 Table of Contents

# 16. The optional Search-Order word set

See: A.16 The optional Search-Order word set

## 16.1 Introduction

## 16.2 Additional terms and notation

**compilation word list:**
The word list into which new definition names are placed.
**search order:**
A list of word lists specifying the order in which the dictionary will be searched.

See: A.16.2 Additional terms

## 16.3 Additional usage requirements

### 16.3.1 Data types

Word list identifiers are implementation-dependent single-cell values that identify word lists.

Append table 16.1 to table 3.1.

Table 16.1 - Data types

| Symbol | Data type | Size on stack |
|---|---|---|
| wid | word list identifiers | 1 cell |

See: 3.1 Data types, 3.4.2 Finding definition names, 3.4 The Forth text interpreter.

### 16.3.2 Environmental queries

Append table 16.2 to table 3.5.

See: 3.2.6 Environmental queries

Table 16.2 - Environmental query strings

| String | Value data type | Constant? | Meaning |
|---|---|---|---|
| SEARCH-ORDER | flag | no | search-order word set present |
| SEARCH-ORDER-EXT | flag | no | search-order extensions word set present |
| WORDLISTS | n | yes | maximum number of word lists usable in the search order |

### 16.3.3 Finding definition names

When searching a word list for a definition name, the system shall search each word list from its last definition to its first. The search may encompass only a single word list, as with SEARCH-WORDLIST, or all the word lists in the search order, as with the text interpreter and FIND.

Changing the search order shall only affect the subsequent finding of definition names in the dictionary.

A system with the Search-Order word set shall allow at least eight word lists in the search order.

An ambiguous condition exists if a program changes the compilation word list during the compilation of a definition or before modification of the behavior of the most recently compiled definition with ;CODE, DOES>, or IMMEDIATE.

A program that requires more than eight word lists in the search order has an environmental dependency.

See: 3.4.2 Finding definition names, A.16.3.3 Finding definition names.

---

### 16.3.4 Contiguous regions

The regions of data space produced by the operations described in 3.3.3.2 Contiguous regions may be non-contiguous if WORDLIST is executed between allocations.

---

## 16.4 Additional documentation requirements

---

### 16.4.1 System documentation

---

#### 16.4.1.1 Implementation-defined options

- maximum number of word lists in the search order (16.3.3 Finding definition names, 16.6.1.2197 SET-ORDER);
- minimum search order (16.6.1.2197 SET-ORDER, 16.6.2.1965 ONLY).

---

#### 16.4.1.2 Ambiguous conditions

- changing the compilation word list (16.3.3 Finding definition names);
- search order empty (16.6.2.2037 PREVIOUS);
- too many word lists in search order (16.6.2.0715 ALSO).

---

#### 16.4.1.3 Other system documentation

- no additional requirements.

---

### 16.4.2 Program documentation

---

#### 16.4.2.1 Environmental dependencies

- requiring more than eight word-lists in the search order (16.3.3 Finding definition names).

---

#### 16.4.2.2 Other program documentation

- no additional requirements.

---

## 16.5 Compliance and labeling

---

### 16.5.1 ANS Forth systems

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

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

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

---

### 16.5.2 ANS Forth programs

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

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

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

---

## 16.6 Glossary

---

### 16.6.1 Search-Order words

---

16.6.1.1180 **DEFINITIONS**
SEARCH

```
( -- )
```

Make the compilation word list the same as the first word list in the search order. Specifies that the names of subsequent definitions will be placed in the compilation word list. Subsequent changes in the search order will not affect the compilation word list.

See: 16.3.3 Finding Definition Names

---

16.6.1.1550 **FIND**
SEARCH

Extend the semantics of 6.1.1550 FIND to be:

```
( c-addr -- c-addr 0  |  xt 1  |  xt -1 )
```

Find the definition named in the counted string at c-addr. If the definition is not found after searching all the word lists in the search order, return c-addr and zero. If the definition is found, return xt. If the definition is immediate, also return one (1); otherwise also return minus-one (-1). For a given string, the values returned by FIND while compiling may differ from those returned while not compiling.

See: 3.4.2 Finding definition names, 6.1.0070 ' , 6.1.2033 POSTPONE , 6.1.2510 [’] , D.6.7 Immediacy.

---

16.6.1.1595 **FORTH-WORDLIST**
SEARCH

```
( -- wid )
```

Return wid, the identifier of the word list that includes all standard words provided by the implementation. This word list is initially the compilation word list and is part of the initial search order.

---

16.6.1.1643 **GET-CURRENT**
SEARCH

```
( -- wid )
```

Return wid, the identifier of the compilation word list.

---

16.6.1.1647 **GET-ORDER**
SEARCH

```
( -- widn ... wid1 n )
```

Returns the number of word lists n in the search order and the word list identifiers widn … wid1 identifying these word lists. wid1 identifies the word list that is searched first, and widn the word list that is searched last. The search order is unaffected.

See: RFI 0002

---

16.6.1.2192 **SEARCH-WORDLIST**
SEARCH

```
( c-addr u wid -- 0 | xt 1 | xt -1 )
```

Find the definition identified by the string c-addr u in the word list identified by wid. If the definition is not found, return zero. If the definition is found, return its execution token xt and one (1) if the definition is immediate, minus-one (-1) otherwise.

See: A.16.6.1.2192 SEARCH-WORDLIST

---

16.6.1.2195 **SET-CURRENT**
SEARCH

```
( wid -- )
```

Set the compilation word list to the word list identified by wid.

---

16.6.1.2197 **SET-ORDER**
SEARCH

```
( widn ... wid1 n -- )
```

Set the search order to the word lists identified by widn … wid1. Subsequently, word list wid1 will be searched first, and word list widn searched last. If n is zero, empty the search order. If n is minus one, set the search order to the implementation-defined minimum search order. The minimum search order shall include the words FORTH-WORDLIST and SET-ORDER. A system shall allow n to be at least eight.

See: RFI 0002

---

16.6.1.2460 **WORDLIST**
SEARCH

```
( -- wid )
```

Create a new empty word list, returning its word list identifier wid. The new word list may be returned from a pool of preallocated word lists or may be dynamically allocated in data space. A system shall allow the creation of at least 8 new word lists in addition to any provided as part of the system.

---

### 16.6.2 Search-Order extension words

---

16.6.2.0715 **ALSO**
SEARCH EXT

```
( -- )
```

Transform the search order consisting of widn, … wid2, wid1 (where wid1 is searched first) into widn, … wid2, wid1, wid1. An ambiguous condition exists if there are too many word lists in the search order.

See: A.16.6.2.0715 ALSO , RFI 0002

---

16.6.2.1590 **FORTH**
SEARCH EXT

( -- )

Transform the search order consisting of widn, ... wid2, wid1 (where wid1 is searched first) into widn, ... wid2, wid FORTH-WORDLIST.

---

16.6.2.1965 **ONLY**
SEARCH EXT

( -- )

Set the search order to the implementation-defined minimum search order. The minimum search order shall include the words FORTH-WORDLIST and SET-ORDER.

See: RFI 0002

---

16.6.2.1985 **ORDER**
SEARCH EXT

( -- )

Display the word lists in the search order in their search order sequence, from first searched to last searched. Also display the word list into which new definitions will be placed. The display format is implementation dependent.

ORDER 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

---

16.6.2.2037 **PREVIOUS**
SEARCH EXT

( -- )

Transform the search order consisting of widn, ... wid2, wid1 (where wid1 is searched first) into widn, ... wid2. An ambiguous condition exists if the search order was empty before PREVIOUS was executed.

---

Table of Contents
Next Section