

class PKCS5::PBKDF2

Table of Contents

- 1 [Synopsis](#)
- 2 [Methods](#)
- 2.1 [new](#)
- 2.2 [derive](#)
- 2.3 [derive-hex](#)

```
package PKCS5 { class PBKDF2 { ... } }
```

[Synopsis](#)

```
use PKCS5::PBKDF2;

my PKCS5::PBKDF2 $p .= new;

my Str $spw = $p.derive-hex(
    Buf.new('pencil'.encode),
    Buf.new( 65, 37, 194, 71, 228, 58, 177, 233, 60, 109, 255, 118),
    4096,
);

# returns '1d96ee3a529b5a5f9e47c01f229a2cb8a6e15f7d'
```

[Methods](#)

[new](#)

Defined as

```
submethod BUILD (
    Callable :$PRF = &sha1,
    Int :$dklen,
)
```

Use

```
my PKCS5::PBKDF2 $p .= new;
```

Initialize the derivation function. The pseudorandom function PRF is set to sha1 from the

openssl::Digest by default, Other supported subs are sha256 and md5 also from that module. Md5 can also be used from Digest::MD5 but is very much slower.

Dklen is the number of bytes output from the `derive()` function. When not given, it becomes the size of the output length of the PRF.

[derive](#)

Defined as

```
method derive ( Buf $pw, Buf $salt, Int $i --> Buf )
```

Calculate the derived key given the password `$pw` and a salt `$salt`. It returns a Buf of length `dklen` specified to `new()` when initializing.

[derive-hex](#)

Defined as

```
method derive-hex ( Buf $pw, Buf $salt, Int $i --> Str )
```

Does the same as `derive()` but converts the output Buf into a hexadecimal string.