Structures

A structure is a group of variables collected together under a single name. One of these variables is called an element of the structure, in a similar way to the numbered elements of an array.

A structure element is identified by the structure name and the variable:

```
struct.element%
```

Structures are implemented by a special use of a floating point variable called an *entity*. This holds a pointer to the structure data and the prototype to interpret it.

DEF Defines a new structure prototype as a list of element names. A prototype cannot be deleted.

```
prototype=DEF(int%,str$,float)
```

Variable types are not identical to BASIC variables, but may look and act very similarly:

`integer%`	a 4-byte integer
`string$`	a string, max 255 bytes, a 5-byte SIB
`string$[n]`	a buffered string, max `n` bytes, truncated read and write as `string$`
`string!`	an indirected string in an external buffer write as `string!=buffer%` read as `string$`
`byte?`	a 1-byte integer
`float|`	a 5-byte floating point number
`substr{proto}`	a substructure to be created
`substr{}`	an indirected substructure
`array*(n)`	an array with `n+1` elements * may be `% $ ? | !`
`array#(n)`	a bit array with `n+1` elements, byte aligned
`arrary*()`	an indirected normal array * may be `% $ |`
`array(n){proto}`	an array of `n+1` substructures

NEW Creates a new structure from a prototype that is already defined.

```
struct=NEW(prototype)
```

An existing block of memory can also be used as a structure by linking it to a prototype.

DELETE Deletes a structure and all its dependencies.

**** `\` is used to assign a set of values to the elements of a structure:

```
\struct=1,"two",3.3,vec(5)
```

or on creation:

```
struct=NEW(prototype=1,"two",3.3,vec(5))
```

**** For a single element '\' is used to mean "the value of", hence:

```
\struct.int%=1
```

or

```
integer%=\struct.int%
```

The '\' syntax can be used almost anywhere that a standard variable is used, except that it cannot be for formal parameters or local variables.

Nesting A structure can be included as an element of another, provided that the prototype has been defined. The substructure is created as if `NEW` had been used, but it is then exclusive to its superstructure.

```
supertype=NEW(STRUCT=int%,substruct{prototype},str$)
```

Then:

```
superstruct=NEW(supertype)
\superstruct.int%=5
\superstruct.substruct.int%=10
\superstruct.substruct.str$="Hello world"
\superstruct.str$="Goodbye"
```