

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Introduction

Quartus Forth is an on-board development environment for the [Pilot™](#), [PalmPilot™](#), [Palm III™](#), and [IBM WorkPad™](#) series of connected organizers. Quartus Forth compiles [ISO/ANSI Standard Forth](#) source directly to native 68000 machine code.

With the [registered version](#) you can create [fast, compact, stand-alone executables](#) that can be downloaded and distributed with no requirement for a run-time module.

This manual, supplemented by additional relevant information in the [Quartus Forth File Area](#), will be interactive and developmental. As additional creative insights come along, and in reaction to developers' feedback there will be revisions, refinements and additions over time.

In other words, you should come back frequently to the [Quartus Forth Home Page](#) -- both for updates to this manual and to check out the [forum discussions](#) for insights into what others are doing with Quartus Forth.

Table of Contents (manual revision 1.6.1, current as at Quartus Forth version 1.2.1)

- [License Agreement and Limited Warranty](#)
- [Typographical Conventions](#)
- [How to Register Quartus Forth](#)
- [About Forth](#)
- [About Quartus Forth](#)

Getting Started:

- [System Requirements](#)
- [Quick Start](#)
- [Setting up your System](#)
- [The Quartus Forth Console](#)

Writing Applications:

- [Structure of an Application](#)
- [Event Handling](#)
- [Creating Stand-Alone Executables](#)
- [Common Programming Errors](#)
- [Library Files](#)
- [Documentation for Selected Library Words](#)
- [Sample Applications](#)

Questions and Answers:

- [Support Contacts](#)
- [Online Resources](#)
- [Frequently Asked Questions](#)

Appendix:

- [Glossary of Terms](#)
- [Words Specific to Quartus Forth](#)
- [C Typedefs](#)
- [Pronunciation of Forth Words](#)
- [asm68k Inline Assembler](#)
- [ISO/ANSI Compliant Documentation](#)
- [Data Types](#)
- [Exception Messages](#)
- [Startup and Termination Sequences](#)
- [Floating Point Format](#)
- [Internal Structure](#)
- [Development Environment](#)
- [Testing Procedures](#)
- [History](#)

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Registering Quartus Forth

Quartus Forth is available both in evaluation and registered versions. These versions have the same [features](#) except that the registered version gives you the ability to turn your applications into stand-alone native-code executables (PRC files) that can be freely distributed without royalties.

Registered users also receive 60 days of technical support via e-mail.

To buy a [license](#) for the registered version of Quartus Forth, you have two options:



(Note that for RegSoft.com, if ordering via voice or fax, you'll need to ask for product ID #9066.)

[Write for information](#) on corporate site or global licenses.

When ordering, be sure and provide your correct HotSync user name; your registration code depends on it.

After your order has been processed, you'll be e-mailed your registration code and a registered copy of Quartus Forth. Install the registered version of Quartus Forth via the Desktop HotSync software; it will over-write the unregistered version in your PalmPilot.

To confirm that the registered version is installed, look at the Quartus Forth welcome screen -- the version number will end in an 'R' for the registered version, 'U' for the evaluation version.

To enter your registration code, create a memo in the MemoPad as follows:

```
\ startup.quartus  
  
registered xxx-yyyy-zzzz
```

where xxx-yyyy-zzzz is your registration code (the actual registration codes may vary in format). Take care when entering the code to enter i and l (the letters) and 1 (the number) correctly, and to distinguish between B and 8, and O (the letter) and 0 (zero).

If you already have a startup.quartus memo, you need only add the registered line to it.

Each time you restart Quartus Forth, your registration code will be processed. If, when using

registered functions such as [MakePRC, CopyRSRC and DelRSRC](#), you receive an 'invalid registration code' message, check that you have entered the correct code in startup.quartus.

When requesting [technical support via e-mail](#), always supply your registration code and HotSync user name.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

About Quartus Forth

Quartus Forth is an on-board [ISO/ANSI Standard Forth](#) optimizing native-code compiler for the [Pilot™](#), [PalmPilot™](#), [Palm III™](#), and [IBM WorkPad™](#) connected organizers.

"Quartus Forth"

The name "Quartus Forth" is a play on words. When Charles Moore originally created Forth, he felt it represented an evolutionary step beyond what were then called *third generation* languages (such as FORTRAN), and hence he wanted to name it 'Fourth'. Because of the 5-character limit on the length of filenames in the computer system he was using, he shortened the name to 'Forth' instead.

"Quartus" is the Latin word for "fourth".

Features

Quartus Forth gives you a full-featured Forth development environment in the palm of your hand.

The [evaluation](#) version:

- Full Standard Forth language support, with an ever-growing [source library](#) of extensions for graphics, sound, pen input, event handling, MemoPad and database access, encryption, and more
- Compilation takes place on-board the Palm™ device
- Compiles Forth programs directly into optimized 68000 machine language
- Full control over system events and PalmOS GUI resources (forms, buttons, pop-up lists, etc.)
- Seamless access to over [800 PalmOS functions](#)
- [Integrated ROM-independent single-precision floating-point](#)
- Source files can be kept in the MemoPad or in compressed Doc format
- Source files can be nested up to 64 levels deep
- On-board console/interpreter
- Automatic word-wrap and scrolling of text output
- [Integrated 68000 symbolic in-line assembler](#)
- Comes with [the full source to professional-quality example applications](#)

The [registered](#) version has all of the above features, plus:

- [Generates efficient stand-alone executables](#) (PRCs) that require no run-time library; generated apps are freely distributable without royalties

Other good things:

- Quartus Forth is written in hand-coded assembler for optimal performance
- [Each release of Quartus Forth is regression-tested with over 800 functional and unit tests](#)
- Exception handling (CATCH and THROW)
- [Full-text messages for all ISO/ANSI Standard exceptions](#)
- Generates fast, tight code
- Compilation speed approximately 1500 generated bytes/second
- Optimizations include:
 - Register caching of the top datastack item
 - Hashed dictionary lookup with full wordlist support (multiple namespaces)
 - Inlining support with automatic inlining of short code sequences and constants
 - Tail-call elimination
 - Short branch and literal optimization
- [Generated applications](#) can have up to 64K code segments and 32K data segments
- Locals support; recursion is supported up to 512 levels deep
- Protected memory model eliminates danger of accidentally over-writing other apps or data

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

ISO/ANS Compliant Documentation

1. [Standard Labeling](#)
 2. [Required Documentation](#)
 1. [Implementation-defined Options](#)
 1. [EXCEPTION word set](#)
 2. [FACILITY word set](#)
 3. [FILE word set](#)
 4. [FLOAT word set](#)
 5. [TOOLS word set](#)
 6. [SEARCH word set](#)
 2. [Ambiguous Conditions](#)
 1. [DOUBLE word set](#)
 2. [FACILITY word set](#)
 3. [FILE word set](#)
 4. [FLOAT word set](#)
 5. [TOOLS word set](#)
 6. [SEARCH word set](#)
 3. [Other Standard Documentation](#)
-

ISO/ANS Standard Labeling

Quartus Forth is an ISO/ANS Forth Standard System:

- Providing all Core Extension words except those marked obsolescent (i.e. #TIB
CONVERT EXPECT QUERY SPAN TIB);
- Providing the Double-Number word set;
- Providing the Double-Number Extensions word set;
- Providing the Exception word set;
- Providing the Exception Extensions word set;
- Providing the Facility word set;
- Providing the Facility Extensions word set;

- Providing the Floating-Point word set;
 - Providing FS. F. PRECISION SET-PRECISION F~ from the Floating-Point Extensions word set;
 - Providing the Programming-Tools word set;
 - Providing AHEAD ASSEMBLER BYE CODE [ELSE] [IF] [THEN] from Programming-Tools Extensions word set;
 - Providing the Search-Order word set;
 - Providing the Search-Order Extensions word set;
 - Providing the String word set (except -TRAILING CMOVE CMOVE>);
 - Providing the String Extensions word set;
 - Providing INCLUDED and REFILL from the File Access wordset.
-

Required ISO/ANS Standard Documentation

Implementation-defined options

Aligned address requirements (3.1.3.3 Addresses):

The DragonBall CPU architecture mandates that word- and long-size reads and writes of RAM occur on even-byte boundaries. In Forth terms, cell- or double-cell-width reads and writes must occur on even-byte boundaries.

Behavior of 6.1.1320 EMIT for non-graphic characters:

EMIT treats all characters as graphic. If the font selected has no character representation for a given value, a 'no-character' symbol (an unfilled rectangle) is displayed.

Character editing of 6.1.0695 ACCEPT and 6.2.1390 EXPECT:

Standard PalmOS editing facilities (marking text, undo, cut, copy, paste, popup keyboard) are available in ACCEPT. . Additionally, while in the interpreter, the Page Up button retrieves the previously input line. EXCEPT, being obsolescent, is not provided

Character set (3.1.2 Character types, 6.1.1320 EMIT, 6.1.1750 KEY):

Characters in the standard, large, and bold PalmPilot fonts from 0x20 to 0x7E correspond to ASCII. Other characters vary by font selection.

Character-aligned address requirements (3.1.3.3 Addresses):

All address are valid for character-width (byte) access.

Character-set-extensions matching characteristics (3.4.2 Finding definition names):

Matching is both case- and accent-insensitive (i.e. a, à, and A all match each other).

Conditions under which control characters match a space delimiter (3.4.1.1 Delimiters):

When parsing, all control characters match the space delimiter.

Format of the control-flow stack (3.2.3.2 Control-flow stack):

The data stack serves as the control-flow stack.

Conversion of digits larger than thirty-five (3.2.1.2 Digit conversion):

Input conversion will fail. Output conversion will be attempted, with digits larger than 35 mapped into the character-set range beyond Z (i.e. [, \,], and so forth).

Display after input terminates in 6.1.0695 ACCEPT and 6.2.1390 EXPECT:

The display is not changed; the cursor is positioned at the end of the input text.

Exception abort sequence (as in 6.1.0680 ABORT"):

The text message associated with the ABORT" is displayed, and execution continues as per ABORT.

Input line terminator (3.2.4.1 User input device):

At the Quartus Forth console, the input line terminator is the carriage return (Graffiti stroke upper-right to lower-left). In INCLUDED files, input is terminated via carriage-return, or end-of-file.

Maximum size of a counted string, in characters (3.1.3.4 Counted strings, 6.1.2450 WORD):

255.

Maximum size of a parsed string (3.4.1 Parsing):

255.

Maximum size of a definition name, in characters:

31.

Maximum string length for 6.1.1345 ENVIRONMENT?, in characters:

31.

Method of selecting 3.2.4.1 User input device:

Input is from the keyboard (pop-up keyboard, or Graffiti input) unless redirected via INCLUDED, INCLUDE, or NEEDS.

Method of selecting 3.2.4.2 User output device:

No facilities are currently provided for redirection of output.

Methods of dictionary compilation (3.3 The Forth dictionary):

Code is compiled to the dictionary using what the Standard calls a *type 3* mechanism (native-code with optimizations).

Number of bits in one address unit (3.1.3.3 Addresses):

8.

Number representation and arithmetic (3.2.1.1 Internal number representation):

Numbers are represented internally as two's-complement, big-endian values.

Ranges for n, +n, u, d, +d, and ud (3.1.3 Single-cell types, 3.1.4 Cell-pair types):

<i>n</i>	-32767..32767
<i>+n</i>	0..32767
<i>u</i>	0..65535
<i>d</i>	-214783648..2147483647

$+d$	0..2147483647
ud	0..4294967295

Read-only data-space regions (3.3.3 Data space):

All dataspace regions are read/write.

Size of buffer at 6.1.2450 WORD (3.3.3.6 Other transient regions):

40 characters.

Size of one cell in address units (3.1.3 Single-cell types):

2.

Size of one character in address units (3.1.2 Character types):

1.

Size of the keyboard terminal input buffer (3.3.3.5 Input buffers):

80 characters.

Size of the pictured numeric output string buffer (3.3.3.6 Other transient regions):

34 characters.

Size of the scratch area whose address is returned by 6.2.2000 PAD (3.3.3.6 Other transient regions):

Depends upon the user implementation of PAD. Recommended:

```
CREATE PAD 84 CHARS ALLOT
```

System case-sensitivity characteristics (3.4.2 Finding definition names):

The system is case-insensitive.

