

IEN: 39
Section: 2.5.3.4.2

J. Postel
ISI
17 May 1978

NSW Data Representation (NSWB8)

Data transmitted between NSW processes is encoded into a standard form. This standard form, called NSW8, is based on a few atomic data types and a list data type.

Data Structure Types and Encoding

EMPTY

TYPE (1 byte) = 1

VALUE (none) empty

BOOLEAN

TYPE (1 byte) = 2

VALUE (1 byte) boolean

FALSE=0

TRUE=1

INDEX

TYPE (1 byte) = 3

VALUE (2 bytes) index

The value represents a positive integer in the range 0 through $2^{16} - 1$.

The most significant byte is first.

INTEGER

TYPE (1 byte) = 4

VALUE (4 bytes) two's complement integer

The most significant byte is first. [If we don't use $(2^{32}-1)$ it will be easier on one's complement machines.]

INTERNET Notebook
NSW Data Representation (NSWB8)

BITSTR

TYPE (1 byte) = 5

COUNT (2 bytes)

VALUE (count bits) left adjusted in (count+7)/8 bytes

CHARSTR

TYPE (1 byte) = 6

COUNT (2 bytes)

VALUE (count bytes) ASCII text

LIST

TYPE (1 byte) = 7

COUNT (2 bytes)

Count Data Structures

PAD

TYPE (1 byte) = 9

VALUE (none)

Any PAD elements should be completely ignored. They are not to be counted (for example as elements of a LIST). The concept of a PAD element has been useful in forming data structures, especially when the structure cannot be built sequentially.

The first byte of a data structure is a type code. The following bytes depend on the type code. The type code zero is reserved. The type code 8 is reserved for possible use as REPEAT (data compression) element.

Data Structure Format

element

```
empty *-----*  
* 1 *  
*-----*  
1
```

```
boolean *-----*  
* 2 * 0 or 1 * 0 for FALSE or 1 for TRUE  
*-----*  
1 1
```

```
index *-----*  
* 3 * index * small nonnegative integer  
*-----*  
1 2
```

```
integer *-----*  
* 4 * integer * two's complement integer  
*-----*  
1 4
```

```
bitstr *-----*  
* 5 * count * bits *  
*-----*  
1 2 count (count+7)/8 bytes
```

```
charstr *-----*  
* 6 * count * text * Network ASCII  
*-----*  
1 2 count
```

```
list *-----*  
* 7 * count * count-structures *  
*-----*  
1 2
```

Examples

Empty

```
*-----*  
* 1 *  
*-----*
```

Boolean "TRUE"

```
*-----*  
* 2 * 1 *  
*-----*
```

Index "7"

```
*-----*  
* 3 * 0 * 7 *  
*-----*
```

Integer "-3"

```
*-----*  
* 4 * 255 * 255 * 255 * 253 *  
*-----*
```

Bit string "10001111101011"

```
*-----*  
* 5 * 0 * 14 * 143 * 172 *  
*-----*
```

Character string "ABCDE"

```
*-----*  
* 6 * 0 * 5 * A * B * C * D * E *  
*-----*
```

List of a character string "ABC" and a boolean "FALSE".

```
*-----*  
* 7 * 0 * 2 * 6 * 0 * 3 *  
*-----*
```

```
*-----*  
* A * B * C * 2 * 0 *  
*-----*
```