 Table of Contents

15. The optional Programming-Tools word set

See: 15. The optional Programming-Tools word set

15.1 Introduction

This optional word set contains words most often used during the development of applications.

15.2 Additional terms and notation

None.

15.3 Additional usage requirements

15.3.1 Environmental queries

Append table 15.1 to table 3.5.

See: 3.2.6 Environmental queries

Table 15.1 - Environmental query strings

String	Value data type	Constant?	Meaning
TOOLS	flag	no	programming-tools word set present
TOOLS-EXT	flag	no	programming-tools extensions word set present

15.3.2 The Forth dictionary

A program using the words CODE or ;CODE associated with assembler code has an environmental dependency on that particular instruction set and assembler notation.

Programs using the words EDITOR or ASSEMBLER require the Search Order word set or an equivalent implementation-defined capability.

See: 3.3 The Forth dictionary

15.4 Additional documentation requirements

15.4.1 System documentation

15.4.1.1 Implementation-defined options

- ending sequence for input following 15.6.2.0470 ;CODE and 15.6.2.0930 CODE;
- manner of processing input following 15.6.2.0470 ;CODE and 15.6.2.0930 CODE;
- search-order capability for 15.6.2.1300 EDITOR and 15.6.2.0740 ASSEMBLER (15.3.2 The Forth dictionary);
- source and format of display by 15.6.1.2194 SEE.

15.4.1.2 Ambiguous conditions

- deleting the compilation word-list (15.6.2.1580 FORGET);
- fewer than u+1 items on control-flow stack (15.6.2.1015 CSPICK, 15.6.2.1020 CSROLL);
- name can't be found (15.6.2.1580 FORGET);
- name not defined via 6.1.1000 CREATE (15.6.2.0470 ;CODE);
- 6.1.2033 POSTPONE applied to 15.6.2.2532 [IF];
- reaching the end of the input source before matching 15.6.2.2531 [ELSE] or 15.6.2.2533 [THEN] (15.6.2.2532 [IF]);
- removing a needed definition (15.6.2.1580 FORGET).

15.4.1.3 Other system documentation

- no additional requirements.

15.4.2 Program documentation

15.4.2.1 Environmental dependencies

- using the words 15.6.2.0470 ;CODE or 15.6.2.0930 CODE.

15.4.2.2 Other program documentation

- no additional requirements.

15.5 Compliance and labeling

15.5.1 ANS Forth systems

The phrase **Providing the Programming-Tools word set** shall be appended to the label of any Standard System that provides all of the Programming-Tools word set.

The phrase **Providing name(s) from the Programming-Tools Extensions word set** shall be appended to the label of any Standard System that provides portions of the Programming-Tools Extensions word set.

The phrase **Providing the Programming-Tools Extensions word set** shall be appended to the label of any Standard System that provides all of the Programming-Tools and Programming-Tools Extensions word sets.

15.5.2 ANS Forth programs

The phrase **Requiring the Programming-Tools word set** shall be appended to the label of Standard Programs that require the system to provide the Programming-Tools word set.

The phrase **Requiring name(s) from the Programming-Tools Extensions word set** shall be appended to the label of Standard Programs that require the system to provide portions of the Programming-Tools Extensions word set.

The phrase **Requiring the Programming-Tools Extensions word set** shall be appended to the label of Standard Programs that require the system to provide all of the Programming-Tools and Programming-Tools Extensions word sets.

15.6 Glossary

15.6.1 Programming-Tools words

15.6.1.0220 **.S**

dot-s TOOLS

```
( -- )
```

Copy and display the values currently on the data stack. The format of the display is implementation-dependent.

.S may be implemented using pictured numeric output words. Consequently, its use may corrupt the transient region identified by #>.

See: A.15.6.1.0220 .S , 3.3.3.6 Other transient regions.

15.6.1.0600 **?**
question TOOLS

```
( a-addr -- )
```

Display the value stored at a-addr.