System prompt (3.4 The Forth text interpreter, 6.1.2050 QUIT):

ok

Type of division rounding (3.2.2.1 Integer division, 6.1.0100 */, 6.1.0110 */MOD, 6.1.0230 /, 6.1.0240 /MOD, 6.1.1890 MOD):

Symmetrical.

Values of 6.1.2250 STATE when true:

Compilation state is indicated by a 1 in STATE.

Values returned after arithmetic overflow (3.2.2.2 Other integer operations):

Overflow results in integer rollover (e.g. 65535 1+ -> 0)

Whether the current definition can be found after 6.1.1250 DOES> (6.1.0450 :):

The current definition cannot be found until ; (except via RECURSE).

Implementation-defined options: EXCEPTION word set

Values used in the system by 9.6.1.0875 CATCH and 9.6.1.2275 THROW (9.3.1 THROW values, 9.3.5 Possible actions on an ambiguous condition):

-1 -2 -4 -9 -13 -14 -16 -19 -29 -31 -38 -42 -43 -45

Implementation-defined options: FACILITY word set

Encoding of keyboard events (10.6.2.1305 EKEY):

EKEY returns a `keyDownEvent` for each keyboard event received. EKEY>CHAR interrogates the event buffer to retrieve the received character.

Duration of a system clock tick:

10 milliseconds.

Repeatability to be expected from execution of 10.6.2.1905 ms:

Repeatable to +/- 10 milliseconds.

Implementation-defined options: FILE word set

**File access methods used by 11.6.1.0765 BIN, 11.6.1.1010
CREATE-FILE, 11.6.1.1970 OPEN-FILE, 11.6.1.2054 R/O, 11.6.1.2056
R/W, and 11.6.1.2425 W/O:**

Not applicable.

File exceptions:

-38.

File line terminator (11.6.1.2090 READ-LINE):

Line-feed (0x0A) (REFILL).

File name format (11.3.1.4 File names):

Case- and accent-insensitive non-whitespace names not to exceed 31 characters in length.

Information returned by 11.6.2.1524 FILE-STATUS:

Not applicable.

**Input file state after an exception (11.6.1.1717 INCLUDE-FILE,
11.6.1.1718 INCLUDED):**

The file (and any nesting file) is closed.

lor values and meaning (11.3.1.2 I/O results):

Not applicable.

Maximum depth of file input nesting (11.3.4 Input source):

Maximum nesting depth is 64 levels deep.

Maximum size of input line (11.3.6 Parsing):

100 characters.

Methods for mapping block ranges to files (11.3.2 Blocks in files):

Not applicable.

Number of string buffers provided (11.6.1.2165 S"):

Not applicable.

Size of string buffer used by 11.6.1.2165 S":

Not applicable.

Implementation-defined options: FLOAT word set

Format and range of floating-point numbers (12.3.1 Data types, 12.6.1.2143 REPRESENT):

Motorola Fast Floating Point (FFP) format. See the [floating point specification](#) for detailed specifications.

Results of 12.6.1.2143 REPRESENT when float is out of range:

Not applicable.

Rounding or truncation of floating-point numbers (12.3.1.2 Floating-point numbers):

Floats are rounded to 24 bits of precision.

Size of floating-point stack (12.3.3 Floating-point stack):

8 floats.

Width of floating-point stack (12.3.3 Floating-point stack):

2 cells.

Implementation-defined options: TOOLS word set

Ending sequence for input following 15.6.2.0470 ;CODE and 15.6.2.0930 CODE:

END-CODE.

Manner of processing input following 15.6.2.0470 ;CODE and 15.6.2.0930 CODE:

Input is processed as either prefix or postfix 68000 assembler, as described in '[asm68k.html](#)'.

Search-order capability for 15.6.2.1300 EDITOR and 15.6.2.0740 ASSEMBLER (15.3.3 The Forth dictionary):

ASSEMBLER allows all [asm68k](#) words to be found. FORTH (SEARCH EXT) is available in the ASSEMBLER wordlist for easy return to the default search-order.

Source and format of display by 15.6.1.2194 SEE.

Not applicable.

Implementation-defined options: SEARCH word set

Maximum number of word lists in the search order (16.3.3 Finding definition names, 16.6.1.2197 SET-ORDER):

8.

Minimum search order (16.6.1.2197 SET-ORDER, 16.6.2.1965 ONLY):

FORTH-WORDLIST.

Ambiguous conditions

A name is neither a valid definition name nor a valid number during text interpretation (3.4 The Forth text interpreter):

-13 THROW. When at the Quartus Forth console, the system beeps, prints the invalid name on the screen in underlined text, followed by a question-mark, and then performs the semantics of ABORT.

A definition name exceeded the maximum length allowed (3.3.1.2 Definition names):

-19 THROW.

Addressing a region not listed in 3.3.3 Data Space:

Attempts to write to write-protected RAM are trapped by the PalmOS and result in a system exception.

Argument type incompatible with specified input parameter, e.g., passing a flag to a word expecting an n (3.1 Data types):

Not detected. False flags (0) are always legal values. True flags are legal values for data types n and u.

Attempting to obtain the execution token, (e.g., with 6.1.0070 ', 6.1.1550 FIND, etc.) of a definition with undefined interpretation semantics:

Returns the execution token representing the compilation semantics of the definition.

Dividing by zero (6.1.0100 */, 6.1.0110 */MOD, 6.1.0230 /, 6.1.0240 /MOD, 6.1.1561 FM/MOD, 6.1.1890 MOD, 6.1.2214 SM/REM, 6.1.2370 UM/MOD, 8.6.1.1820 M*/):

Trapped by the PalmOS, resulting in a system exception.

Insufficient data-stack space or return-stack space (stack overflow):

Not detected.

Insufficient space for loop-control parameters:

Not detected.

Insufficient space in the dictionary:

-8 THROW.

Interpreting a word with undefined interpretation semantics:

Permitted. An attempt will be made to execute the compilation semantics, potentially resulting in an ambiguous condition.

Modifying the contents of the input buffer or a string literal (3.3.3.4 Text-literal regions, 3.3.3.5 Input buffers):

Permitted.

Overflow of a pictured numeric output string:

Not detected.

Parsed string overflow:

Not detected.

Producing a result out of range, e.g., multiplication (using *) results in a value too big to be represented by a single-cell integer (6.1.0090 *, 6.1.0100 */ , 6.1.0110 */MOD, 6.1.0570 >NUMBER, 6.1.1561 FM/MOD, 6.1.2214 SM/REM, 6.1.2370 UM/MOD, 6.2.0970 CONVERT, 8.6.1.1820 M*/):

Not detected.

Reading from an empty data stack or return stack (stack underflow):

Stack underflow is detected during interpretation after the execution of each word. Not detected during run-time except by an explicit check.

Unexpected end of input buffer, resulting in an attempt to use a zero-length string as a name:

-16 THROW.

>IN greater than size of input buffer (3.4.1 Parsing):

Treated as though the end of the input buffer has been reached.

6.1.2120 RECURSE appears after 6.1.1250 DOES>:

Causes recursion into the word containing the DOES>.

Argument input source different than current input source for 6.2.2148 RESTORE-INPUT:

Not detected.

Data space containing definitions is de-allocated (3.3.3.2 Contiguous regions):

Not detected.

Data space read/write with incorrect alignment (3.3.3.1 Address alignment):

Trapped by the PalmOS, resulting in a system exception.

Data-space pointer not properly aligned (6.1.0150 ,, 6.1.0860 C,):

For cell-sized reads and writes, unaligned accesses are trapped by the PalmOS, resulting in a system exception.

Less than u+2 stack items (6.2.2030 PICK, 6.2.2150 ROLL):

Not detected.

Loop-control parameters not available (6.1.0140 +LOOP, 6.1.1680 I, 6.1.1730 J, 6.1.1760 LEAVE, 6.1.1800 LOOP, 6.1.2380 UNLOOP):

Not detected.

Most recent definition does not have a name (6.1.1710 IMMEDIATE):

Not detected.

Name not defined by 6.2.2405 VALUE used by 6.2.2295 TO:

Not detected.

Name not found (6.1.0070 ', 6.1.2033 POSTPONE, 6.1.2510 ['], 6.2.2530 [COMPILE]):

-13 THROW.

Parameters are not of the same type (6.1.1240 DO, 6.2.0620 ?DO, 6.2.2440 WITHIN):

Not detected.

6.1.2033 POSTPONE or 6.2.2530 [COMPILE] applied to 6.2.2295 TO:

Not detected.

String longer than a counted string returned by 6.1.2450 WORD:

Not detected.

u greater than or equal to the number of bits in a cell (6.1.1805 LSHIFT, 6.1.2162 RSHIFT):

Not detected.

Word not defined via 6.1.1000 CREATE (6.1.0550 >BODY, 6.1.1250 DOES>):

>BODY attempts to identify CREATE words, and performs -31 THROW for words failing this identification. DOES> does not attempt to identify words as having been defined by CREATE.

Words improperly used outside 6.1.0490 <# and 6.1.0040 #> (6.1.0030 #, 6.1.0050 #S, 6.1.1670 HOLD, 6.1.2210 SIGN):

Not detected.

Ambiguous conditions: DOUBLE word set

d outside range of n in 8.6.1.1140 D>S:

Not detected.

Ambiguous conditions: FACILITY word set

10.6.1.0742 AT-XY operation can't be performed on user output device.

AT-XY assumes the user output device is the Palm LCD screen.

Ambiguous conditions: FILE word set

Attempting to position a file outside its boundaries (11.6.1.2142 REPOSITION-FILE):

Not applicable.

**Attempting to read from file positions not yet written
(11.6.1.2080 READ-FILE, 11.6.1.2090 READ-LINE)**

Not applicable.

fileid is invalid (11.6.1.1717 INCLUDE-FILE):

Not applicable.

**I/O exception reading or closing fileid (11.6.1.1717
INCLUDE-FILE, 11.6.1.1718 INCLUDED):**

Not applicable.

Named file cannot be opened (11.6.1.1718 INCLUDED):

-38 THROW.

**Requesting an unmapped block number (11.3.2 Blocks in
files):**

Not applicable.

Using 11.6.1.2218 SOURCE-ID when 7.6.1.0790 BLK is not zero.

Not applicable.

Ambiguous conditions: FLOAT word set

**DF@ or DF! is used with an address that is not double-float
aligned:**

Not applicable.

F@ or F! is used with an address that is not float aligned:

If the address is not cell-aligned, the PalmOS will trap the access and a system exception will result.

Floating point result out of range (e.g., in 12.6.1.1430 F/):

-43 THROW (when detected).

SF@ or SF! is used with an address that is not single-float aligned:

Not applicable.

BASE is not decimal (12.6.1.2143 REPRESENT, 12.6.2.1427 F., 12.6.2.1513 FE., 12.6.2.1613 FS.):

BASE is not required to be decimal for floating-point conversion.

Both arguments equal zero (12.6.2.1489 FATAN2):

Not applicable.

Cosine of argument is zero for 12.6.2.1625 FTAN:

Not applicable.

d can't be precisely represented as float in 12.6.1.1130 D>F:

Not detected.

Dividing by zero (12.6.1.1430 F/):

-42 THROW.

Exponent too big for conversion (12.6.2.1203 DF!, 12.6.2.1204 DF@, 12.6.2.2202 SF!, 12.6.2.2203 SF@):

Not applicable.

Float less than one (12.6.2.1477 FACOSH):

Not applicable.

Float less than or equal to minus-one (12.6.2.1554 FLNP1):

Not applicable.

Float less than or equal to zero (12.6.2.1553 FLN, 12.6.2.1557 FLOG):

Not applicable.

Float less than zero (12.6.2.1487 FASINH, 12.6.2.1618 FSQRT):

Not detected.

Float magnitude greater than one (12.6.2.1476 FACOS, 12.6.2.1486 FASIN, 12.6.2.1491 FATANH):

Not applicable.

Integer part of float can't be represented by d in 12.6.1.1470 F>D:

-43 THROW.

String larger than pictured-numeric output area (12.6.2.1427 F., 12.6.2.1513 FE., 12.6.2.1613 FS.):

Not applicable.

Ambiguous conditions: TOOLS word set

Deleting the compilation word-list (15.6.2.1580 FORGET):

Not detected.

