

# Module sui::bcs

This module implements BCS (de)serialization in Move. Full specification can be found here: <https://github.com/diem/bcs>

Short summary (for Move-supported types):

Usage example:

A helper struct that saves resources on operations. For better vector performance, it stores reversed bytes of the BCS and enables use of `vector::pop_back`.

For when bytes length is less than required for deserialization.

For when the boolean value different than 0 or 1.

For when ULEB byte is out of range (or not found).

Get BCS serialized bytes for any value. Re-exports stdlib [bcs::to\\_bytes](#).

Creates a new instance of BCS wrapper that holds inversed bytes for better performance.

Unpack the [BCS](#) struct returning the leftover bytes. Useful for passing the data further after partial deserialization.

Read address from the bcs-serialized bytes.

Read a bool value from bcs-serialized bytes.

Read u8 value from bcs-serialized bytes.

Read u16 value from bcs-serialized bytes.

Read u32 value from bcs-serialized bytes.

Read u64 value from bcs-serialized bytes.

Read u128 value from bcs-serialized bytes.

Read u256 value from bcs-serialized bytes.

Read ULEB bytes expecting a vector length. Result should then be used to perform `peel_*` operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

Peel `vector<T>` from serialized bytes, where `peel: |& mut BCS | -> $T` gives the functionality of peeling each value.

Peel a vector of address from serialized bytes.

Peel a vector of address from serialized bytes.

Peel a vector of u8 (eg string) from serialized bytes.

Peel a vector< (eg vec of string) from serialized bytes.

Peel a vector of u16 from serialized bytes.

Peel a vector of u32 from serialized bytes.

Peel a vector of u64 from serialized bytes.

Peel a vector of u128 from serialized bytes.

Peel a vector of u256 from serialized bytes.

Peel enum from serialized bytes, where `$f` takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

Peel Option< address > from serialized bytes.

Peel Option from serialized bytes.

Peel Option from serialized bytes.

Peel Option from serialized bytes.

Peel Option from serialized bytes.

Peel Option from serialized bytes.

Peel Option from serialized bytes.

Peel Option from serialized bytes.

## Struct

A helper struct that saves resources on operations. For better vector performance, it stores reversed bytes of the BCS and enables use of `vector::pop_back`.

```
```bash
```

```
```
```

For when bytes length is less than required for deserialization.

```
```bash
```

```
```
```

For when the boolean value different than 0 or 1.

```
```bash
```

```
```
```

For when ULEB byte is out of range (or not found).

```
```bash
```

```
```
```

Get BCS serialized bytes for any value. Re-exports `std::lib::bcs::to_bytes`.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Creates a new instance of BCS wrapper that holds inversed bytes for better performance.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Unpack the [BCS](#) struct returning the leftover bytes. Useful for passing the data further after partial deserialization.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read address from the bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read a bool value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u8 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u16 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u32 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u64 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u128 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u256 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u8 (eg string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector> (eg vec of string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u16 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u128 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< address > from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Constants

For when bytes length is less than required for deserialization.

```
'''bash
```

```
'''
```

For when the boolean value different than 0 or 1.

```
'''bash
```

```
'''
```

For when ULEB byte is out of range (or not found).

```
'''bash
```

```
'''
```

Get BCS serialized bytes for any value. Re-exports stdlib [bcs::to\\_bytes](#) .

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Creates a new instance of BCS wrapper that holds inversed bytes for better performance.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Unpack the [BCS](#) struct returning the leftover bytes. Useful for passing the data further after partial deserialization.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read address from the bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read a bool value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u8 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u16 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```



```
```bash
```

```
```
```

Read u32 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u64 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u128 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u256 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u8 (eg string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector< (eg vec of string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u16 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u128 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< address > from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Get BCS serialized bytes for any value. Re-exports stdlib [bcs::to\\_bytes](#) .

