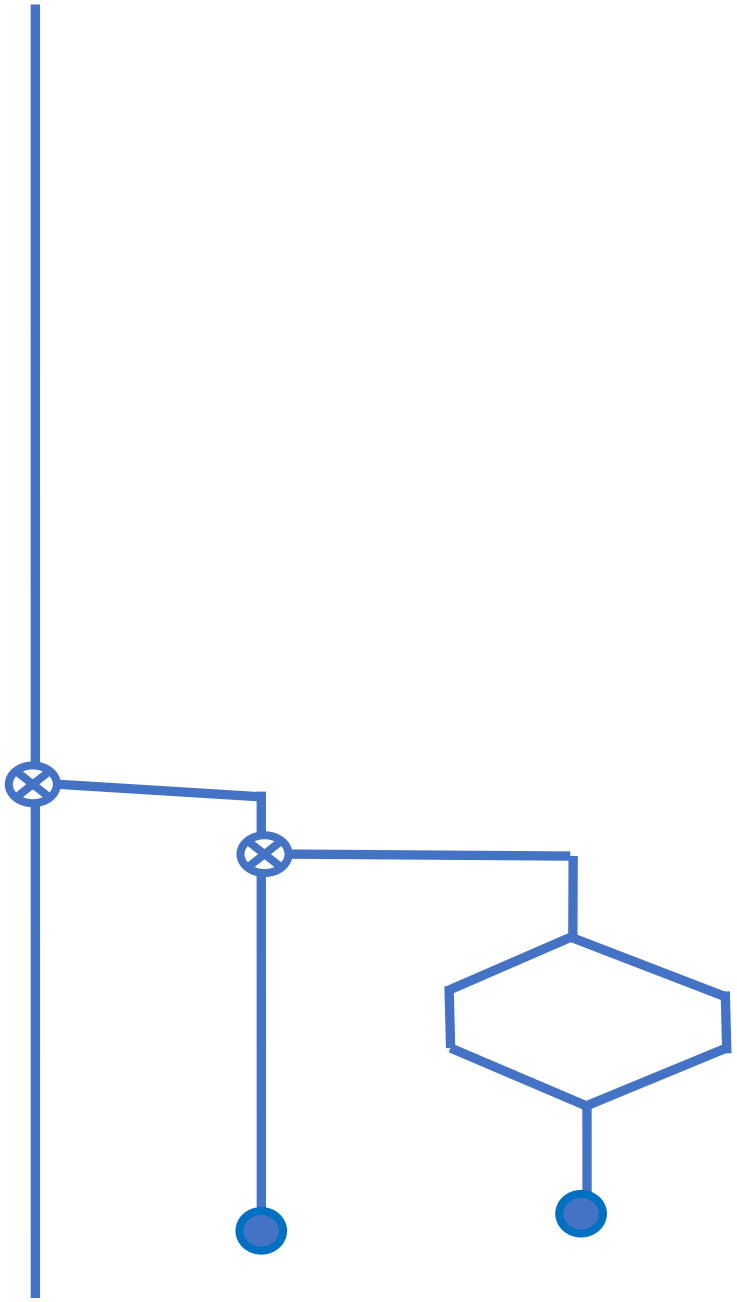

N-Way Extensible Locator

SVFIG

Sept. 27, 2025

Bill Ragsdale

Scope Of Discovery

What is a locator?

How can it be used?

What form can it take?

Use in Win32Forth: Error Messages

Locator Structure

Selects using a given locator number.

Located in-line in dictionary.

Link, one cell

Discriminator, one cell

Payload, as needed. Text, pointer, code, etc.

Payload

A string for a numeric code.

ANS Error Code. Error -4 is ”stack underflow”.

Could be a compiled cfa for a word to be executed.

A name to match an employee number.

A file name from an assigned number.

Link Generator

```
: LINK,  ( addr --- ) \ add link from the top-level pointer
      HERE   \  new link destination
      OVER   \
      @      \  address of current link
      @      \  content of current link
      ,      \  compile prior link into HERE
      SWAP   \
      ! ;    \   place address of HERE in top level pointer
```

Compiled Structure

```
449AB0    94 9A 44 00        \ link
          65 00 00 00        \ discriminator
          0D 46 6F 75 6E     \ . F o u n
          64 20 6D 65        \ d _ m e
          73 73 61 67 65     \ s s a g e
```

The Selector

```
\ produces: address & true or else false
: .search   ( n addr --- addr true | false )
begin @ 2dup cell+
      @  =                                \ a match?
      if nip 2 cells+ true exit then    \ match
      dup @ 0=
      if 2drop false exit then          \ no match
    again ;
```

Message Selection I

```
: .message ( n addr ---  )  \ display message n from list
     .search
     if  count  type
         else  ." No message found"
     then  ;
```

Message Selection II

```
Variable MyList     MyList off       \ new list

MyList    link,   999 ,    ," Found message 999“

MyList    link,   101 ,    ," Found mesage 101"
```

Message Selection III

999 Mylist .message \ See: 'Found message 999' ok

101 Mylist .message \ See: 'Found message 101' ok

111 Mylist .message \ See: 'No message found' ok

Execution Selection I

A word to select and execute code from a locator.

```
: .execute ( n addr ---  )  \ execute code by selection
     .search
     if  @  execute
         else   ." No code found"
        then  ;
```

Execution Selection II

```
: Ex_999   ." Executed 999"  ;
: Ex_101   ." Executed 101"  ;

Variable MyList  MyList off

MyList   link, 999 ,  [compile]  Ex_999
MyList   link, 101 ,  [compile]  Ex_101
```

Execution Selection III

```
999  Mylist  .execute  \ see:  'Executed 999'  ok

101  Mylist  .execute  \ see:  'Executed 101'  ok

111  Mylist  .execute  \ see:  'No code found'  ok
```

ANS Error Codes

Usage: -373 THROW_MSGS .message

```
: .message ( n addr ---  )  \ display message n from list
    begin @ 2dup cell+ @  =
       if nip 2 cells+ count type exit then
       dup @ 0=
       if 2drop ." No message found" exit then
    again ;
```

Some ANS/Win32F Error Codes

-4095 to -256 are for implementation defined errors

-255 to -1 are for ANS standard defined errors

0 signifies no exception

>0 codes for applications

```
  -4 THROW_MSGS   .message      stack underflow
 -14 THROW_MSGS   .message      is compilation only
 -16 THROW_MSGS   .message      requires a name
 -38 THROW_MSGS   .message      file not found
-280 THROW_MSGS   .message      create-file failed
-281 THROW_MSGS   .message      read-file failed  ok
```

Conclusion

The link format code is very compact.

Uses are very general.

Messages may be referred to in code much earlier than their definition.

Centralizes messages rather than them being distributed across an application.

Internal error messages easily used in applications.