

Construct

Sticky

Version 2.10 was released in January 2020. It removes major neckpains (Embedded and EmbeddedSwitch). See Transition page.

Happy to announce collaboration with **Kaitai Struct**, a multi-platform binary parser.

Please use **github issues** to ask general questions, make feature requests (and vote for them), report issues and bugs, and to submit PRs. Feel free to request any changes that would support your project.

User Guide ¶

Introduction

Example

Development and support

Requirements

Installing

Type Hints / Type Annotations

Visual Editor

Transition to 2.8

Overall

Transition to 2.9

Overall

Transition to 2.10

Overall

The Basics

Fields

Variable-length fields

Structs

Sequences

Repeaters

Processing on-the-fly

The Basics, part 2

Integers and floats

Bytes and bits

Strings

Mappings

Processing files

Documenting fields

The Bit/Byte Duality

History

BitStruct

Important notes

Fields that work with bits

Fields that work with bytes

Fields that do both

Fields that do not work and fail

The Context

Nesting

Referring to inlined constructs

Using `this` expression

Chapters

User Guide

API Reference

External links

Code repository

Travis CI (deprecated)

Questions on Issues

Downloads

Personal sites



- Using `len_`` expression
- Using `obj_` expression
- Using `list_` expression
- Known deficiencies
- Miscellaneous
 - Special
 - Miscellaneous
 - Conditional
 - Alignment and padding
- Stream manipulation
 - Field wrappers
 - Pure side effects
 - Other fields
- Tunneling tactics
 - Obtaining raw bytes
 - Endianness
 - Working with bytes subsets
 - Working with different bytes
 - Processing data with XOR and ROL
 - Compression and checksumming
 - Encryption and authentication
- Lazy parsing
 - Lazy
 - LazyStruct
 - LazyArray
 - LazyBound
- Adapters and Validators
 - Adapting
 - Validating and filtering
- Extending Construct
 - Adapters
 - Constructs
- Debugging Construct
 - Probe
 - Debugger
- Compilation feature
 - Overall
 - Compiling schemas
 - Motivation
 - Motivation, part 2
 - Motivation, part 3
 - Comparison with Kaitai Struct

API Reference

Core API: Abstract classes

- Construct
- Subconstruct
- Adapter
- SymmetricAdapter
- Validator
- Tunnel
- Compiled

Core API: Exception types

- ConstructError()
- SizeofError()
- AdaptationError()

```
ValidationError()
StreamError()
FormatFieldError()
IntegerError()
StringError()
MappingError()
RangeError()
RepeatError()
ConstError()
IndexFieldError()
CheckError()
ExplicitError()
NamedTupleError()
TimestampError()
UnionError()
SelectError()
SwitchError()
StopFieldError()
PaddingError()
TerminatedError()
RawCopyError()
ChecksumError()
CancelParsing()
CipherError()
```

Core API: Bytes and bits

```
Bytes()
GreedyBytes()
setGlobalPrintFullStrings()
Bitwise()
Bytewise()
```

Core API: Integers and Floats

```
FormatField()
BytesInteger()
BitsInteger()
VarInt()
ZigZag()
```

Core API: Strings

```
possiblestringencodings
PaddedString()
PascalString()
CString()
GreedyString()
setGlobalPrintFullStrings()
```

Core API: Mappings

```
Flag()
Enum()
FlagsEnum()
setGlobalPrintFalseFlags()
Mapping()
```

Core API: Structs and Sequences

```
Struct()
Sequence()
AlignedStruct()
BitStruct()
```

Core API: Repeaters

```
Array()
GreedyRange()
```

RepeatUntil()

Core API: Special

Renamed()

Core API: Miscellaneous

Const()

Computed()

Index()

Rebuild()

Default()

Check()

Error()

FocusedSeq()

Pickled()

Numpy()

NamedTuple()

Timestamp()

Hex()

HexDump()

Core API: Conditional

Union()

Select()

Optional()

If()

IfThenElse()

Switch()

StopIf()

Core API: Alignment and Padding

Padding()

Padded()

Aligned()

AlignedStruct()

Core API: Streaming

Pointer()

Peek()

OffsettedEnd()

Seek()

Tell()

Pass()

Terminated()

Core API: Tunneling

RawCopy()

ByteSwapped()

BitsSwapped()

Prefixed()

PrefixedArray()

FixedSized()

NullTerminated()

NullStripped()

RestreamData()

Transformed()

Restreamed()

ProcessXor()

ProcessRotateLeft()

Checksum()

Compressed()

CompressedLZ4()

EncryptedSym()

`EncryptedSymAead()`

`Rebuffered()`

Core API: Lazy equivalents

`Lazy()`

`LazyStruct()`

`LazyArray()`

`LazyBound()`

Core API: Debugging

`Probe()`

`setGlobalPrintFullStrings()`

`setGlobalPrintFalseFlags()`

`setGlobalPrintPrivateEntries()`

`Debugger()`

Core API: Adapters and Validators

`ExprAdapter()`

`ExprSymmetricAdapter()`

`ExprValidator()`

`OneOf()`

`NoneOf()`

`Filter()`

`Slicing()`

`Indexing()`

`construct.core` – entire module

`construct.lib` – entire module

Indices and tables

[Index](#)

[Module Index](#)

[Search Page](#)