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Creates a new instance of BCS wrapper that holds inversed bytes for better performance.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Unpack the [BCS](#) struct returning the leftover bytes. Useful for passing the data further after partial deserialization.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read address from the bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read a bool value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u8 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u16 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u32 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u64 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u128 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u256 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u8 (eg string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector> (eg vec of string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of ul6 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of ul28 from serialized bytes.

```
```bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u256 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< address > from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```



```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Creates a new instance of BCS wrapper that holds inversed bytes for better performance.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Unpack the [BCS](#) struct returning the leftover bytes. Useful for passing the data further after partial deserialization.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read address from the bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read a bool value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u8 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u16 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u32 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u64 value from bcs-serialized bytes.

```
'''bash
```

...

```bash

...

Read u128 value from bcs-serialized bytes.

```bash

...

```bash

...

Read u256 value from bcs-serialized bytes.

```bash

...

```bash

...

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

```bash

...

```bash

...

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```bash

...

```bash

...

Peel a vector of address from serialized bytes.

```bash

...

```bash

...

Peel a vector of address from serialized bytes.

```bash

...

```bash

...

Peel a vector of u8 (eg string) from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel a vector> (eg vec of string) from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel a vector of ul6 from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel a vector of ul28 from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

...

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```bash

...

```bash

...

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```bash

...

```bash

...

Peel Option< address > from serialized bytes.

```bash

...

```bash

...

Peel Option from serialized bytes.

```bash

...

```bash

...

Peel Option from serialized bytes.

```bash

...

```bash

...

Peel Option from serialized bytes.

```bash

...

```bash

...

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Unpack the [BCS](#) struct returning the leftover bytes. Useful for passing the data further after partial deserialization.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read address from the bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read a bool value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

'''

Read u8 value from bcs-serialized bytes.

'''bash

'''

'''bash

'''

'''bash

'''

'''bash

'''

Read u16 value from bcs-serialized bytes.

'''bash

'''

'''bash

'''

Read u32 value from bcs-serialized bytes.

'''bash

'''

'''bash

'''

Read u64 value from bcs-serialized bytes.

'''bash

'''

'''bash

'''

Read u128 value from bcs-serialized bytes.

'''bash

'''

'''bash

'''

Read u256 value from bcs-serialized bytes.

'''bash

'''

'''bash

```
'''
```

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of address from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of address from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u8 (eg string) from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector> (eg vec of string) from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u16 from serialized bytes.

```
'''bash
```



```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u32 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u64 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u128 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u256 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
'''bash
```

```
'''
```

```
'''bash
```

'''

Peel Option< address > from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Read address from the bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read a bool value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u8 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u16 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u32 value from bcs-serialized bytes.

```
```bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u64 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u128 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u256 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of address from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u8 (eg string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector> (eg vec of string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of ul6 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of ul28 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```

Peel a vector of u256 from serialized bytes.

```bash

```

```bash

```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```bash

```

```bash

```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```bash

```

```bash

```

Peel Option< address > from serialized bytes.

```bash

```

```bash

```

Peel Option from serialized bytes.

```bash

```

```bash

```

Peel Option from serialized bytes.

```bash

```

```bash

```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Read a bool value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u8 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

'''

'''bash

'''

'''bash

'''

Read u16 value from bcs-serialized bytes.

'''bash

'''

'''bash

'''

Read u32 value from bcs-serialized bytes.

'''bash

'''

'''bash

'''

Read u64 value from bcs-serialized bytes.

'''bash

'''

'''bash

'''

Read u128 value from bcs-serialized bytes.

'''bash

'''

'''bash

'''

Read u256 value from bcs-serialized bytes.

'''bash

'''

'''bash

'''

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

'''bash

'''



```
```bash
```

```
```
```

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u8 (eg string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector> (eg vec of string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u16 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u64 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u128 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u256 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< address > from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

```
```bash
```

```
```
```

## Function

Read u8 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u16 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u32 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u64 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u128 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u256 value from bcs-serialized bytes.

```
```bash
```

'''

'''bash

'''

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

'''bash

'''

'''bash

'''

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

'''bash

'''

'''bash

'''

Peel a vector of address from serialized bytes.

'''bash

'''

'''bash

'''

Peel a vector of address from serialized bytes.

'''bash

'''

'''bash

'''

Peel a vector of u8 (eg string) from serialized bytes.

'''bash

'''

'''bash

'''

Peel a vector> (eg vec of string) from serialized bytes.

