

Redbin

Table of Contents

1. Introduction	2
2. Overview	2
3. Installation	3
4. Usage	3
4.1. Basic	4
4.2. Advanced	4
4.3. Unset!	4
4.4. None!	4
4.5. Logic!	5
4.6. Block!	5
4.7. Paren!	5
4.8. String!	5
4.9. File!	6
4.10. Url!	6
4.11. Char!	6
4.12. Integer!	6
4.13. Float!	6
4.14. Context!	7
4.15. Word!	7
4.16. Set-word!	7
4.17. Lit-word!	8
4.18. Get-word!	8
4.19. Refinement!	8
4.20. Issue!	8
4.21. Native!	9
4.22. Action!	9
4.23. Op!	9
4.24. Function!	9
4.25. Path!	9
4.26. Lit-path!	10
4.27. Set-path!	10
4.28. Get-path!	10
4.29. Bitset!	10
4.30. Point!	11
4.31. Object!	11
4.32. Typeset!	11

4.33. Error!	11
4.34. Vector!	11
4.35. Pair!	12
4.36. Percent!	12
4.37. Tuple!	12
4.38. Map!	12
4.39. Binary!	12
4.40. Time!	13
4.41. Tag!	13
4.42. Email!	13
4.43. Date!	13
4.44. Reference!	13

RedbinRedREBinRedbinwordany-block!

Redbinload/binary mold/binary

- Redbin64

-
-
-

#

```
magic="REDBIN" (6), version=1 (1), flags (1), length (4), size (4)

flags (000000000000000000000000)
  bit0: 00000000
  bit1: 000000
  bit2: 00000000
  bit3-7: 0000000000

length : 0000000000000000
size   : 000000000000000000
```

```
magic="REDBIN" (6), version=1 (1), flags (1), length (4), size (4)

flags (000000000000000000000000)
  bit0: 00000000
  bit1: 000000
  bit2: 00000000
  bit3-7: 0000000000

length : 0000000000000000
size   : 000000000000000000
```

```
magic="REDBIN" (6), version=1 (1), flags (1), length (4), size (4)

flags (000000000000000000000000)
  bit0: 00000000
  bit1: 000000
  bit2: 00000000
  bit3-7: 0000000000

length : 0000000000000000
size   : 000000000000000000
```

```
magic="REDBIN" (6), version=1 (1), flags (1), length (4), size (4)

flags (000000000000000000000000)
  bit0: 00000000
  bit1: 000000
  bit2: 00000000
  bit3-7: 0000000000

length : 0000000000000000
size   : 000000000000000000
```

```
magic="REDBIN" (6), version=1 (1), flags (1), length (4), size (4)

flags (000000000000000000000000)
  bit0: 00000000
  bit1: 000000
  bit2: 00000000
  bit3-7: 0000000000

length : 0000000000000000
size   : 000000000000000000
```

```
magic="REDBIN" (6), version=1 (1), flags (1), length (4), size (4)

flags (000000000000000000000000)
  bit0: 00000000
  bit1: 000000
  bit2: 00000000
  bit3-7: 0000000000

length : 0000000000000000
size   : 000000000000000000
```

```
magic="REDBIN" (6), version=1 (1), flags (1), length (4), size (4)

flags (000000000000000000000000)
  bit0: 00000000
  bit1: 000000
  bit2: 00000000
  bit3-7: 0000000000

length : 0000000000000000
size   : 000000000000000000
```

```
magic="REDBIN" (6), version=1 (1), flags (1), length (4), size (4)

flags (000000000000000000000000)
  bit0: 00000000
  bit1: 000000
  bit2: 00000000
  bit3-7: 0000000000

length : 0000000000000000
size   : 000000000000000000
```

[illegible]

3. 〇〇〇〇〇〇〇〇

Redbin word

- 0x00000000 - 0xFFFFFFFF
- NUL(0) character

[□□□□□□□□□□□□□□□□](#) *index*

[□□□□□□□□□□□□□□□□](#) *index*

word64UTF-8

[illegible]

```
length (4), size (4), offset1 (4), offset2 (4),...
offset3 (4)
```

```
length (4), size (4), offset1 (4), offset2 (4),...
offset3 (4)
```

length [] size []

Red ID Redbin

Red block! series

4. 000000

□□□□□□□□□□32□□□□ header □□□□□□□□□□

- bit31 : new-line
- bit30 : no-values
- bit29 : stack?
- bit28 : self?
- bit27 : set?
- bit26-16 :
- bit15-8 : series
- bit7-0 :

4.1.

header (4)
n/a

header/type=0

64

4.2.

header (4), value (4)

header/type=1

4.3. Unset!

header (4)

header/type=2

4.4. None!

header (4)

header

header/type=3

4.5. Logic!

header (4), value=0|1 (4)

header

header/type=4

4.6. Block!

header (4), head (4), length (4), ...

header

header/type=5

head

header (4), head (4), length (4), ...

length

header (4), head (4), length (4), ...