Fewer than u+1 items on control-flow stack (15.6.2.1015 CSPICK, 15.6.2.1020 CSROLL):

Not applicable.

name can't be found (15.6.2.1580 FORGET):

Not applicable.

Name not defined via 6.1.1000 CREATE (15.6.2.0470 ;CODE):

Not applicable.

6.1.2033 POSTPONE applied to 15.6.2.2532 [IF]:

Not detected.

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]):

Responds as though [THEN] was encountered.

Removing a needed definition (15.6.2.1580 FORGET):

Not detected.

Ambiguous conditions: SEARCH word set

Changing the compilation word list (16.3.3 Finding definition names):

Not detected.

Search order empty (16.6.2.2037 PREVIOUS):

Not detected.

Too many word lists in search order (16.6.2.0715 ALSO):

Not detected.

Other Standard Documentation

List of non-standard words using 6.2.2000 PAD (3.3.3.6 Other transient regions):

None.

Operator's terminal facilities available:

The Quartus Forth console provides a dot-addressable bitmap screen of 160 pixels across by 160 pixels down. This space holds (in the default proportional font) 14 lines of (roughly) 40 characters each. Pixels are square. Coordinate addressing is relative to (0,0) at the top-left of the screen. Clipping is handled by the system.

Screen-scrolling is handled automatically by the output subsystem in Quartus Forth.

While interpreting or during ACCEPT, the operator has access to the full range of PalmOS editing facilities.

The Quartus Forth menu provides ABORT and COLD, along with command-shortcuts for editing

facilities.

Program data space available, in address units:

Varies by PalmOS version. As a rule, at least 12K; under newer versions of the PalmOS, Quartus Forth apps have as much as 32K of dynamic RAM available.

Return stack space available, in cells:

1024 cells.

Stack space available, in cells:

1024 cells.

System dictionary space required, in address units:

The Quartus Forth kernel requires approximately 18K. User-available codespace defaults to 32000 additional bytes, to a configurable combined maximum (system+user) of 65K.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

***asm68k* Documentation**

Table of Contents

- [General Information](#)
- [Implementation Details](#)
- [Caveats](#)
- [Addressing Modes](#)
- [Assembler Directives](#)
- [Assembler Mnemonics and Addressing Modes](#)
- [Quartus Forth Registers](#)
- [Change History](#)

***asm68k* General Information**

asm68k is a full-featured symbolic assembler for the Motorola 68000 processor, ported to run in Quartus Forth. With it you can write assembler code directly for the CPU in the PalmPilot, either as wholly-assembler code words, or as inline assembler sequences embedded within Forth words. Features of *asm68k* include:

- 119 mnemonic instructions
- 13 addressing modes
- Structured conditionals
 - IF...THEN...ELSE
 - BEGIN...AGAIN
 - BEGIN...UNTIL
 - BEGIN...WHILE...REPEAT
 - FOR...NEXT
- Prefix or postfix operation

As at version 1.21, the source for the assembler is in two files, **asm68k** and **asm68k.part2**, together totalling 7191 bytes. When compiled it occupies approximately 7516 bytes of code space, and 292 bytes of data space.

To use it, [download](#) both of the files and import each of them to your pilot as memos using the Pilot Desktop software. From the Quartus Forth command line, type: **include asm68k**
<enter>

You will see: Loading asm68k v1.21...done.

Check the [File Area](#) for sample assembler code to get you started.

The original author of *asm68k* was Michael Perry of **F83** fame (an early and very popular implementation of the Forth-83 standard, together with Henry Laxen). The listing appeared in a special edition of Dr. Dobb's Journal, *the Toolbook of Forth*. His source (F83 compatible, in block format) can be found at

<ftp://ftp.taygeta.com/pub/Forth/Compilers/cross/68000/68kasm.arc>.

***asm68k* Implementation Details**

68000 mnemonics are implemented in *asm68k* as Forth words that process operand information from the data stack and compile machine instructions into code space. In *prefix* mode, the remainder of the current line is evaluated before operand information is processed.

The mnemonics are normal non-immediate words, which means that in order to assemble instructions, they must execute either in interpretation state, or from within another immediate word. Here's an example:

```
code under+ ( a b c -- a+c b )
  tos 2 sp d) add
  ] drop [
end-code
```

Note that because *drop* is a Forth word, we must switch into compilation state with *]* before it and back into interpretation state with *[* after. Here's a differently-coded but otherwise identical implementation of *under+*:

```
: under+ ( a b c -- a+c b )
  [ also assembler tos 2 sp d) add previous ]
  drop
;
```

Here we switch into interpretation state with *[* to assemble the *add* instruction, and then back into compilation state with *]* to compile *drop*.

Another word:

```
code 2- ( n -- n-2 )
  2 tos subq
end-code inline
```

***asm68k* Caveats**

When mixing Forth and assembler within a definition, bear in mind that the following words have a different meaning when found in the ASSEMBLER wordlist:

```
# 0< 0= < > A0 AGAIN AND BEGIN D0 ELSE FOR IF MOVE NEXT OR
REPEAT SWAP THEN UNTIL WHILE
```

asm68k Addressing Modes

Addressing Mode	Generation	asm68k Syntax	asm68k Example (Postfix)
<i>Register Direct Addressing</i>			
Data register direct	$ea = D_n$	D_n	d0 d1 move
Address register direct	$ea = A_n$	A_n	a3 a0 move
<i>Absolute Data Addressing</i>			
Absolute short	$ea = (\text{next word})$	n #)	3700 #) d0 move
Absolute long	$ea = (\text{next two words})$	n L#)	123456. l#) jmp
<i>Program Counter Relative Addressing</i>			
Relative with offset	$ea = PC + d_{16}$	n PCD)	d0 56 pcd) .b move
Relative with index and offset	$ea = PC + X_n + d_8$	n X_n PCDI)	100 d1 pcdi) a0 lea
<i>Register Indirect Addressing</i>			
Register indirect	$ea = (A_n)$	A_n)	d0 a0) .b move
Postincrement register indirect	$ea = (A_n), A_n := A_n + N$	A_n)+	a7)+ d7 .w move
Predecrement register indirect	$A_n := A_n - N, ea = (A_n)$	A_n -)	d0 a6 -) .w move
Register indirect with offset	$ea = (A_n) + d_{16}$	n A_n D)	15 a1 d) .b clr
Indexed register indirect with offset	$ea = (A_n) + (X_n) + d_8$	n X_n A_n DI)	16 d0 a0 di) .l neg
<i>Immediate Data Addressing</i>			
Immediate	data = next word(s)	n #	42 # d0 move
Quick immediate	Inherent data	n	7 d1 addq

asm68k Assembler Directives

Assembler Directive	Action	Example
.W	Cause subsequent generation of word-sized operations (2-byte)	d0 d1 .w move
.L	Cause subsequent generation of long-sized operations (4-byte)	24 # d5 .l add
.B	Cause subsequent generation of byte-sized operations (1-byte)	3 a1 d) .b clr
ASSEMBLER	Replace the first wordlist in the search-order with the ASSEMBLER wordlist.	
FORTH	Replace the first wordlist in the search-order with the FORTH wordlist.	
CODE <name> ... END-CODE	Creates <name>, saves the current search-order, performs .W ALSO ASSEMBLER and assembles code until END-CODE, which restores the former search-order. <name> becomes a findable Forth word that can be flagged IMMEDIATE or INLINE.	code under+ tos 2 sp d) add] drop [end-code
POSTFIX	Switch the assembler to postfix mode (the default), where operands preceed instructions.	code rot- prefix move tos d0 move sp) tos move 2 sp d) sp) move d0 2 sp d) postfix end-code
PREFIX	Switch the assembler to prefix mode, where operands follow instructions. Note that in this mode, each mnemonic/operand sequence must be on its own line.	
Structured Conditionals (8-bit displacement)		

0= 0<> 0< 0>= < >= <= >	Branch conditions; use where <condition> appears below. Note: these test the flags in the 68000 status register, and do not consume cells from the stack as do their Forth counterparts.	
<condition> IF ... ELSE ... THEN	Conditional branching, as in Forth.	0< if d0 neg else 1 d0 subq then
BEGIN ... AGAIN	A simple loop. As in Forth.	
BEGIN ... <condition> UNTIL	As in Forth.	
BEGIN ... <condition> WHILE ... REPEAT	As in Forth.	
Dn FOR ... NEXT	Loops backwards from Dn-1 to 0.	

asm68k Assembler Mnemonics and Addressing Modes

Addressing Modes	Instructions
()	RESET NOP RTE RTS
(n)	ANDI>SR EORI>SR ORI>SR STOP TRAP
(n ea)	ORI ANDI SUBI ADDI EORI CMPI ADDQ SUBQ MOVEM> MOVEM<
(n An)	LINK
(n Dn)	MOVEQ
(ea)	SET SNI SLS SCC SCS SNE SEQ SVC SVS SPL SMI SSE SLT SGT SLE JSR JMP MOVE>CCR MOVE<SCR MOVE>SCR NBCD PEA TAS CLR NOT NEG NEGX TST
(ea ea)	MOVE

(ea An)	ADDA CMPA LEA SUBA
(ea Dn)	CMP CHK DIVU DIVS MULU MULS
(ea Dn) (Dn ea)	ADD AND OR SUB
(ea Dn) (ea n #)	BCHG BCLR BSET BTST
(An)	MOVE<USP MOVE>USP UNLK
(Dn)	EXT SWAP
(Dn ea)	EOR
(Dm Dn) (m # Dn) (ea)	ASL ASR LSL LSR ROL ROR ROXL ROXR
(Da d An) (d An Da)	MOVEP
(Dn Dm) (An@- Am@)	ABCD SBCD ADDX SUBX
(An@+ Am@+)	CMPM
(Xn Xa)	EXG
(target)	BRA BSR BHI BLS BCC BCS BNE BEQ BVC BVS BPL BMI BGE BLT BGT BLE
(target Dn)	DBRA DBHI DBLS DBCC DBCS DBNE DBEQ DBVC DBVS DBPL DBMI DBGE DBLT DBGT DBLE

Quartus Forth Registers

Register	Symbolic name	Purpose within Quartus Forth
A2	CS	Codespace segment pointer
A4	SP	Data stack pointer
A5	DS	Dataspace segment pointer
A7	RP	Return stack pointer
D7	TOS	Top element of the data stack

asm68k Change History

1998.7.20	Fixed a bug relating to MULU. Released version 1.21.

1998.7.5	Fixed a bug in the original source affecting <i>imm</i> words. Fixed a bug affecting <i>isr</i> words. Released version 1.2.
1998.7.5	Removed LABEL, due to a conflict with the semantics of CODE in my port. Released version 1.1.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Library Files

The following files are provided in Memopad Archive format (Library.MPA). The latest versions can also be found in the [Quartus File Area](#) in text format.

See also the [Library Documentation](#) section of the manual.

<i>arcfour</i>	The <i>arcfour</i> symmetric encryption algorithm.
<i>asm68k</i> <i>asm68k.part2</i>	Integrated inline 68K assembler.
<i>assert</i>	Provides compile-time assertions; useful when debugging.
<i>backlight</i>	Allows program control over the backlight on the PalmPilot.
<i>bitmap</i>	Allows the definition of inline bitmap objects.
<i>calendar</i>	ANSI Forth Gregorian & ISO date routines. Algorithms are from 'Calendrical Calculations', N. Dershowitz & E.M. Reingold, 1997, ISBN 0-521-56474-3
<i>case</i>	Implementation of the Standard CORE EXT words CASE OF ENDOF ENDCASE.
<i>condthens</i>	A solution for conditional constructs (simpler than CASE . . ENDCASE). Courtesy of Wil Baden.
<i>core-ext</i>	Provides words from the Standard CORE EXT wordset that are not defined in the Quartus Forth kernel.
<i>dblmath</i>	Additional double arithmetic words.
<i>disasm</i> <i>disasm.part2</i> <i>disasm.part3</i> <i>disasm.part4</i>	68K disassembler, including an implementation of the Standard TOOLS word SEE .
<i>doc</i>	Library code allowing access to compressed Doc-format (AportisDoc, etc.) files.
<i>docinc</i>	Allows source to be read from Doc-format files.
<i>double</i>	Provides words from the Standard DOUBLE wordset that are not defined in the Quartus Forth kernel.
<i>DataMgr</i>	Library code for accessing PalmOS databases.

