter parameter parameter
parameter parameter parameter parameter
generation generation
generation generation generation generation generation
generation generation generation generation
EVP_PKEY_paramgen
parameter
generation
EVP_PKEY_keygen_init key key key
key key key key key key
key key
generation generation
generation generation generation generation generation
generation generation generation generation
EVP_PKEY_keygen
key
generation
EVP_PKEY_gen
parameter key
generation
generation
EVP_PKEY_CTX_get_params newed digest verify

verify encrypt decrypt derive encapsulate

decapsulate parameter key

sign

recover

generation generation

EVP_PKEY_CTX_set_params newed digest verify

verify encrypt decrypt derive encapsulate

decapsulate parameter key

sign

recover

generation generation

EVP_PKEY_CTX_gettable_params newed digest verify

verify encrypt decrypt derive encapsulate

decapsulate parameter key

sign

recover

generation generation

EVP_PKEY_CTX_settable_params newed digest verify

verify encrypt decrypt derive encapsulate

decapsulate parameter key

sign

recover

generation generation

EVP_PKEY_CTX_free freed freed freed freed

freed freed freed freed freed freed

freed freed

NOTES

At some point the EVP layer will begin enforcing the transitions

described herein.

SEE ALSO

EVP_PKEY_new(3), EVP_PKEY_decapsulate(3), EVP_PKEY_decrypt(3),

EVP_PKEY_encapsulate(3), EVP_PKEY_encrypt(3), EVP_PKEY_derive(3),
EVP_PKEY_keygen(3), EVP_PKEY_sign(3), EVP_PKEY_verify(3),
EVP_PKEY_verify_recover(3)

HISTORY

The provider PKEY interface was introduced in OpenSSL 3.0.

COPYRIGHT