

november 2025

Commacode for noForth t

Loading the assembler in order to define one or two words of your program in assembler takes relatively a lot of space.

COMMACODE produces 'comma code' for the most recently defined word and makes it easy to compile a low level forth word without the use of an active assembler.

```
: COMMACODE ( -- ) \ assembler code, not using the assembler
  created lfa> @+ swap \ doer body
  cr 2dup = if ." code " \ Normal CODE word?
  else 2dup chere within if ." create " \ Word with data?
  else over ['] create >body \ No, (sub)routine?
    <> ?abort ." routine "
  then then
  created lfa>n count hx 1F and type \ .name
  begin cr 4 0
    do dup chere < 0= \ End range?
      if drop 0 leave then
        @+ u. ." , "
        2dup = if cr ." code> " then \ adr = doer?
      loop ?dup 0=
    until drop ." end-code " cr ;
```

How to use it

1. include the assembler in noForth t
2. include the COMMACODE definition (select and copy the code and paste it into your noForth terminal)
3. define your word in assembler
4. execute the word COMMACODE

Example:

```
code 2@ ( a -- lo hi )
  day tos 4 #) ldr, \ @lo
  tos tos ) ldr, \ @hi
  day sp -) str, \ lo
next, end-code
```

The output will be:

```
code 2@ 681B685D , 600D3904 , CA10C804 , FFFF46A7 , end-code
```

◆ Paste the output into your program.

Forth addresses in COMMACODE

It is not a good idea to have absolute addresses of variables, values, subroutines, etc. in commacode. Replace them with their forth names. noForth t uses this method to handle these, start with:

```
(DATA ..data field.. DATA)
```

The W register contains the address where the data starts.

Example : An interrupt routine that uses a value

```
0 value SUM
routine TALLY ( -- )    \ 1 sum +!
  (DATA adr sum , DATA)
  day w ) ldr,    \ adr sum
  moon day ) ldr, \ sum
  moon 1 # adds,  \ 1 +
  moon day ) str, \ to sum
  lr bx, end-code commacode
```

```
routine TALLY    E001467A , 21008EAC , 682F6815 ,
                  602F3701 , FFFF4770 , end-code
```

◆ Replace the address of the value with "ADR SUM".

```
adr sum . ← 21008EAC OK.0
```

```
routine PLUS    E001467A , adr sum , 682F6815 ,
                  602F3701 , FFFF4770 , end-code
```