<i>environment</i>	Provides ENVIRONMENT? from the Standard CORE wordset.
<i>Events</i>	Defines all PalmOS event types .
<i>facility-ext</i>	Provides TIME&DATE from the Standard FACILITY EXT wordset.
<i>facility</i>	Provides KEY? and AT-XY from the Standard FACILITY wordset.
<i>file</i>	Provides S" from the Standard FILE wordset.
<i>fonts</i>	Defines all Palm OS font IDs.
<i>Fields</i>	Helper words for Palm OS GUI field manipulation.
<i>Forms</i>	Helper words for Palm OS GUI form manipulation.
<i>graphics</i>	Various graphics words (line, circle, rounded-rectangle, etc.)
<i>ids</i>	Provides an easy way to specify creator IDs and resource types.
<i>input</i>	Provides pop-up dialogs for string input.
<i>memo</i>	Allows the creation of new memos in the Memopad application.
<i>memory</i>	Provides the Standard MEMORY wordset.
<i>music</i>	A simple lexicon for producing music from the Palm internal speaker.
<i>Mem</i>	Words allowing the allocation of additional memory from the Palm OS dynamic RAM heap.
<i>random</i>	Access to the Palm OS random number generator.
<i>regs</i>	Displays the current contents of the sixteen MC68000 CPU data and address registers.
<i>resources</i>	Simple access to external GUI resources from within your applications.
<i>roman</i>	Provides Roman numeral manipulation words. Used in Duco .
<i>safe</i>	Adds extra error checking to a number of Quartus Forth words. Useful during debugging.
<i>simple-handler</i>	A menu handler used in the sample sources . No longer necessary, as menu events are now passed directly to the application via EKEY . The sample sources will soon be updated.
<i>sound</i>	Tone generation from the Palm OS internal speaker.
<i>string</i>	Additional words from the Standard STRING wordset.

<i>tester</i>	John Hayes' module allowing simple { test -> result } tests.
<i>textalign</i>	Right, left, and center-alignment of displayed text.
<i>tinylocals</i>	A non-standard but quite usable implementation of variables with local scope.
<i>tinynums</i>	Provides tiny (4x5) digits. Used by Year .
<i>toolkit</i>	A number of generally useful words.
<i>tools-ext</i>	Words from the Standard TOOLS EXT wordset.
<i>trig</i>	Provides integer SIN and COS to three decimal places of accuracy.
<i>turtle</i>	Simple LOGO-like 'turtle-graphics' words.
<i>Ver</i>	A word returning the PalmOS version.
<i>zstrings</i>	Defining word for zero-delimited strings.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Documentation for Selected Library Words

This section documents a number of the words in the Quartus Forth library.

Resources.txt

OpenResDB (*creator-id. type. -- DbOpenRef.*)

Open a resource database with the specified creator ID and type; returns a database handle.

use-resources (*creator-id. type. --*)

Opens a resource database with the specified creator-id and type; aborts with an error if the resource database isn't available. This is used to make external resources available during compilation.

backlight.txt

backlight (*bool --*)

A **true** argument will switch the backlight on; **false** switches it off.

backlight? (*-- bool*)

Returns the current on/off status of the backlight.

assert.txt

assert-level (*addr --*)

assert0((*"..." --*)

assert1((*"..." --*)

assert2((*"..." --*)

assert3((*"..." --*)

assert((*"..." --*) \ alias for **assert1**(

Used as follows:

```
needs assert
```

```
1 assert-level !
```

```
: divide ( a b - a/b )
```

```
  assert( dup 0 <> )
```

```
  / ;
```

An assertion acts as a comment unless its assertion level (0, 1, 2, or 3) is less-than or equal-to the current value of **assert-level**. This gives you the ability to compile with assertions while developing and debugging, and ship with some or all assertions not present in the code.

The code within an assertion must have no side-effects, and no net effect on the stack contents.

Events.txt

itemid (*&event. -- itemid*)

When passed the absolute address of an event structure, returns the item ID of the event (button/control number, menu item, etc.)

coords@ (*-- y x*)

Returns the pen coordinates of the last received event.

DataMgr.txt

CreateDB (*resDB? type. creator. &zname zlen --*)

Creates a new database with the name, type, and creator specified. If *resDB?* is **true**, the resulting database will be a 'resource' database. The string at *&zname* must be zero-delimited.

CloseDB (*dbr. --*)

Close the database specified by the database handle *dbr*.

OpenDB (*mode zaddr len -- dbr.*)

Open a database by name, with a specified access mode. The string at *zaddr* must be zero-delimited.

UseCard (*n --*)

Set **OpenDB** to use a different memory card. This defaults to 0 and can normally be left unchanged.

dblmath.txt

dm* (*d n -- d*n*)

Multiply *d* by *n*. Returns a double-cell value.

dm/ (*d +n -- d/n*)

Divide *d* by *+n*. Returns a double-cell value.

ud/mod (*ud1 ud2 -- udr udq*)

Divides *ud1* by *ud2*, returning the double-cell values *udr* (remainder) and *udq* (quotient).

dmod (*d +n -- +m*)

Returns a single-cell value *+m*, which is the remainder of the division of *d* by *+n*.

admod (*d +n -- +m*)

Like **dmod**, but *m=n* when result=0.

Fields.txt

FieldFocus (*fieldid --*)

Gives focus to the field specified by *fieldid*.

string>Field (*c-addr u fieldID -- Err*)

Puts the string specified by *c-addr u* into the field specified by *fieldID*. Return non-zero on failure.

Field>string (*c-addr fieldID -- len*)

Copies the contents of the field specified by *fieldID* into memory at *c-addr*. The buffer at *c-addr* must be large enough to hold the field contents. Returns *len*, the number of characters copied.

FlushQueue (*--*)

Flushes the event queue by processing any stored events. The **Fld*** PalmOS sys traps used by **Field>string**, etc. generate events which will fill the event queue unless processed. Words requiring this handling call **FlushQueue** themselves.

Forms.txt

PopupForm (*formID* --)

Display the specified form as a popup over the current form.

until-drawn (*formID* --)

Returns when the specified form has been drawn on the screen.

SetLabel (*&string. controlid* --)

Sets the control specied by *controlid* to have the label specified by the zero-delimited string at the absolute address *&string*.

SetControlValue (*flag controlid* --)

Sets the control specified by *controlid* (e.g. a checkbox or other object) to the value of *flag*.

GetControlValue (*controlid* -- *flag*)

Returns the current value of the control specified by *controlid*.

GetObjectPtr (*objectid* -- *ptr.*)

Given an object ID, returns the object pointer.

GetObjectIndex (*objectid* -- *index*)

Given an object ID, returns the object index.

toolkit.txt

place (*a1 n1 a2* --)

Places the string *a1 n1* as a counted string at *a2*. Counted strings have their length in the their first character.

place, (*a1 n1* --)

Places the string at *a1 n1* at **here**.

bounds (*a n* -- *a+n a*)

Computes the loop bounds for a string.

>lower (*C* -- *c*)

Converts a character to lower-case.

>upper (*c* -- *C*)

Converts a character to upper-case.

append (*at ut as us* -- *at ut+us*)

Appends string *s* to the end of string *t*.

array (*cells <name>* --)

Defines array words; allocates *cells* space. The defined array words have the stack picture (*index – addr*).

under+ (*n1 x1 n2* -- *n1+n2 x1*)

Adds *n2* to *n1*.

enum (*n "name"* --)

Used to create enum structures, as follows:

```
0 enum Item
  Item Apple
  Item Orange
  Item Grape

Orange . 1 ok
Apple . 0 ok
```

Grape . 2 ok

[end] (--)

Skips the rest of a file.

[defined] ("*name*" – *flag*)

Checks to see if *name* is a defined word; returns *flag* accordingly. Used with **[if]**, **[else]**, and **[then]**.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Event Handling

In a typical C program for the Palm OS, the body of the program is an *event handling loop* that processes events from the operating system and takes required actions, such as passing events to the system for default handling, initializing forms, etc.

In a Quartus Forth application, you don't need to worry about handling most events; they're handled for you whenever you use **KEY** or **EKEY**. Quartus Forth automatically handles:

- System events
- Menu events
- Form load and Form open events
- Form event dispatching

Events not automatically handled are returned on the Forth stack via **EKEY**, where you can read them and take any desired action.

Here is a code snippet showing how to detect events using **EKEY**. It will detect stylus taps on the LCD screen:

needs events

```
: go ( -- )
  begin ekey
    dup penDownEvent = if
      ." Pen Down detected" cr
    else dup penUpEvent = if
      ." Pen Up detected" cr
    then
  then drop
again ;
```

Of special interest is how *appStopEvents* are handled. Quartus Forth gives you full control over what happens when your application exits. When Quartus Forth (or a stand-alone Quartus Forth application) receives an *appStopEvent* from the system, a -257 **THROW** code is generated.

Under normal conditions, this **THROW** code will be caught by the default exception handler, and the application will simply exit via (**bye**); you may, however wish to **CATCH** this exception code and take care of any required cleanup before calling (**bye**) yourself.

This code snippet shows an example of special **BYE** handling:

```
-257 constant byeThrow
: go ( -- )
  MainForm
```



```

." Go ahead, start another app." cr
begin
  [''] key catch
  byeThrow = if
    ." Exiting in 5 seconds!"
    500. SysTaskDelay
    (bye)
  then drop
again ;

```

Table of Event Types (returned by EKEY):

Event #	Event Type
0	nilEvent
1	penDownEvent
2	penUpEvent
3	penMoveEvent
4	keyDownEvent
5	winEnterEvent
6	winExitEvent
7	ctlEnterEvent
8	ctlExitEvent
9	ctlSelectEvent
10	ctlRepeatEvent
11	lstEnterEvent
12	lstSelectEvent
13	lstExitEvent
14	popSelectEvent
15	fldEnterEvent
16	fldHeightChangedEvent
17	fldChangedEvent

18	tblEnterEvent
19	tblSelectEvent
20	daySelectEvent
21	menuEvent
22	appStopEvent
23	frmLoadEvent
24	frmOpenEvent
25	frmGotoEvent
26	frmUpdateEvent
27	frmSaveEvent
28	frmCloseEvent
29	frmTitleEnterEvent
30	frmTitleSelectEvent
31	tblExitEvent
32	sclEnterEvent
33	sclExitEvent
34	sclRepeatEvent
32767	firstUserEvent

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Exception Messages (ISO/ANSI and Quartus Forth)

Throw value	Message
-3	stack overflow
-4	stack underflow
-5	return stack overflow
-6	return stack underflow
-7	do-loops nested too deeply
-8	dictionary overflow
-9	invalid memory address
-10	division by zero
-11	result out of range
-12	argument type mismatch
-13	undefined word
-14	compile-only word
-15	invalid FORGET
-16	name argument missing
-17	numeric output overflow
-18	parsed string overflow
-19	definition name too long

-20	write to a read-only location
-21	unsupported operation
-22	control structure mismatch
-23	address alignment
-24	invalid numeric argument
-25	return stack imbalance
-26	loop parameters unavailable
-27	invalid recursion
-28	user interrupt
-29	compiler nesting error
-30	obsolescent feature
-31	>BODY used on non-CREATED word
-32	invalid name argument
-33	block read
-34	block write
-35	invalid block number
-36	invalid file position
-37	file I/O error
-38	non-existent file
-39	unexpected end of file
-40	invalid BASE for float conversion
-41	loss of precision
-42	float divide by zero

-43	float out of range
-44	float stack overflow
-45	float stack underflow
-46	float invalid argument
-47	compilation wordlist deleted
-48	invalid POSTPONE
-49	search-order overflow
-50	search-order underflow
-51	compilation wordlist changed
-52	control-flow stack overflow
-53	exception stack overflow
-54	float underflow
-55	float unidentified fault
-56	QUIT
-57	character I/O error
-58	[IF], [ELSE], or [THEN]
Quartus Forth Exceptions	
-256	Insufficient dynamic memory
-257	System exit requested
-258	Invalid registration code
-259	Registered version only

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Sample Applications

The source to two complete sample stand-alone applications are provided here:

- [Year](#)
 - Shows a full year's calendar on one screen.
- [Duco](#)
 - A four-function calculator with a difference -- this one works in Roman numerals.

