



**Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!**

***Rocky Enterprise Linux 9.2 Manual Pages on command 'Crypt::Digest::SHA3\_512.3pm'***

***\$ man Crypt::Digest::SHA3\_512.3pm***

Crypt::Digest::SHA3\_512(3pm) User Contributed Perl Documentation Crypt::Digest::SHA3\_512(3pm)

**NAME**

Crypt::Digest::SHA3\_512 - Hash function SHA3-512 [size: 512 bits]

**SYNOPSIS**

### Functional interface:

```
use Crypt::Digest::SHA3_512 qw( sha3_512 sha3_512_hex sha3_512_b64 sha3_512_b64u
                                sha3_512_file sha3_512_file_hex sha3_512_file_b64 sha3_512_file_b64u );
```

# calculate digest from string/buffer

```
$sha3_512_raw = sha3_512('data string');
```

```
$sha3_512_hex = sha3_512_hex('data string');
```

```
$sha3_512_b64 = sha3_512_b64('data string');
```

```
$sha3_512_b64u = sha3_512_b64u('data string');
```

# calculate digest from file

```
$sha3_512_raw = sha3_512_file('filename.dat');
```

```
$sha3_512_hex = sha3_512_file_hex('filename.dat');
```

```
$sha3_512_b64 = sha3_512_file_b64('filename.dat');
```

```
$sha3_512_b64u = sha3_512_file_b64u('filename.dat');
```

# calculate digest from filehandle

```
$sha3_512_raw = sha3_512_file(*FILEHANDLE);
```

```
$sha3_512_hex = sha3_512_file_hex(*FILEHANDLE);
```

```
$sha3_512_b64 = sha3_512_file_b64(*FILEHANDLE);  
$sha3_512_b64u = sha3_512_file_b64u(*FILEHANDLE);
```

### OO interface:

```
use Crypt::Digest::SHA3_512;
```

```
$d = Crypt::Digest::SHA3_512->new;
```

```
$d->add('any data');
```

```
$d->addfile('filename.dat');
```

```
$d->addfile(*FILEHANDLE);
```

```
$result_raw = $d->digest; # raw bytes
```

```
$result_hex = $d->hexdigest; # hexadecimal form
```

```
$result_b64 = $d->b64digest; # Base64 form
```

```
$result_b64u = $d->b64udigest; # Base64 URL Safe form
```

## DESCRIPTION

Provides an interface to the SHA3\_512 digest algorithm.

## EXPORT

Nothing is exported by default.

You can export selected functions:

```
use Crypt::Digest::SHA3_512 qw(sha3_512 sha3_512_hex sha3_512_b64 sha3_512_b64u  
    sha3_512_file sha3_512_file_hex sha3_512_file_b64 sha3_512_file_b64u);
```

Or all of them at once:

```
use Crypt::Digest::SHA3_512 ':all';
```

## FUNCTIONS

sha3\_512

Logically joins all arguments into a single string, and returns its SHA3\_512 digest

encoded as a binary string.

```
$sha3_512_raw = sha3_512('data string');
```

```
#or
```

```
$sha3_512_raw = sha3_512('any data', 'more data', 'even more data');
```

#### sha3\_512\_hex

Logically joins all arguments into a single string, and returns its SHA3\_512 digest encoded as a hexadecimal string.

```
$sha3_512_hex = sha3_512_hex('data string');
```

```
#or
```

```
$sha3_512_hex = sha3_512_hex('any data', 'more data', 'even more data');
```

#### sha3\_512\_b64

Logically joins all arguments into a single string, and returns its SHA3\_512 digest encoded as a Base64 string, with trailing '=' padding.

```
$sha3_512_b64 = sha3_512_b64('data string');
```

```
#or
```

```
$sha3_512_b64 = sha3_512_b64('any data', 'more data', 'even more data');
```

#### sha3\_512\_b64u

Logically joins all arguments into a single string, and returns its SHA3\_512 digest encoded as a Base64 URL Safe string (see RFC 4648 section 5).

```
$sha3_512_b64url = sha3_512_b64u('data string');
```

```
#or
```

```
$sha3_512_b64url = sha3_512_b64u('any data', 'more data', 'even more data');
```

#### sha3\_512\_file

Reads file (defined by filename or filehandle) content, and returns its SHA3\_512 digest encoded as a binary string.

```
$sha3_512_raw = sha3_512_file('filename.dat');
```

```
#or
```

```
$sha3_512_raw = sha3_512_file(*FILEHANDLE);
```

#### sha3\_512\_file\_hex

Reads file (defined by filename or filehandle) content, and returns its SHA3\_512 digest encoded as a hexadecimal string.

```
$sha3_512_hex = sha3_512_file_hex('filename.dat');
```

```
#or
```

```
$sha3_512_hex = sha3_512_file_hex(*FILEHANDLE);
```

BEWARE: You have to make sure that the filehandle is in binary mode before you pass it as argument to the `addfile()` method.

#### sha3\_512\_file\_b64

Reads file (defined by filename or filehandle) content, and returns its SHA3\_512 digest encoded as a Base64 string, with trailing '=' padding.

```
$sha3_512_b64 = sha3_512_file_b64('filename.dat');
```

```
#or
```

```
$sha3_512_b64 = sha3_512_file_b64(*FILEHANDLE);
```

#### sha3\_512\_file\_b64u

Reads file (defined by filename or filehandle) content, and returns its SHA3\_512 digest encoded as a Base64 URL Safe string (see RFC 4648 section 5).

```
$sha3_512_b64url = sha3_512_file_b64u('filename.dat');
```

```
#or
```

```
$sha3_512_b64url = sha3_512_file_b64u(*FILEHANDLE);
```

The OO interface provides the same set of functions as Crypt::Digest.

new

```
$d = Crypt::Digest::SHA3_512->new();
```

clone

```
$d->clone();
```

reset

```
$d->reset();
```

add

```
$d->add('any data');
```

#or

```
$d->add('any data', 'more data', 'even more data');
```

addfile

```
$d->addfile('filename.dat');
```

#or

```
$d->addfile(*FILEHANDLE);
```

add\_bits

```
$d->add_bits($bit_string); # e.g. $d->add_bits("111100001010");
```

#or

```
$d->add_bits($data, $nbits); # e.g. $d->add_bits("\xF0xA0", 16);
```

hashsize

```
$d->hashsize;
```

#or

```
Crypt::Digest::SHA3_512->hashsize();
```

#or

```
Crypt::Digest::SHA3_512::hashsize();
```

digest

```
$result_raw = $d->digest();
```

hexdigest

```
$result_hex = $d->hexdigest();
```

b64digest

```
$result_b64 = $d->b64digest();
```

b64udigest

```
$result_b64url = $d->b64udigest();
```

SEE ALSO

? CryptX, Crypt::Digest

? <<https://en.wikipedia.org/wiki/SHA-3>>

perl v5.34.0

2022-02-06

Crypt::Digest::SHA3\_512(3pm)