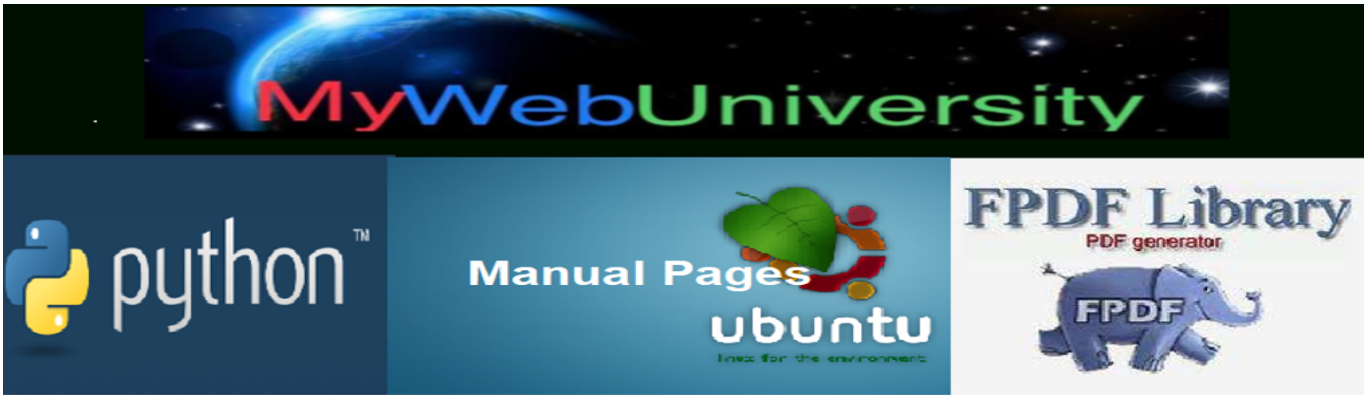

Rocky Enterprise Linux 9.2 Manual Pages on command 'Crypt::PRNG.3pm'

$ man Crypt::PRNG.3pm

NAME

Crypt::PRNG - Cryptographically secure random number generator

SYNOPSIS

```
### Functional interface:
use Crypt::PRNG qw(random_bytes random_bytes_hex random_bytes_b64 random_bytes_b64u
                   random_string random_string_from rand irand);

$octets = random_bytes(45);
$hex_string = random_bytes_hex(45);
$base64_string = random_bytes_b64(45);
$base64url_string = random_bytes_b64u(45);
$alphanumeric_string = random_string(30);
$string = random_string_from('ACGT', 64);
$floating_point_number_0_to_1 = rand;
$floating_point_number_0_to_88 = rand(88);
$unsigned_32bit_int = irand;

### OO interface:
use Crypt::PRNG;
```

```
$prng = Crypt::PRNG->new;
#or
$prng = Crypt::PRNG->new("RC4");
#or
$prng = Crypt::PRNG->new("RC4", "some data used for seeding PRNG");

$octets = $prng->bytes(45);
$hex_string = $prng->bytes_hex(45);
$base64_string = $prng->bytes_b64(45);
$base64url_string = $prng->bytes_b64u(45);
$alphanumeric_string = $prng->string(30);
$string = $prng->string_from('ACGT', 64);
$floating_point_number_0_to_1 = $prng->double;
$floating_point_number_0_to_88 = $prng->double(88);
$unsigned_32bit_int = $prng->int32;
```

DESCRIPTION

Provides an interface to the ChaCha20 based pseudo random number generator (thread-safe and fork-safe).

FUNCTIONS

random_bytes

```
$octets = random_bytes($length);
```

Returns $length random octects.

random_bytes_hex

```
$hex_string = random_bytes_hex($length);
```

Returns $length random octects encoded as hexadecimal string.

random_bytes_b64

```
$base64_string = random_bytes_b64($length);
```

Returns $length random octects Base64 encoded.

random_bytes_b64u

```
$base64url_string = random_bytes_b64u($length);
```

Returns $length random octects Base64 URL Safe (RFC 4648 section 5) encoded.

random_string_from

```
$string = random_string_from($range, $length);
#e.g.
$string = random_string_from("ABCD", 10);
```

Returns a random string made of $length chars randomly chosen from $range string.

random_string

```
$alphanumeric_string = random_string($length);
#or
$alphanumeric_string = random_string;  # default length = 20
```

Similar to random_string_from, only $range is fixed to
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789'.

rand

```
$n = rand;
#or
$n = rand($limit);
```

Returns a random floating point number from range "[0,1)" (if called without parameter) or
"[0,$limit)".

irand

```
$i = irand;
```

Returns a random unsigned 32bit integer - range "0 .. 0xFFFFFFFF".

METHODS

new

```
$prng = Crypt::PRNG->new;
#or
$prng = Crypt::PRNG->new($alg);
#or
$prng = Crypt::PRNG->new($alg, $seed);

# $alg  ... algorithm name 'Frotuna' (DEFAULT), 'RC4', 'Sober128' or 'Yarrow'
# $seed ... will be used as an initial entropy for seeding PRNG
```

If $seed is not specified the PRNG is automatically seeded with 32bytes random data taken from "/dev/random" (UNIX) or "CryptGenRandom" (Win32)

add_entropy

```
$prng->add_entropy($random_data);
#or
$prng->add_entropy();
```

If called without parameter it uses 32bytes random data taken from "/dev/random" (UNIX) or "CryptGenRandom" (Win32).

BEWARE: you probably do not need this function at all as the module does automatic seeding on initialization as well as reseeding after fork and thread creation.

bytes

```
$octets = $prng->bytes($length);
```

See random_bytes

bytes_hex

```
$hex_string = $prng->bytes_hex($length);
```

See random_bytes_hex

bytes_b64

```
$base64_string = $prng->bytes_b64($length);
```

See random_bytes_b64

bytes_b64u

```
$base64url_string = $prng->bytes_b64u($length);
```

See random_bytes_b64u

string

```
$alphanumeric_string = $prng->string($length);
#or
$alphanumeric_string = $prng->string;
```

See random_string

string_from

```
$string = $prng->string_from($range, $length);
```

See random_string_from

double

```
$n = $prng->double;
#or
$n = $prng->double($limit);
```

See rand

int32

$i = $prng->int32;

See irand

SEE ALSO

Crypt::PRNG::Fortuna, Crypt::PRNG::RC4, Crypt::PRNG::Sober128, Crypt::PRNG::Yarrow