See the [Quartus Forth File Area](#) for other examples.

Special note: The examples here use a [library file](#) called *simple-handler* to handle menu events. This is no longer required; the menu handling abilities have been greatly simplified. The sample code will soon be updated to reflect this. Also, more sample source is coming very soon.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual

[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

The Sources to "Year"

© Neal Bridges, November 1998. (info@quartus.net)

If you're unfamiliar with Quartus Forth, please take a look at the Quartus home page:
<<http://www.quartus.net/products/forth/>>

About This Package:

This package contains the source to a polished Quartus Forth application, complete with a menu and icon. By following the provided directions, you can use the registered version of Quartus to compile the app on-board your Palm PDA, and HotSync the resulting .PRC file down to your PC or Mac.

The sources to *Year* are provided for two purposes: first, as a demonstration of how to write a complete Quartus Forth application, with integrated PalmOS resources; and second, to demonstrate the creation of stand-alone PRCs from Quartus apps.

Beyond that, *Year* is a handy little program in its own right, and may be of interest from a coding perspective to new Forth programmers.

Files in this package:

- `ReadMe.htm` ; this file.
- [Year.txt](#) ; The Quartus Forth source to *Year* (fits in one MemoPad memo).
- [Makeyear.txt](#) ; A Quartus script to compile *Year* into a stand-alone PRC.
- `year.mpa` ; `Year.txt` and `Makeyear.txt` in MemoPad Archive format, for use with the Palm Desktop software.
- [Year.rcp](#) ; A PilRC script defining PalmOS resources used by *Year*.
- [IconBig.bmp](#) ; a 1-bpp icon bitmap.
- `YearRsrc.bat` ; A batch file to generate `YearRsrc.prc` (requires PilRC and Pila)
- `YearRsrc.prc` ; Generated from `YearRsrc.rcp`. Provided as a convenience for those of you lacking PilRC and Pila.
- `Year.prc` ; The generated stand-alone application, ready to be installed. This is the production release; it also has a small icon, and has no splash-screen.

Other Required Sources:

Here's a breakdown of what files need what other files, to help you ensure that you have all the supporting library sources installed as memos in your MemoPad: All the files listed here are available in the [Quartus File Area](#).

- *year*:
 - needs *tinynums* \ for tiny numbers, each a 3x5 bitmap
 - needs *bitmap*
 - needs *graphics* \ for pixel cursor-positioning
 - needs *DataMgr* \ for opening the resource database
 - needs *ids* \ for 4-character resource types
 - needs *Events* \ for menuEvent and menuitem
 - needs *toolkit* \ for enum
 - needs *core-ext*
- *make-year*:
 - needs *ids* \ for the creator ID

About Year and Make-Year:

Year will compile and run under the free version of Quartus. *Make-year* generates a stand-alone PRC of *Year*, and contains commands that will only function properly under a registered copy of Quartus (or the development beta, during the beta test period).

If you just want to try *Year*, skip these steps and simply use the Palm Desktop software to HotSync *Year*.PRC into your Palm device. If you want to compile it yourself, however, read on.

To compile and run Year from the sources provided:

(note: <**Return**> means a Graffiti Return -- a pen stroke in the Graffiti area from upper right to lower left)

1. Using the Palm Desktop software, add *Year*.txt and all supporting files as memos in the MemoPad on your Palm device.
2. HotSync *YearRsrc*.prc into your Palm device.
3. You'll also need Quartus itself HotSync'd in (either the free or the registered version, version 0.8.1 or later).
4. Start Quartus.
5. At the prompt, type **include year** <**Return**>
6. To run *Year*: when prompted "ok", type **go** <**Return**>

To create a stand-alone PRC from the sources provided:

1. Follow steps 1-3 above.
2. Using the Palm Desktop software, add `makeyear.txt` as a memo in the MemoPad.
3. Start the registered version of Quartus.
4. Type **include make-year** <Return>
5. To run *Year*: from the Palm Application Launcher (or equivalent), start the "Year" icon.
6. HotSync the device again; *Year*.PRC will be copied into the backup directory under your Palm Desktop user directory.

Notes:

Quartus compiles the sources to *Year* into native-code in under 8 seconds.

Make-year extracts approximately 2K of active application code from codespace, and builds it into a new .PRC file along with several required PalmOS resources (a form, menu, about box, help text, and icon). The result: *Year*.PRC, about 4.5K in size.

Year.PRC is automatically flagged for backup, and will be copied into `C:\...\username\backup\Year.PRC` at the next HotSync. *Year*.PRC is a proper stand-alone application that requires no runtime support, and can be HotSync'd (or beamed) into any another Pilot, PalmPilot, or Palm III.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998 [Neal Bridges](#). All rights reserved.

```

\ year 98.11.12 10:48 pm NAB
\ Neal Bridges
\ nbridges@interlog.com

\ Displays a 2-column year calendar.
\ Similar to Dale Gass' YearCal.
\ Up/down buttons change the year.
\ Reads 'week starts' pref
\ as set in system Preferences.

\ For: tinyemit tinytype tinycr
needs tinynums
\ For: at cursor-position
needs graphics
needs resources

\ From the PalmOS:
\ TimGetSeconds ( -- seconds. )
\ TimSecondsToDateTime
\ ( &datetime. seconds. -- )
\ WinDrawChars ( y x # c-addr. -- )
\ DaysInMonth ( y m -- days )
\ DayOfWeek ( y d m -- dow[0..6] )
\ PrefGetPreferences ( &prefs. -- )

\ For TimSecondsToDateTime:
create datetime 7 cells allot
\ Year cell within time buffer:
datetime 5 cells + constant year

: current-year ( -- year )
  datetime >abs
  TimGetSeconds
  TimSecondsToDateTime year @ ;

: week-start-preference
( -- 0=Sunday|1=Monday )
  here dup >abs PrefGetPreferences
\ 5th byte contains week start pref
\ 0=Sunday, 1=Monday:
  5 + c@ ;

\ Globals:
variable day-of-week
variable columnedge

\ Pixel coordinates & offsets:
1 constant leftmargin
2 constant topmargin
69 constant rightedge
10 constant daywidth
10 74 2constant yearpos1
16 74 2constant yearpos2

\ Keypress values:
11 constant UpButton
12 constant DownButton

: top ( -- ) topmargin 0 at ;

: column1 ( -- )
  top leftmargin columnedge ! ;

```

```

: column2 ( -- )
  column1 82 columnedge +! ;

: >twodigits ( n -- c-addr )
  0 <# # # #> drop ;

: clear-leading-zero
( c -addr -- c-addr )
  dup c@ [char] 0 = if bl over c! then ;

:noname s" JFMAMJJASOND" ;
execute drop constant monthletters

: draw-month-letter ( month -- )
  >r
  1 font drop \ use bold font
\ At right-hand edge of current line:
\ (4 pixels up to align the baselines)
  cursor-position drop 4 - \ y coord
  columnedge @ rightedge + \ x coord
  1 \ number of chars to draw
  monthletters r> 1- + >abs
  WinDrawChars ;

: draw-day ( day# -- )
  >twodigits clear-leading-zero
\ Position cursor, draw day#:
  cursor-position drop ( -- y )
  day-of-week @ daywidth *
  columnedge @ + ( -- y x )
  at 2 tinytype ;

: nextline ( -- out-of-room? )
  tinycr
\ If too close to the bottom edge:
  cursor-position drop 155 > if
\ if already in column 2, done:
  columnedge @ leftmargin <>
  if true else column2 false then
  else false then ;

: draw-days ( #days -- )
  1+ 1 do
    i draw-day
  \ Go to the next weekday:
    1 day-of-week @ + 7 mod
    dup day-of-week !
  \ If end of week, go to next line:
    0= if nextline if leave then then
  loop ;

: draw-month ( year month -- )
  dup draw-month-letter
  DaysInMonth draw-days ;

: draw-year# ( year -- )
\ Display last 2 digits of year:
  >twodigits
  dup c@ yearpos1 at tinyemit
  char+ c@ yearpos2 at tinyemit ;

1 constant January

```

```

: draw-year ( year -- )
  dup 1 January DayOfWeek
\ Correct for desired week start:
  week-start-preference
  - 7 + 7 mod day-of-week !
  dup draw-year#
  column1 \ position cursor
  13 1 do
    ( -- year ) dup i draw-month
  loop drop ( -- ) ;

: key>delta ( key -- delta )
  dup DownButton = 1 and
  swap UpButton = -1 and
  + ;

: draw-years ( year -- )
  begin
    dup draw-year
    begin ( -- year )
      key key>delta
      tuck + swap ( -- year changed? )
    until
    page
  again ;

(ID) p4ap (ID) Year use-resources

\ Resources:
1001 constant YearForm

needs simple-handler

\ Main entry point:
: go ( -- )
  YearForm ShowForm
  ['] simple-handler eventhandler !
  current-year draw-years ;

```

```
\ make-year 98.11.12 10:13 pm NAB

\ Compiles Year to a stand-alone PRC
\ that is flagged to be HotSync'd to
\ C:\...\username\backup\Year.PRC

\ The program:
needs year
\ ...leaves open its resdb with all req'd
\ resources.

\ Compile it to a stand-alone PRC:
needs ids
' go (id) Year MakePRC Year
\ Note that Year is a creator-ID
\ registered with
\ http://palm.3com.com/devzone
\ Each released app must have a
\ unique, registered creator ID.

\ Add required resources:
2000 (id) MBAR copyrsrc \ menu
YearForm (id) tFRM copyrsrc
AboutBox (id) Talt copyrsrc
HelpString (id) tSTR copyrsrc

\ Copy icons:
1000 (id) tAIB copyrsrc \ normal
1001 (id) tAIB copyrsrc \ small

\ Delete unneeded forms:
MainFormID (id) tFRM delrsrc drop
TitledFormID (id) tFRM delrsrc drop

\ Done!
```

// PilRC resources for Year. 1998/11/11 Neal Bridges.

MENU ID 2000

BEGIN

PULLDOWN "Options"

BEGIN

MENUITEM "Help" ID 2002 "H"

MENUITEM "About..." ID 2001 "A"

END

END

FORM ID 1001 AT (0 0 160 160)

MENUID 2000

BEGIN

FIELD ID 1103 AT (0 80 160 11) NONUSABLE FONT 0 MAXCHARS 80

END

ALERT ID 3000

HELPID 3001

INFORMATION

BEGIN

TITLE "About Year"

MESSAGE "Version 1.0\n"

"© 1998 Neal Bridges\n\n"

"This is a Quartus Forth application.\n\n"

"Use PgUp/PgDown to change the year."

BUTTONS "ok"

END

STRING ID 3001 "Year was written and compiled on-board a Palm III "\

"using Quartus Forth. "\

"It's a freeware demonstration of Quartus' ability to produce "\

"tight, fast, stand-alone apps.\n\n"

"Check out both Quartus and the source to Year at:\n"

"http://www.interlog.com/~nbridges\n\n"

"Year is functionally similar to Dale Gass' YearCal. "\

"It's 5K in size (Dale's is 11K).\n\n"

"One enhancement: the 'week starts' setting in Preferences "\

"determines whether the calendar week begins on Sunday or Monday.\n\n"

"Use PgUp and PgDown to change the year.\n\n"

"Have fun!\n"

"- Neal Bridges\n"

"nbridges@interlog.com"

Quartus Forth Manual

[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

The Sources to "Duco"

© Neal Bridges, November 1998. (info@quartus.net)

If you're unfamiliar with Quartus Forth, please take a look at the Quartus home page:
<<http://www.quartus.net/products/forth/>>

About This Package:

This package contains the source to a polished Quartus Forth application, complete with form controls (buttons), a menu and an icon. By following the provided directions, you can use the registered version of Quartus (or the development beta, during the beta-test period) to compile the app on-board your Palm PDA, and HotSync the resulting .PRC file down to your PC or Mac.

Duco is a standard four-function calculator with one key difference -- it works in Roman numerals.

The sources to *Duco* are provided for two purposes: first, as a demonstration of how to write a complete Quartus Forth application, with integrated PalmOS resources; and second, to demonstrate the creation of stand-alone PRCs from Quartus apps.

Beyond that, *Duco* is an fun program in its own right, and may be of interest from a coding perspective to new Forth programmers.

