 Table of Contents

7. The optional Block word set

See: A.7 The optional Block word set

7.1 Introduction

7.2 Additional terms

block:
1024 characters of data on mass storage, designated by a block number.

block buffer:
A block-sized region of data space where a block is made temporarily available for use. The current block buffer is the block buffer most recently accessed by BLOCK, BUFFER, LOAD, LIST, or THRU.

See: A.7.2 Additional terms

7.3 Additional usage requirements

7.3.1 Environmental queries

Append table 7.1 to table 3.5.

See: 3.2.6 Environmental queries

Table 7.1 - Environmental Query Strings

String	Value data type	Constant?	Meaning
BLOCK	flag	no	block word set present
BLOCK-EXT	flag	no	block extensions word set present

7.3.2 Data space

A program may access memory within a valid block buffer.

See: 3.3.3 Data Space

7.3.3 Block buffer regions

The address of a block buffer returned by BLOCK or BUFFER is transient. A call to BLOCK or BUFFER may render a previously-obtained block-buffer address invalid, as may a call to any word that:

- parses:
- displays characters on the user output device, such as TYPE or EMIT;
- controls the user output device, such as CR or AT-XY;
- receives or tests for the presence of characters from the user input device such as ACCEPT or KEY;
- waits for a condition or event, such as MS or EKEY;
- manages the block buffers, such as FLUSH, SAVE-BUFFERS, or EMPTY-BUFFERS;
- performs any operation on a file or file-name directory that implies I/O, such as REFILL or any word that returns an ior;
- implicitly performs I/O, such as text interpreter nesting and un-nesting when files are being used (including

un-nesting implied by THROW).

If the input source is a block, these restrictions also apply to the address returned by SOURCE.

Block buffers are uniquely assigned to blocks.

7.3.4 Parsing

The Block word set implements an alternative input source for the text interpreter. When the input source is a block, BLK shall contain the non-zero block number and the input buffer is the 1024-character buffer containing that block.

A block is conventionally displayed as 16 lines of 64 characters.

A program may switch the input source to a block by using LOAD or THRU. Input sources may be nested using LOAD and EVALUATE in any order.

A program may reposition the parse area within a block by manipulating >IN. More extensive repositioning can be accomplished using SAVE-INPUT and RESTORE-INPUT.

See: 3.4.1 Parsing

7.3.5 Possible action on an ambiguous condition

- A system with the Block word set may set interpretation state and interpret a block.

See: 3.4.4 Possible action on an ambiguous condition

7.4 Additional documentation requirements

7.4.1 System documentation

7.4.1.1 Implementation-defined options

- the format used for display by 7.6.2.1770 LIST (if implemented);
- the length of a line affected by 7.6.2.2535 \ (if implemented).

7.4.1.2 Ambiguous conditions

- Correct block read was not possible;
- I/O exception in block transfer;
- Invalid block number (7.6.1.0800 BLOCK, 7.6.1.0820 BUFFER, 7.6.1.1790 LOAD);
- A program directly alters the contents of 7.6.1.0790 BLK;
- No current block buffer for 7.6.1.2400 UPDATE.

7.4.1.3 Other system documentation

- any restrictions a multiprogramming system places on the use of buffer addresses;
- the number of blocks available for source text and data.

7.4.2 Program documentation

- the number of blocks required by the program.

7.5 Compliance and labeling

7.5.1 ANS Forth systems

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

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

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

7.5.2 ANS Forth programs

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

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

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

7.6 Glossary

7.6.1 Block words

7.6.1.0790 **BLK**
b-l-k BLOCK

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

a-addr is the address of a cell containing zero or the number of the mass-storage block being interpreted. If BLK contains zero, the input source is not a block and can be identified by SOURCE-ID, if SOURCE-ID is available. An ambiguous condition exists if a program directly alters the contents of BLK.

See: 7.3.3 Block buffer regions

7.6.1.0800 **BLOCK**
BLOCK

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

a-addr is the address of the first character of the block buffer assigned to mass-storage block u. An ambiguous condition exists if u is not an available block number.

If block u is already in a block buffer, a-addr is the address of that block buffer.