? may be implemented using pictured numeric output words. Consequently, its use may corrupt the transient region identified by #>.

See: 3.3.3.6 Other transient regions

15.6.1.1280 **DUMP**
TOOLS

```
( addr u -- )
```

Display the contents of u consecutive addresses starting at addr. The format of the display is implementation dependent.

DUMP may be implemented using pictured numeric output words. Consequently, its use may corrupt the transient region identified by #>.

See: 3.3.3.6 Other transient regions

15.6.1.2194 **SEE**
TOOLS

```
( "<spaces>name" -- )
```

Display a human-readable representation of the named word's definition. The source of the representation (object-code decompilation, source block, etc.) and the particular form of the display is implementation defined.

SEE may be implemented using pictured numeric output words. Consequently, its use may corrupt the transient region identified by #>.

See: 3.3.3.6 Other transient regions, A.15.6.1.2194 SEE

15.6.1.2465 **WORDS**
TOOLS

```
( -- )
```

List the definition names in the first word list of the search order. The format of the display is implementation-dependent.

WORDS may be implemented using pictured numeric output words. Consequently, its use may corrupt the transient region identified by #>.

See: 3.3.3.6 Other transient regions, A.15.6.1.2465 WORDS

15.6.2 Programming-Tools extension words

15.6.2.0470 **;CODE**
semicolon-code TOOLS EXT

```
Interpretation: Interpretation semantics for this word are undefined.

Compilation: ( C: colon-sys -- )
```

Append the run-time semantics below to the current definition. End the current definition, allow it to be found in the dictionary, and enter interpretation state, consuming colon-sys.

Subsequent characters in the parse area typically represent source code in a programming language, usually some form of assembly language. Those characters are processed in an implementation-defined manner, generating the corresponding machine code. The process continues, refilling the input buffer as needed, until an implementation-defined ending sequence is processed.

```
Run-time: ( -- ) ( R: nest-sys -- )
```

Replace the execution semantics of the most recent definition with the name execution semantics given below. Return control to the calling definition specified by nest-sys. An ambiguous condition exists if the most recent definition was not defined with CREATE or a user-defined word that calls CREATE.

name Execution: (i*x -- j*x)

Perform the machine code sequence that was generated following ;CODE.

See: A.15.6.2.0470 ;CODE , 6.1.1250 DOES>

15.6.2.0702 **AHEAD**
TOOLS EXT

```
Interpretation: Interpretation semantics for this word are undefined.

Compilation: ( C: -- orig )
```

Put the location of a new unresolved forward reference orig onto the control flow stack. Append the run-time semantics given below to the current definition. The semantics are incomplete until orig is resolved (e.g., by THEN).

```
Run-time: ( -- )
```

Continue execution at the location specified by the resolution of orig.

15.6.2.0740 **ASSEMBLER**
TOOLS EXT

```
( -- )
```

Replace the first word list in the search order with the ASSEMBLER word list.

See: 16. The optional Search-Order word set

15.6.2.0830 **BYE**
TOOLS EXT

```
( -- )
```

Return control to the host operating system, if any.

15.6.2.0930 **CODE**
TOOLS EXT

```
( "<spaces>name" -- )
```

Skip leading space delimiters. Parse name delimited by a space. Create a definition for name, called a **code definition**, with the execution semantics defined below.

Subsequent characters in the parse area typically represent source code in a programming language, usually some form of assembly language. Those characters are processed in an implementation-defined manner, generating the corresponding machine code. The process continues, refilling the input buffer as needed, until an implementation-defined ending sequence is processed.

name Execution: (i*x -- j*x)

Execute the machine code sequence that was generated following CODE.

See: A.15.6.2.0930 CODE , 3.4.1 Parsing.

15.6.2.1015 **CS-PICK**
c-s-pick TOOLS EXT

Interpretation: Interpretation semantics for this word are undefined.