Files in this package:

- `ReadMe.htm` ; this file.
- [Duco.txt](#) ; The Quartus Forth source to *Duco* (fits in one MemoPad memo).
- [MakeDuco.txt](#) ; A Quartus script to compile *Duco* into a stand-alone PRC.
- `duco.mpa` ; `Duco.txt` and `MakeDuco.txt` in MemoPad archive format, for use with the Palm Desktop software.
- [Duco.rcp](#) ; A PilRC script defining PalmOS resources used by *Duco*.
- [IconBig.bmp](#) ; a 1-bpp icon bitmap.
- `DucoRsrc.bat` ; A batch file to generate `DucoRsrc.prc` (requires PilRC and Pila)
- `DucoRsrc.prc` ; Generated from `DucoRsrc.rcp`. Provided as a convenience for those of you lacking PilRC and Pila.

- `Duco.prc` ; The generated stand-alone application, ready to be installed. This is the production release; it also has a small icon, and has no splash-screen.

Other Required Sources:

Here's a breakdown of what files need what other files, to help you ensure that you have all the supporting library sources installed as memos in your MemoPad: All the files listed here are available in the [Quartus File Area](#).

- *duco*:
 - needs *toolkit* \ for enum
 - needs *core-ext* \ for value, etc.
 - needs *roman* \ for Roman numeral manipulation
 - needs *bitmap* \ to create the memory-indicator bitmap
 - needs *textalign* \ for type.right
 - needs *graphics* \ for pixel cursor-positioning, line and frame
 - needs *condthens* \ for cond and then
 - needs *fonts* \ for named font types
 - needs *toolkit* \ for enum
 - needs *DataMgr* \ for opening the resource database
 - needs *ids* \ for 4-character resource types
 - needs *Events* \ for named event types
 - needs *simple-handler* \ for menu handling
- *make-duco*:
 - needs *ids* \ for the creator ID

About Duco and Make-Duco:

Duco will compile and run under the free version of Quartus. *Make-Duco* generates a stand-alone PRC of *Duco*, and contains commands that will only function properly under a registered copy of Quartus (or the development beta, during the beta test period).

If you just want to try *Duco*, skip these steps and simply use the Palm Desktop software to HotSync `Duco.PRC` into your Palm device. If you want to compile it yourself, however, read on.

To compile and run Duco from the sources provided:

(note: <**Return**> means a Graffiti Return -- a pen stroke in the Graffiti area from upper right to lower left)

1. Using the Palm Desktop software, add `Duco.txt` and all supporting files as memos in the MemoPad on your Palm device.

2. HotSync `DucoRsrc.prc` into your Palm device.
3. You'll also need Quartus itself HotSync'd in (either the free or the registered version, version 0.8.2 or later).
4. Start Quartus.
5. At the prompt, type **`include duco`** <Return>
6. To run *Duco*: when prompted "ok", type **`go`** <Return>

To create a stand-alone PRC from the sources provided:

1. Follow steps 1-3 above.
2. Using the Palm Desktop software, add `MakeDuco.txt` as a memo in the MemoPad.
3. Start the registered version of Quartus.
4. Type **`include make-duco`** <Return>
5. To run *Duco*: from the Palm Application Launcher (or equivalent), start the "Duco" icon.
6. HotSync the device again; `Duco.PRC` will be copied into the backup directory under your Palm Desktop user directory.

Notes:

Quartus compiles the sources to *Duco* into native-code in about 10 seconds.

Make-Duco extracts approximately 4K of active application code from codespace, and builds it into a new .PRC file along with several required PalmOS resources (a form, menu, about box, help text, and icon). The result: `Duco.PRC`, about 8K in size.

`Duco.PRC` is automatically flagged for backup, and will be copied into `C:\...\username\backup\Duco.PRC` at the next HotSync. `Duco.PRC` is a proper stand-alone application that requires no runtime support, and can be HotSync'd (or beamed) into any another Pilot, PalmPilot, or Palm III.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998 [Neal Bridges](#). All rights reserved.

\ duco 98.11.18 5:23 pm NAB

needs toolkit
needs core-ext
needs roman
needs textalign
needs bitmap
needs condthens
needs graphics
needs fonts
needs resources
needs Events
needs ids
needs simple-handler

5 7 2constant m-size
20 5 2constant m-position

m-size bitmap memory-indicator
2 base !
1110110000000000 ,
1001001000000000 ,
1001001000000000 ,
1001001000000000 ,
1001001000000000 ,
1001001000000000 ,
decimal

\ For interim values:
0 value a
0 value b

2variable display
\ High-cell of display indicates
\ range errors:
display constant out-of-range

0 value memory

: get-display (-- u)
 display 2@ d>s ;

: set-display (d. --) display 2! ;

: zero-display (--) 0. set-display ;

: error (--) -1. set-display ;

: add-symbol (char --)
 >r
 display 2@ drop (-- u)
 split-roman (-- nnn 1000's)
\ Add char to end of low Roman:
 swap >roman 2dup + r> swap c!
\ Convert back to integer:
 1+ roman> (-- 1000's newnnn)
\ Combine new nnn and 1000's:
 >r 1000 m* r> m+ set-display ;

\ Opcodes:
0 enum OpType
 OpType do-nothing
 OpType add
 OpType subtract

```

OpType multiply
OpType divide

do-nothing value nextoperator

false value pendingequals

: clear ( -- )
  zero-display
  false to pendingequals
  do-nothing to nextoperator
  0 to a 0 to b ;

: refresh ( -- )
  boldFont font drop
\ Clear Roman display:
  16 107 22 44 erase-rectangle
\ Clear Arabic display:
  14 32 24 6 erase-rectangle
\ Display Roman:
  24 45 at
  out-of-range @ if
    s" error" type zero-display 0
  else get-display dup romantype
  then
\ Display Arabic:
  0 <# #s #> 38 type.right
\ Memory indicator:
  memory if
    m-position memory-indicator
  else
    m-size m-position erase-rectangle
  then ;

: args ( -- b. a. )
  b 0 a 0 ;

: interim ( -- )
  a to b get-display to a ;

: equals ( -- )
  pendingequals if interim then
  false to pendingequals
  nextoperator cond
    dup add = if
      args d+ set-display
    else dup subtract = if
      args d- set-display
    else dup multiply = if
      args drop nip m* set-display
    else dup divide = if
      a if
        args drop
        1 swap m*/ set-display
      else error
    then
  then drop refresh
  get-display to b ;

: operator ( opcode -- )
  pendingequals if equals then
  to nextoperator
  true to pendingequals
  interim zero-display ;

```

```
\ Button ids run from 2000 to 2016.
2000 constant first-item-id
```

```
\ Map form ctlEvents to chars:
: item>button ( id -- char )
  first-item-id -
  s" MDCLXVI/*+ -=wraek" drop + c@
;
```

```
\ Actions for each button:
: do-button ( char -- )
  cond
    dup [char] k = if clear refresh
    else dup [char] e = if \ clear error
      zero-display refresh
    else dup [char] w = if \ mc
      0 to memory refresh
    else dup [char] a = if \ m+
      get-display memory + to memory
      refresh
    else dup [char] r = if \ mr
      memory 0 set-display refresh
    else dup [char] + = if
      add operator
    else dup [char] - = if
      subtract operator
    else dup [char] * = if
      multiply operator
    else dup [char] / = if
      divide operator
    else dup [char] = = if equals
    else dup add-symbol refresh
  then drop ;
```

```
(ID) p4ap (ID) Duco use-resources
```

```
1000 constant DucoForm
```

```
: show-panel ( -- )
\ The buttons:
  DucoForm ShowForm
  ['] simple-handler eventhandler !
\ The display area:
  20 154 19 3 dialogFrame frame
  39 40 19 40 line ;
```

```
: do-event ( ekey -- )
  ctlSelectEvent = if
    event >abs itemid
    item>button do-button
  then ;
```

```
\ Main entry point:
: go
  clear show-panel
  0 to memory refresh
  begin ekey do-event again ;
```

\ make-duco 98.11.20 NAB

needs duco

' go (ID) Duco makeprc Duco

DucoForm (ID) tFRM copyrsrc

2000 (ID) MBAR copyrsrc \ menu

AboutBox (ID) Talt copyrsrc

HelpString (ID) tSTR copyrsrc

1000 (ID) tAIB copyrsrc \ large icon

1001 (ID) tAIB copyrsrc \ small icon

BlankFormID (ID) tFRM delrsrc drop

TitledFormID (ID) tFRM delrsrc drop

FORM ID 1000 AT (0 0 160 160)

MENUID 2000

BEGIN

```
  BUTTON "I" ID 2006 AT (35 116 27 28) LEFTANCHOR BOLDFRAME FONT 1
  BUTTON "=" ID 2011 AT (68 116 60 28) LEFTANCHOR BOLDFRAME FONT 2
  BUTTON "D" ID 2001 AT (35 81 27 28) LEFTANCHOR BOLDFRAME FONT 1
  BUTTON "L" ID 2003 AT (68 81 27 28) LEFTANCHOR BOLDFRAME FONT 1
  BUTTON "V" ID 2005 AT (101 81 27 28) LEFTANCHOR BOLDFRAME FONT 1
  BUTTON "M" ID 2000 AT (35 46 27 28) LEFTANCHOR BOLDFRAME FONT 1
  BUTTON "C" ID 2002 AT (68 46 27 28) LEFTANCHOR BOLDFRAME FONT 1
  BUTTON "X" ID 2004 AT (101 46 27 28) LEFTANCHOR BOLDFRAME FONT 1
  BUTTON "mr" ID 2013 AT (3 69 26 18) LEFTANCHOR BOLDFRAME FONT 2
  BUTTON "m+" ID 2014 AT (3 92 26 18) LEFTANCHOR BOLDFRAME FONT 2
  BUTTON "ce" ID 2015 AT (3 115 26 18) LEFTANCHOR BOLDFRAME FONT 2
  BUTTON "c" ID 2016 AT (3 138 26 18) LEFTANCHOR BOLDFRAME FONT 2
  BUTTON "mc" ID 2012 AT (3 46 26 18) LEFTANCHOR BOLDFRAME FONT 2
  BUTTON "÷" ID 2007 AT (134 46 23 19) LEFTANCHOR BOLDFRAME FONT 2
  BUTTON "x" ID 2008 AT (134 70 23 18) LEFTANCHOR BOLDFRAME FONT 2
  BUTTON "-" ID 2010 AT (134 93 23 18) LEFTANCHOR BOLDFRAME FONT 2
  BUTTON "+" ID 2009 AT (134 116 23 40) LEFTANCHOR BOLDFRAME FONT 2
  TITLE "Duco"
  LABEL "© 1998 Neal Bridges" ID 2100 AT (40 147) FONT 0
```

END

MENU ID 2000

BEGIN

PULLDOWN "Options"

BEGIN

MENUIITEM "Help" ID 2002 "H"

MENUIITEM "About..." ID 2001 "A"

END

END

ALERT ID 3000

HELPID 3001

INFORMATION

BEGIN

TITLE "About Duco"

MESSAGE "Roman numeral calculator version 1.1\n"\

"© 1998 Neal Bridges\n"\

"This is a Quartus Forth application."