'''bash

'''

'''bash

'''

Peel a vector of u16 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u128 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

'''

'''bash

'''

Peel Option< address > from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Macro function

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u16 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u32 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u64 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u128 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```



Read u256 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u8 (eg string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector> (eg vec of string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u16 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u128 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel Option< address > from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel Option from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel Option from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel Option from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel Option from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel Option from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Read u16 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u32 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u64 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u128 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read u256 value from bcs-serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of address from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of address from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u8 (eg string) from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector> (eg vec of string) from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u16 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u128 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

'''

'''bash

'''

Peel Option< address > from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Read u32 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u64 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u128 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u256 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

```
```bash
```

```
```
```



```
```bash
```

```
```
```

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u8 (eg string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector< (eg vec of string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u16 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u64 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u128 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u256 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< address > from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

```
```bash
```

```
```
```

## Function

Read u64 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u128 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u256 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u8 (eg string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector> (eg vec of string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of ul6 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of ul28 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
'''
```

Peel a vector of u256 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< address > from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Read u128 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read u256 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
'''
```

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of address from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of address from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u8 (eg string) from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector> (eg vec of string) from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u16 from serialized bytes.

```
'''bash
```



```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u32 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u64 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u128 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u256 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
'''bash
```

```
'''
```

```
'''bash
```

'''

Peel Option< address > from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Read u256 value from bcs-serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u8 (eg string) from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel a vector> (eg vec of string) from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel a vector of ul6 from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel a vector of ul28 from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

...

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```bash

...

```bash

...

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```bash

...

```bash

...

Peel Option< address > from serialized bytes.

```bash

...

```bash

...

Peel Option from serialized bytes.

```bash

...

```bash

...

Peel Option from serialized bytes.

```bash

...

```bash

...

Peel Option from serialized bytes.

```bash

...

```bash

...

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Read ULEB bytes expecting a vector length. Result should then be used to perform peel\_\* operation LEN times.

In BCS vector length is implemented with ULEB128; See more here: <https://en.wikipedia.org/wiki/LEB128>

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u8 (eg string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector> (eg vec of string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of ul6 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of ul28 from serialized bytes.

```
```bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u256 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< address > from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```



```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Macro function

Peel vector< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling each value.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of address from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of address from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u8 (eg string) from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector> (eg vec of string) from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u16 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u32 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u64 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u128 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< address > from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of address from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u8 (eg string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector> (eg vec of string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u16 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u128 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< address > from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Peel a vector of address from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u8 (eg string) from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector> (eg vec of string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of ul6 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of ul28 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```



'''

'''bash

'''

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

'''bash

'''

'''bash

'''

Peel Option< address > from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Peel a vector of u8 (eg string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector< (eg vec of string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u16 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
'''
```

Peel a vector of u64 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u128 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel a vector of u256 from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option< address > from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Peel a vector< (eg vec of string) from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u16 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u128 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< address > from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Peel a vector of ul6 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of ul28 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< address > from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```



Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Peel a vector of u32 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u128 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< address > from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Peel a vector of u64 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u128 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< address > from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Peel a vector of ul28 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< address > from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel Option from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel Option from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel Option from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel Option from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

## Function

Peel a vector of u256 from serialized bytes.

```
```bash
```

```
...
```

```
```bash
```

```
...
```

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< address > from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```



Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Peel enum from serialized bytes, where \$f takes a tag value and returns the corresponding enum variant. Move enums are limited to 127 variants, however the tag can be any u32 value.

Example:

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< address > from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Macro function

Peel Option< T> from serialized bytes, where peel: |& mut [BCS](#) | -> \$T gives the functionality of peeling the inner value.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option< address > from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Peel Option< address > from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

## Function

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

'''

'''bash

'''

Peel Option from serialized bytes.

'''bash

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

Peel Option from serialized bytes.

```
```bash
```

```
```
```

```
```bash
```

```
```
```

## Function

Peel Option from serialized bytes.



```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```

## Function

Peel Option from serialized bytes.

```
'''bash
```

```
'''
```

```
'''bash
```

```
'''
```