If block u is not already in memory and there is an unassigned block buffer, transfer block u from mass storage to an unassigned block buffer. a-addr is the address of that block buffer.

If block u is not already in memory and there are no unassigned block buffers, unassign a block buffer. If the block in that buffer has been UPDATEd, transfer the block to mass storage and transfer block u from mass storage into that buffer. a-addr is the address of that block buffer.

At the conclusion of the operation, the block buffer pointed to by a-addr is the current block buffer and is assigned to u.

7.6.1.0820 **BUFFER**
BLOCK

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

a-addr is the address of the first character of the block buffer assigned to block u. The contents of the block are unspecified. An ambiguous condition exists if u is not an available block number.

If block u is already in a block buffer, a-addr is the address of that block buffer.

If block u is not already in memory and there is an unassigned buffer, a-addr is the address of that block buffer.

If block u is not already in memory and there are no unassigned block buffers, unassign a block buffer. If the block in that buffer has been UPDATEd, transfer the block to mass storage. a-addr is the address of that block buffer.

At the conclusion of the operation, the block buffer pointed to by a-addr is the current block buffer and is assigned to u.

See: 7.6.1.0800 BLOCK

7.6.1.1360E **EVALUATE**
BLOCK

Extend the semantics of 6.1.1360 EVALUATE to include:

Store zero in BLK.

7.6.1.1559 **FLUSH**
BLOCK

```
( -- )
```

Perform the function of SAVE-BUFFERS, then unassign all block buffers.

7.6.1.1790 **LOAD**
BLOCK

```
( i*x u -- j*x )
```

Save the current input-source specification. Store u in BLK (thus making block u the input source and setting the input buffer to encompass its contents), set >IN to zero, and interpret. When the parse area is exhausted, restore the prior input source specification. Other stack effects are due to the words LOADed.

An ambiguous condition exists if u is zero or is not a valid block number.

See: 3.4 The Forth text interpreter

7.6.1.2180 **SAVE-BUFFERS**
BLOCK

```
( -- )
```

Transfer the contents of each UPDATEd block buffer to mass storage. Mark all buffers as unmodified.

7.6.1.2400 **UPDATE**
BLOCK

```
( -- )
```

Mark the current block buffer as modified. An ambiguous condition exists if there is no current block buffer.

UPDATE does not immediately cause I/O.

See: 7.6.1.0800 BLOCK , 7.6.1.0820 BUFFER , 7.6.1.1559 FLUSH , 7.6.1.2180 SAVE-BUFFERS

7.6.2 Block extension words

7.6.2.1330 **EMPTY-BUFFERS**
BLOCK EXT

```
( -- )
```

Unassign all block buffers. Do not transfer the contents of any UPDATEd block buffer to mass storage.

See: 7.6.1.0800 BLOCK

7.6.2.1770 **LIST**
BLOCK EXT

```
( u -- )
```

Display block u in an implementation-defined format. Store u in SCR.

See: 7.6.1.0800 BLOCK

7.6.2.2125 **REFILL**
BLOCK EXT

```
( -- flag )
```

Extend the execution semantics of 6.2.2125 REFILL with the following:

When the input source is a block, make the next block the input source and current input buffer by adding one to the value of BLK and setting >IN to zero. Return true if the new value of BLK is a valid block number, otherwise false.

See: 11.6.2.2125 REFILL

7.6.2.2190 **SCR**
s-c-r BLOCK EXT

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

a-addr is the address of a cell containing the block number of the block most recently LISTed.

See: A.7.6.2.2190 SCR

7.6.2.2280 **THRU**
BLOCK EXT

```
( i*x u1 u2 -- j*x )
```

LOAD the mass storage blocks numbered u1 through u2 in sequence. Other stack effects are due to the words LOADed.

7.6.2.2535 \
backslash BLOCK EXT

Extend the semantics of 6.2.2535 \ to be:

```
Compilation: Perform the execution semantics given below.

Execution: ( ccc<eol>-- )
```

If BLK contains zero, parse and discard the remainder of the parse area; otherwise parse and discard the portion of the parse area corresponding to the remainder of the current line. \ is an immediate word.

Table of Contents
Next Section