BUTTONS "ok"

END

STRING ID 3001 "'Duco' is Latin for 'to calculate'. Duco is a standard
4-function "\

"calculator with one key difference -- it works in Roman numerals.\n"\

"Duco was written and compiled on-board a Palm III using Quartus Forth. It's "\

"a freeware demonstration of Quartus' ability to produce tight, fast, "\

"stand-alone apps.\n"\

"Check out both Quartus and the source to Duco at: \n"\

"http://www.interlog.com/~nbridges\n"\

"Duco only operates on Roman numbers in the range {0..65535}. Any value "\

"outside of this range will result in an error.\n"\

"A line appearing over a Roman numeral multiplies its value by 1000.\n"\

"Duco is intelligent about Roman numbers, and won't allow an invalid number "\

"to be entered.\n"\

"Have fun!\n"\

"- Neal Bridges\n"\

"nbridges@interlog.com"

Documentation for Words Specific to Quartus Forth

Quartus provides a number of kernel words beyond those specified by the ISO/ANSI Forth Standard:

- [Interface Words](#)
- [System Words](#)
- [Memory Words](#)
- [Stand-alone PRC Generation Words \(registered version only\)](#)
- [Dictionary Words](#)
- [Floating-Point Words](#)
- [General-Purpose Words](#)
- [PalmOS System Calls \(systraps\)](#)

Interface Words:

- **(ekey)** (*ticks. -- eventType*)
Waits a specified number of ticks (1/100ths of a second) for an event; if none is received, returns *nilEvent*. Like **EKEY**, but with an adjustable timeout parameter (**EKEY** has a fixed timeout of 500 milliseconds).
- **currentx** (-- *addr*)
Variable containing the current X-coordinate for text output, in pixels.
- **currenty** (-- *addr*)
Variable containing the current Y-coordinate for text output, in pixels.
- **>byte** (*char -- char<<8*)
Converts a parameter for a systrap expecting a byte value on the stack. Equivalent to 8 LSHIFT.
- **BlankFormID** (-- *BlankFormID*)
Returns the ID of the blank form resource provided by Quartus. For use with **ShowForm**.
- **MainForm** (--)
Displays the default Quartus form, and establishes a menu event handler for it.
- **MainFormID** (-- *MainFormID*)
Returns the ID of the main form resource provided by Quartus. For use with **ShowForm**.
- **ShowForm** (*formID --*)

Displays the form specified by **formID**, and establishes a default handler for it. Sets **window-bounds** to the form's dimensions.

- **TitledFormID** (-- *TitledFormID*)
Returns the ID of the titled form resource provided by Quartus. For use with **ShowForm**.
- **a0** (-- *a0*)
Places the 32-bit contents of the 68000 CPU A0 register on the data stack. Primarily for use with **systrap**.
- **d0** (-- *d0*)
Places the 32-bit contents of the 68000 CPU D0 register on the data stack. Primarily for use with **systrap**.
- **d0!** (*x*. --)
Sets the 68000 CPU D0 register to the 32-bit value provided.
- **about** (--)
Displays the Quartus "About Box" alert.
- **beep** (--)
One beep, no waiting. Equivalent to `2 >byte SndPlaySystemSound`.
- **callback** (--)
Temporarily makes the return stack the data stack. For use in event handlers, etc. Used in combination with **end-callback**.
- **end-callback** (--)
Reverses the action of **callback**.
- **event** (-- *addr*)
Address of the event structure containing the last received PalmOS event.
- **eventhandler** (-- *addr*)
The address of a variable containing the *xt* of an event handler called during event handling for forms displayed with **ShowForm**. Provides a 'hook' directly into the Quartus event handler.
- **handled** (*flag* --)
Used inside callback event handlers to indicate that a given event has, or has not, been handled.
- **font** (*newFontID* -- *oldFontID*)
Sets the current font to *newfontID* and returns the previous font as *oldfontID*.
- **hash** (*c-addr u* -- *hashvalue*)
Generates a 16-bit hash value for the string provided. Used internally by the Forth dictionary to index Forth word names for quick lookup.
- **include** (<*filename*> --)
Searches the MemoPad for <*filename*> and processes the source found there. Files may be nested to any reasonable depth.
- **needs** (<*filename*> --)
Like **include**, but will not re-include a file that has already been included.
- **more** (*newsetting* -- *oldsetting*)
Sets the on-screen pagination flag, and returns the previous value. Used by **WORDS**.

- **systrip** (*i*j u -- k*j*)
Calls PalmOS systrip *u*, with parameters *i*j* (or none), returning *k*j* (or none).
- **window-bounds** (-- *addr*)
Returns the address of a four-cell data structure containing the dimensions of the text area of the current form (topX, topY, width, height). Can be adjusted within applications, affecting the screen area used for text-output, word-wrap, and scrolling by **CR**, **TYPE**, **EMIT**, and **PAGE**.
- **wrap** (*newsetting -- oldsetting*)
Sets the word-wrap flag (referenced by **TYPE** and **EMIT**), and returns the previous value.

System Words:

- **cold** (--)
Resets the system state. The dictionary is reset to containing only kernel words, the memo *startup-quartus* is re-read, etc. Equivalent to exiting and restarting Quartus.
- **(bye)** (--)
Called internally by Quartus when an appStopEvent is received. Terminates the current Quartus session.

Memory Words:

- **c@a** (*addr. -- char*)
Fetches *char* from a 32-bit absolute address.
- **c!a** (*char addr. --*)
Stores *char* at a 32-bit absolute address.
- **@a** (*addr. -- x*)
Fetches *x* from a 32-bit absolute address.
- **!a** (*x addr. --*)
Stores *x* at a 32-bit absolute address.
- **2@a** (*addr. -- x1 x2*)
Fetches the cell pair *x1 x2* from a 32-bit absolute address.
- **2!a** (*x1 x2 addr. --*)
Stores the cell pair *x1 x2* at a 32-bit absolute address.
- **>abs** (*addr -- addr.*)
Converts a 16-bit dataspace address to a 32-bit absolute address.
- **>rel** (*addr. -- addr*)
Converts a 32-bit absolute dataspace address to a relative 16-bit address.
- **xt>abs** (*xt -- xt.*)
Converts an *xt* to a 32-bit absolute address.
- **mem** (-- *largest-block. total-available.*)
Compacts unallocated space in the dynamic memory heap, and reports dynamic memory remaining.
- **csc@** (*code-addr -- char*)

Fetches *char* from a codespace address.

- **csc!** (*char code-addr --*)
Stores *char* at a codespace address.
- **cs@** (*code-addr -- x*)
Fetches *x* from a codespace address.
- **cs!** (*x code-addr --*)
Stores cell *x* at a codespace address.
- **cs,** (*x --*)
Stores cell *x* at the current codespace offset; increments the codespace pointer by one cell (2 bytes).
- **csc,** (*char --*)
Stores *char* at the current codespace offset; increments the codespace pointer by one character (1 byte).
- **rp@** (*-- rp-addr.*)
Puts the 32-bit contents of the return stack pointer on the data stack.
- **rp!** (*rp-addr. --*)
Sets the return stack pointer to the 32-bit address provided.
- **sp@** (*-- sp-addr.*)
Puts the 32-bit contents of the data stack pointer on the data stack.
- **sp!** (*sp-addr. --*)
Sets the data stack pointer to the 32-bit address provided.
- **cshere** (*-- cshere*)
Returns the codespace pointer.
- **csunused** (*-- csunused*)
Returns the number of unused bytes remaining in codespace.

Stand-alone PRC Generation Words ([registered version](#) only):

- **MakePRC** (*xt creator-id. <name> --*)
Generates a stand-alone executable (PRC) with the specified *creator-id* and name. Recursively extracts code from the main entry point specified by *xt*, and creates a relocatable executable image in the target PRC. Processes dataspace; creates a data image in the target PRC.
- **NewRsrc** (*resID resType. usize -- 0/err*)
Creates a new resource in the target, with the specified size. Must only be used *after MakePRC*.
- **CopyRsrc** (*resID resType. --*)
Searches open resource databases for the specified resource and copies it into the target PRC. Must only be used *after MakePRC*.
- **DelRsrc** (*resID resType. -- 0/err*)
Deletes the specified resource from the target PRC. Must only be used *after MakePRC*.
- **generate-symbols** (*flag --*)
true generate-symbols will cause **MakePRC** to generate symbolic debugging

information in the stand-alone PRC. These debugging symbols allow for Palm OS Emulator profiling and debugging; Forth words will be listed by name in the output. Must be used *before* **MakePRC**.

Dictionary Words:

- **allwords** (--)
Displays all words in all wordlists in the current search order.
- **(find)** (*c-addr u -- 0 / xt flag*)
Works just as does the Standard word **FIND**, but takes a standard string as a parameter, rather than a counted string.
- **(header)** (*c-addr u -- colon-sys*)
Creates a dictionary header with the name specified in the passed string. Called by **:**, **CREATE**, etc.
- **inline** (--)
Flags the most recently defined word as an 'inline' word.
- **lfa>xt** (*lfa -- xt*)
Given the address of a header in codespace, returns the execution token.
- **xt>name** (*xt -- 0 / cs-addr u -1*)
Returns the codespace address and length of the name of the routine specified by *xt* and a -1 to indicate success, or 0 if *xt* does not represent a valid named word. Note that the address returned is a *codespace* address; **cs@** must be used to read it.

Floating-Point Words:

- **fpdissect** (*F: r --*) (*-- sign exponent umantissa.*)
Splits a floating-point number into its components. *sign* is 1 for zero or positive floats, -1 for negative floats. *umantissa.* is the 32-bit representation of the normalized floating-point mantissa; *exponent* is the decimal value of the floating-point exponent.

General-Purpose Words:

- **0** (-- 0)
- **-1** (-- -1)
- **m/mod** (*num. denom -- quot. rem*)
Called by **#**.
- **for** (*+u --*)
Used to make simple, fast, loops, in conjunction with **next**. The loop index **I** counts downward within the loop from *u*-1 to 0.
- **next** (--)
See **for**.
- **noop** (--)
Does nothing.
- **(float)** (*<word> --*)

Handles the next word as a floating-point number.

- **(binary)** (*<word>* --)
Temporarily switches to base 2; parses and evaluates the next word in the input buffer.
- **(decimal)** (*<word>* --)
Temporarily switches to base 10; parses and evaluates the next word in the input buffer.
- **(hex)** (*<word>* --)
Temporarily switches to base 16; parses and evaluates the next word in the input buffer.
- **(octal)** (*<word>* --)
Temporarily switches to base 8; parses and evaluates the next word in the input buffer.
- **(radix)** (*base <word>* --)
Temporarily switches to the specified base; parses and evaluates the next word in the input buffer. An implementation factor of **(binary)**, **(decimal)**, **(hex)**, etc.
- **parse-word** (*<word>* -- *c-addr u*)
Returns the next space-delimited token from the input-buffer.
- **enough?** (*+n* --)
Checks that the stack contains at least *n* cells; if fewer than *n* cells are available, performs a `-4 THROW` (stack underflow).

Inline function tables: **select/xt/end-select**

- **select** (*+n* --)
Marks the beginning of a **select ... end-select** block, containing **xt** statements. Selects an **xt** from a function table. **select ... end-select** are used to create function lookup tables. **select** only works from within a definition.
- **xt** (*"name"* --)
Use only within a **select ... end-select** block. Compiles the **xt** of the name specified as part of a function lookup table.
- **end-select** (-- *xt*)
Marks the end of a **select** block. Returns the *xt* chosen from the table by the argument passed to **select**.

An example:

```
: zero ." Zero" ;  
: one ." One" ;  
: two ." Two" ;
```

```
: go  
select  
xt zero  
xt one  
xt two  
end-select execute ;
```

```
0 go Zero ok
```

1 go **One** ok
2 go **Two** ok

PalmOS System Calls (sysstraps):

Quartus Forth knows the names, parameters and return values of 828 PalmOS system calls.

From the PalmOS SDK documentation provided by Palm:

DmCloseDatabase

Purpose: Close a database.

Prototype: `Err DmCloseDatabase (DmOpenRef dbR)`

Parameters: `dbR` Database access pointer.

Result: Returns 0 if no error or `dmErrInvalidParam` if an error occurs.

The Quartus equivalent:

Purpose: Close a database.

Glossary entry: **DmCloseDatabase** (*dbP. -- Err*)

Parameters: *dbP.* Database access pointer.

Result: Returns 0 if no error or `dmErrInvalidParam` if an error occurs.

Here is a list of all PalmOS system calls, called *sysstraps*, transparently supported by Quartus Forth. Quartus Forth knows how many parameters each expects on the data stack, and what parameters each returns. Forth stack diagrams (--) are provided here for all sysstraps. The parameters are just as described in Palm's [PalmOS SDK documentation](#), but with the parameter order reversed right-to-left.