Execution: (C: destu ... orig0|dest0 -- destu ... orig0|dest0 destu)(S: u --)

Remove u. Copy destu to the top of the control-flow stack. An ambiguous condition exists if there are less than u+1 items, each of which shall be an orig or dest, on the control-flow stack before CS-PICK is executed.

If the control-flow stack is implemented using the data stack, u shall be the topmost item on the data stack.

See: A.15.6.2.1015 CS-PICK

15.6.2.1020 **CS-ROLL**
c-s-roll TOOLS EXT

Interpretation: Interpretation semantics for this word are undefined.

Execution: (C: origu|destu origu-1|destu-1 ... orig0|dest0 --
origu-1|destu-1 ... orig0|dest0 origu|destu)(S: u --)

Remove u. Rotate u+1 elements on top of the control-flow stack so that origu|destu is on top of the control-flow stack. An ambiguous condition exists if there are less than u+1 items, each of which shall be an orig or dest, on the control-flow stack before CS-ROLL is executed.

If the control-flow stack is implemented using the data stack, u shall be the topmost item on the data stack.

See: A.15.6.2.1020 CS-ROLL

15.6.2.1300 **EDITOR**
TOOLS EXT

(--)

Replace the first word list in the search order with the EDITOR word list.

See: 16. The Optional Search-Order Word Set

15.6.2.1580 **FORGET**
TOOLS EXT

(**"<spaces>name"** --)

Skip leading space delimiters. Parse name delimited by a space. Find name, then delete name from the dictionary along with all words added to the dictionary after name. An ambiguous condition exists if name cannot be found.

If the Search-Order word set is present, FORGET searches the compilation word list. An ambiguous condition exists if the compilation word list is deleted.

An ambiguous condition exists if FORGET removes a word required for correct execution.

Note: This word is obsolescent and is included as a concession to existing implementations.

See: A.15.6.2.1580 FORGET , 3.4.1 Parsing, 6.2.1850 MARKER

15.6.2.2250 **STATE**
TOOLS EXT

(-- a-addr)

Extend the semantics of 6.1.2250 STATE to allow ;CODE to change the value in STATE. A program shall not directly alter the contents of STATE.

See: 3.4 The Forth text interpreter, 6.1.0450 : , 6.1.0460 ; , 6.1.0670 ABORT , 6.1.2050 QUIT , 6.1.2500 [, 6.1.2540] , 6.2.0455 :NONAME

15.6.2.2531 **[ELSE]**
bracket-else TOOLS EXT

Compilation: Perform the execution semantics given below.

Execution: (**"<spaces>name"** ... --)

Skipping leading spaces, parse and discard space-delimited words from the parse area, including nested occurrences of [IF] ... [THEN] and [IF] ... [ELSE] ... [THEN], until the word [THEN] has been parsed and discarded. If the parse area becomes exhausted, it is refilled as with REFILL. [ELSE] is an immediate word.

See: 3.4.1 Parsing, A.15.6.2.2531 [ELSE]

15.6.2.2532 **[IF]**
bracket-if TOOLS EXT

Compilation: Perform the execution semantics given below.

Execution: (flag | flag **"<spaces>name"** ... --)

If flag is true, do nothing. Otherwise, skipping leading spaces, parse and discard space-delimited words from the parse area, including nested occurrences of [IF] ... [THEN] and [IF] ... [ELSE] ... [THEN], until either the word [ELSE] or the word [THEN] has been parsed and discarded. If the parse area becomes exhausted, it is refilled as with REFILL. [IF] is an immediate word.

An ambiguous condition exists if [IF] is POSTPONEd, or if the end of the input buffer is reached and cannot be refilled before the terminating [ELSE] or [THEN] is parsed.

See: 3.4.1 Parsing, A.15.6.2.2532 [IF]

15.6.2.2533 **[THEN]**
bracket-then TOOLS EXT

Compilation: Perform the execution semantics given below.

Execution: (--)

Does nothing. [THEN] is an immediate word.

See: A.15.6.2.2533 [THEN]

 Table of Contents
 Next Section