## NAME

life_cycle-kdf - The KDF algorithm life-cycle

## DESCRIPTION

All key derivation functions (KDFs) and pseudo random functions (PRFs) go through a number of stages in their life-cycle:

start This state represents the KDF/PRF before it has been allocated. It is the starting state for any life-cycle transitions.

newed

This state represents the KDF/PRF after it has been allocated.

deriving

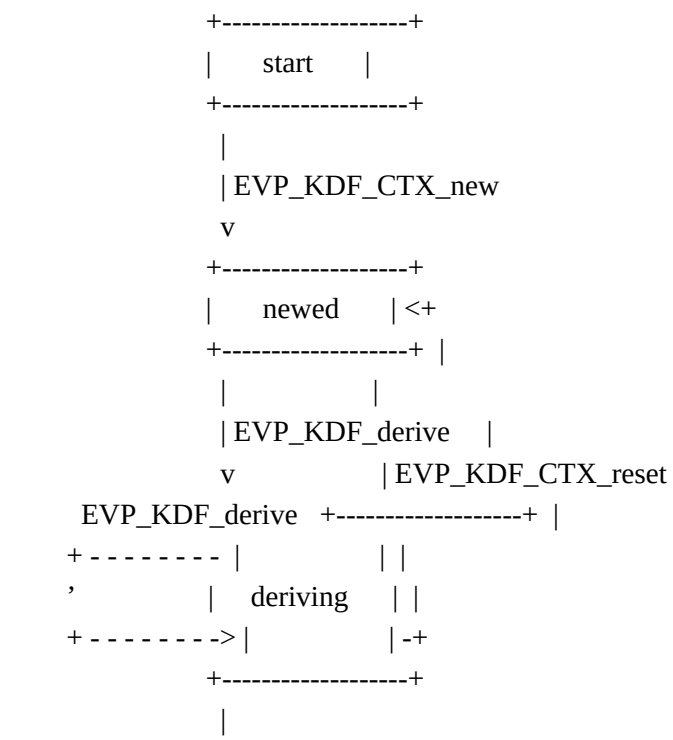
This state represents the KDF/PRF when it is set up and capable of generating output.

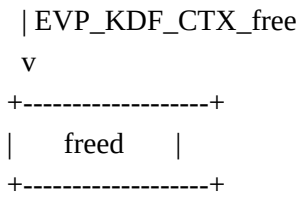
freed

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

### State Transition Diagram

The usual life-cycle of a KDF/PRF is illustrated:






### Formal State Transitions

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

| Function Call | Current State | | | |
|---|---|---|---|---|
| | start | newed | deriving | freed |
| EVP_KDF_CTX_new | | newed | | |
| EVP_KDF_derive | | | deriving | deriving |
| EVP_KDF_CTX_free | | freed | freed | freed |
| EVP_KDF_CTX_reset | | | newed | newed |
| EVP_KDF_CTX_get_params | | | newed | deriving |
| EVP_KDF_CTX_set_params | | | newed | deriving |
| EVP_KDF_CTX_gettable_params | | | newed | deriving |
| EVP_KDF_CTX_settable_params | | | newed | deriving |

## NOTES

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

## SEE ALSO

**provider-kdf**(7), **EVP_KDF**(3).

## HISTORY

The provider KDF interface was introduced in OpenSSL 3.0.

## COPYRIGHT