4.7. Paren!

header (4), head (4), length (4), ...

header

header/type=6

block!header (4), head (4), length (4), ...

4.8. String!

header (4), head (4), length (4), data (unit*length) [, padding (1-3)]

header

header/type=7

header/unit=1|2|4

head header (4), head (4), length (4), data (unit*length) [, padding (1-3)]
unit 16777215(2^24 - 1) UCS-1 UCS-

length

4.9. File!

4.10. Url!

4.11. Char!

4.12. Integer!

4.13. Float!

```
header [padding=0 (4),] header (4), value (8)
```

```
header 00
```

```
header/type=12
```

value 64

4.14. Context!

```
header (4), length (4), symbol1 (4), symbol2 (4),..., value1 [any-type!], value2 [any-type!], ...
```

```
header 00
```

```
header/type=14
```

```
header/no-values=0|1
```

```
header/stack?=0|1
```

```
header/self?=0|1
```

function! object! Redbin word length no-values stack? self? word self

4.15. Word!

```
header (4), symbol (4), context (4), index (4)
```

```
header 00
```

```
header/type=15
```

```
header/set?=0|1
```

context

context! Redbin Redbin word context -1

set? [any-value!]

word word word

4.16. Set-word!

header (4), symbol (4), context (4), index (4)

header

header/type=16

word!

4.17. Lit-word!

header (4), symbol (4), context (4), index (4)

header

header/type=17

word!

4.18. Get-word!

header (4), symbol (4), context (4), index (4)

header

header/type=18

word!

4.19. Refinement!

header (4), symbol (4), context (4), index (4)

header

header/type=19

word!

4.20. Issue!

header (4), symbol (4)

header

header/type=20

4.21. Native!

```
header (4), ID (4), spec [block!]  
header  
  
header/type=21
```

ID `natives/table`

4.22. Action!

```
header (4), ID (4), spec [block!]  
header  
  
header/type=22
```

ID `actions/table`

4.23. Op!

```
header (4), symbol (4),  
TBD  
  
header/type=23
```

`symbol` `op!` `action` `native` `function`

4.24. Function!

```
header (4), context [context!], spec [block!], body [block!], args [block!], obj-  
ctx [context!]  
header  
  
header/type=24
```

4.25. Path!

```
header (4), head (4), length (4), ...  
header  
  
header/type=25
```

block!□□□□□□□□□□□□□□□□

4.26. Lit-path!

Default: header (4), head (4), length (4), ...

Compact: TBD

header/type=26

block!□□□□□□□□□□□□□□□□

4.27. Set-path!

```
header (4), head (4), length (4), ...
```



header/type=27

block!□□□□□□□□□□□□□□□□

4.28. Get-path!

```
header (4), head (4), length (4), ...
```

□ □ □ □ □ □ □ □

header/type=28

block!□□□□□□□□□□□□□□□□

4.29. Bitset!

```

header (4), length (4), bits (length)

```

11111111

header/type=30

length

```
0000000000000000000000000000800000000000000bitset!series000000000000000000000000
```

bits 32 NUL

4.30. Point!

header (4), x (4), y (4), z (4)

header

header/type=31

4.31. Object!

header (4), context [reference!], class-id (4), on-set-idx (4), on-set-arity (4)

header

header/type=32

on-set-idx 0 on-change* 0 on-set-arity 0

4.32. Typeset!

header (4), array1 (4), array2 (4), array3 (4)

header

header/type=33

4.33. Error!

header (4), context [reference!]

header

header/type=34

4.34. Vector!

header (4), head (4), length (4), values (unit*length)

header

header/type=35

unit 1 vector 1024 8 values values unit
1024 32 NUL

4.35. Pair!

```
header (4), x (4), y (4)
```

```
header
```

```
header/type=37
```

4.36. Percent!

```
[padding=0 (4),] header (4), value (8)
```

```
header
```

```
header/type=38
```

percent! 64 value 64

4.37. Tuple!

```
header (4), array1 (4), array2 (4), array3 (4)
```

```
header
```

```
header/type=39
```

4.38. Map!

```
header (4), length (4), ...
```

```
header
```

```
header/type=40
```

length map map length

4.39. Binary!

```
header (4), head (4), length (4), ...
```

```
header
```

```
header/type=41
```

block!

4.40. Time!

```
##### [padding=0 (4),] header (4), value (8)
```

```
#####  
```

```
header/type=43
```

time!##### **value** #####64#####

4.41. Tag!

```
##### header (4), head (4), length (4), data (unit*length)
```

```
#####  
```

```
header/type=44
```

```
header/unit=1|2|4
```

string!#####

4.42. Email!

```
##### header (4), head (4), length (4), data (unit*length)
```

```
#####  
```

```
header/type=45
```

```
header/unit=1|2|4
```

string!#####

4.43. Date!

```
##### header (4), date (4), time (8)
```

```
#####  
```

```
header/type=47
```

date# **red-date!** #####32#####integer#####time#64#####

4.44. Reference!

□ □ □ □ □ □ □ □

any-

14