- **AbtShowAbout** (*creator. --*)
- **AlmAlarmCallback** (--)
- **AlmCancelAll** (--)
- **AlmDisplayAlarm** (*okToDisplay? --*)
- **AlmEnableNotification** (*enable? --*)
- **AlmGetAlarm** (*&refP. dbID. cardNo -- ud.*)
- **AlmInit** (-- *Err*)
- **AlmSetAlarm** (*quiet? alarmSeconds. ref. dbID. cardNo -- Err*)
- **CategoryCreateList** (*resizeList? editingStrID. numUneditableCategories[>byte] showUneditables? showAll? currentCategory &listP. db. --*)
- **CategoryCreateListV10** (*showAll? currentCategory &lst. db. --*)
- **CategoryEdit** (*numUneditableCategories[>byte] titleStrID. &category. db. -- Boolean*)
- **CategoryEditV10** (*&category. db. -- Boolean*)
- **CategoryEditV20** (*titleStrID. &category. db. -- Boolean*)
- **CategoryFind** (*&name. db. -- u*)

- **CategoryFreeList** (*editingStrID. showAll? &listP. db. --*)
- **CategoryFreeListV10** (*&lst. db. --*)
- **CategoryGetName** (*&name. index db. --*)
- **CategoryGetNext** (*index db. -- u*)
- **CategoryInitialize** (*localizedAppInfoStrID &appInfoP. --*)
- **CategorySelect** (*editingStrID. numUneditableCategories[>byte] &categoryName. &categoryP. title? lstID ctlID &frm. db. -- Boolean*)
- **CategorySelectV10** (*&categoryName. &categoryP. title? lstID ctlID &frm. db. -- Boolean*)
- **CategorySetName** (*&nameP. index db. --*)
- **CategorySetTriggerLabel** (*&name. &ctl. --*)
- **CategoryTruncateName** (*maxWidth &name. --*)
- **ClipboardAddItem** (*length &ptr. format[>byte] --*)
- **ClipboardGetItem** (*&length. format[>byte] -- VoidHand.*)
- **ConGetS** (*timeout. &message. -- Err*)
- **ConPutS** (*&message. -- Err*)
- **Crc16CalcBlock** (*crc count &bufP. -- u*)
- **CtlDrawControl** (*&pControl. --*)
- **CtlEnabled** (*&pControl. -- Boolean*)
- **CtlEraseControl** (*&pControl. --*)
- **CtlGetLabel** (*&pControl. -- &CharPtr.*)
- **CtlGetValue** (*&pControl. -- n*)
- **CtlHandleEvent** (*&pEvent. &pControl. -- Boolean*)
- **CtlHideControl** (*&pControl. --*)
- **CtlHitControl** (*&pControl. --*)
- **CtlNewControl** (*leftAnchor? group[>byte] font[>byte] height width y x &textP. style[>byte] iD &formPP. -- &ControlPtr.*)
- **CtlSetEnabled** (*usable? &pControl. --*)
- **CtlSetLabel** (*&newLabel. &pControl. --*)
- **CtlSetUsable** (*usable? &pControl. --*)
- **CtlSetValue** (*newValue &pControl. --*)
- **CtlShowControl** (*&pControl. --*)
- **CtlValidatePointer** (*&pControl. -- Boolean*)
- **DateAdjust** (*adjustment. &dateP. --*)
- **DateDaysToDate** (*&date. days. --*)
- **DateSecondsToDate** (*&date. seconds. --*)
- **DateToAscii** (*&pString. dateFormat[>byte] years days[>byte] months[>byte] --*)
- **DateToDOWDMFormat** (*&pString. dateFormat[>byte] years days[>byte]*)

months[>byte] --)

- **DateToDays** (*date[>byte] -- ud.)*
- **DayDrawDaySelector** (*&pSelector. --)*
- **DayDrawDays** (*&pSelector. --)*
- **DayHandleEvent** (*&pEvent. &pSelector. -- Boolean)*
- **DayOfMonth** (*year day month -- u)*
- **DayOfWeek** (*year day month -- u)*
- **DaysInMonth** (*year month -- u)*
- **DbgCommSettings** (*&flagsP. &baudP. -- Err)*
- **DbgControl** (*DbgControl -- DbgControlFuncType)*
- **DbgGetMessage** (*timeout. &bufferP. -- &CharPtr.)*
- **DbgSrcMessage** (*&debugStr. --)*
- **DlkControl** (*¶m2P. ¶m1P. op[>byte] -- Err)*
- **DlkDispatchRequest** (*&sessP. -- Err)*
- **DlkGetSyncInfo** (*&logLenP. &logBufP. &nameBufP. &syncStateP. &lastSyncDateP. &succSyncDateP. -- Err)*
- **DlkSetLogEntry** (*append? textLen &textP. --)*
- **DlkStartServer** (*¶mP. -- Err)*
- **DmArchiveRecord** (*index dbP. -- Err)*
- **DmAttachRecord** (*&oldHP. newH. &atP. dbP. -- Err)*
- **DmAttachResource** (*resID resType. newH. dbP. -- Err)*
- **DmCloseDatabase** (*dbP. -- Err)*
- **DmCreateDatabase** (*resDB? type. creator. &nameP. cardNo -- Err)*
- **DmCreateDatabaseFromImage** (*&bufferP. -- Err)*
- **DmDatabaseInfo** (*&creatorP. &typeP. &sortInfoIDP. &appInfoIDP. &modNumP. &bckUpDateP. &modDateP. &crDateP. &versionP. &attributesP. &nameP. dbID. cardNo -- Err)*
- **DmDatabaseProtect** (*protect? dbID. cardNo -- Err)*
- **DmDatabaseSize** (*&dataBytesP. &totalBytesP. &numRecordsP. dbID. cardNo -- Err)*
- **DmDeleteCategory** (*categoryNum dbR. -- Err)*
- **DmDeleteDatabase** (*dbID. cardNo -- Err)*
- **DmDeleteRecord** (*index dbP. -- Err)*
- **DmDetachRecord** (*&oldHP. index dbP. -- Err)*
- **DmDetachResource** (*&oldHP. index dbP. -- Err)*
- **DmFindDatabase** (*&nameP. cardNo -- LocalID.)*
- **DmFindRecordByID** (*&indexP. uniqueID. dbP. -- Err)*
- **DmFindResource** (*resH. resID resType. dbP. -- n)*
- **DmFindResourceType** (*typeIndex resType. dbP. -- n)*

- **DmFindSortPosition** (*other &compar. &newRecordInfo. &newRecord. dbP. -- u*)
- **DmFindSortPositionV10** (*other &compar. &newRecord. dbP. -- u*)
- **DmGet1Resource** (*id type. -- VoidHand.*)
- **DmGetAppInfoID** (*dbP. -- LocalID.*)
- **DmGetDatabase** (*index cardNo -- LocalID.*)
- **DmGetLastError** (*-- Err*)
- **DmGetNextDatabaseByTypeCreator** (*&dbIDP. &cardNoP. onlyLatestVers? creator. type. &stateInfoP. newSearch? -- Err*)
- **DmGetRecord** (*index dbP. -- VoidHand.*)
- **DmGetResource** (*id type. -- VoidHand.*)
- **DmGetResourceIndex** (*index dbP. -- VoidHand.*)
- **DmInit** (*-- Err*)
- **DmInsertionSort** (*other &compar. dbR. -- Err*)
- **DmMoveCategory** (*dirty? fromCategory toCategory dbP. -- Err*)
- **DmMoveOpenDBContext** (*&dbP. &listHeadP. -- Err*)
- **DmMoveRecord** (*to from dbP. -- Err*)
- **DmNewHandle** (*size. dbP. -- VoidHand.*)
- **DmNewRecord** (*size. &atP. dbP. -- VoidHand.*)
- **DmNewResource** (*size. resID resType. dbP. -- VoidHand.*)
- **DmNextOpenDatabase** (*currentP. -- DmOpenRef.*)
- **DmNextOpenResDatabase** (*dbP. -- DmOpenRef.*)
- **DmNumDatabases** (*cardNo -- u*)
- **DmNumRecords** (*dbP. -- u*)
- **DmNumRecordsInCategory** (*category dbP. -- u*)
- **DmNumResources** (*dbP. -- u*)
- **DmOpenDatabase** (*mode dbID. cardNo -- DmOpenRef.*)
- **DmOpenDatabaseByTypeCreator** (*mode creator. type. -- DmOpenRef.*)
- **DmOpenDatabaseInfo** (*&resDBP. &cardNoP. &modeP. &openCountP. &dbIDP. dbP. -- Err*)
- **DmPositionInCategory** (*category index dbP. -- u*)
- **DmQueryNextInCategory** (*category &indexP. dbP. -- VoidHand.*)
- **DmQueryRecord** (*index dbP. -- VoidHand.*)
- **DmQuickSort** (*other &compar. dbP. -- Err*)
- **DmRecordInfo** (*&chunkIDP. &uniqueIDP. &attrP. index dbP. -- Err*)
- **DmReleaseRecord** (*dirty? index dbP. -- Err*)
- **DmReleaseResource** (*resourceH. -- Err*)
- **DmRemoveRecord** (*index dbP. -- Err*)
- **DmRemoveResource** (*index dbP. -- Err*)

- **DmRemoveSecretRecords** (*dbP. -- Err*)
- **DmResetRecordStates** (*dbP. -- Err*)
- **DmResizeRecord** (*newSize. index dbP. -- VoidHand.*)
- **DmResizeResource** (*newSize. resourceH. -- VoidHand.*)
- **DmResourceInfo** (*&chunkLocalIDP. &resIDP. &resTypeP. index dbP. -- Err*)
- **DmSearchRecord** (*&dbPP. recH. -- n*)
- **DmSearchResource** (*&dbPP. resH. resID resType. -- n*)
- **DmSeekRecordInCategory** (*category direction offset &indexP. dbP. -- Err*)
- **DmSet** (*value[>byte] bytes. offset. &recordP. -- Err*)
- **DmSetDatabaseInfo** (*&creatorP. &typeP. &sortInfoIDP. &appInfoIDP. &modNumP. &bckUpDateP. &modDateP. &crDateP. &versionP. &attributesP. &nameP. dbID. cardNo -- Err*)
- **DmSetRecordInfo** (*&uniqueIDP. &attrP. index dbP. -- Err*)
- **DmSetResourceInfo** (*&resIDP. &resTypeP. index dbP. -- Err*)
- **DmStrCopy** (*&srcP. offset. &recordP. -- Err*)
- **DmWrite** (*bytes. &srcP. offset. &recordP. -- Err*)
- **DmWriteCheck** (*bytes. offset. &recordP. -- Err*)
- **EncDES** (*encrypt? &dstP. &keyP. &srcP. -- Err*)
- **EncDigestMD4** (*digestP[>byte] strLen &strP. -- Err*)
- **EncDigestMD5** (*digestP[>byte] strLen &strP. -- Err*)
- **ErrDisplayFileLineMsg** (*&msg. lineNo &filename. --*)
- **ErrExceptionList** (*-- &Ptr.*)
- **ErrThrow** (*err. --*)
- **EvtAddEventToQueue** (*&event. --*)
- **EvtAddUniqueEventToQueue** (*inPlace? id. &eventP. --*)
- **EvtCopyEvent** (*&dest. &source. --*)
- **EvtDequeueKeyEvent** (*peek &eventP. -- Err*)
- **EvtDequeuePenPoint** (*&retP. -- Err*)
- **EvtDequeuePenStrokeInfo** (*&endPtP. &startPtP. -- Err*)
- **EvtEnableGraffiti** (*enable? --*)
- **EvtEnqueueKey** (*modifiers keycode ascii -- Err*)
- **EvtEnqueuePenPoint** (*&ptP. -- Err*)
- **EvtEventAvail** (*-- Boolean*)
- **EvtFlushKeyQueue** (*-- Err*)
- **EvtFlushNextPenStroke** (*-- Err*)
- **EvtFlushPenQueue** (*-- Err*)
- **EvtGetEvent** (*timeout. &event. --*)
- **EvtGetPen** (*&pPenDown. &pScreenY. &pScreenX. --*)

- **EvtGetPenBtnList** (&numButtons. -- &PenBtnInfoPtr.)
- **EvtGetSysEvent** (timeout. &eventP. --)
- **EvtInitialize** (--)
- **EvtKeyQueueEmpty** (-- Boolean)
- **EvtKeyQueueSize** (-- ud.)
- **EvtPenQueueSize** (-- ud.)
- **EvtProcessSoftKeyStroke** (&endPtP. &startPtP. -- Err)
- **EvtResetAutoOffTimer** (-- Err)
- **EvtSetKeyQueuePtr** (size. &keyQueueP. -- Err)
- **EvtSetPenQueuePtr** (size. &penQueueP. -- Err)
- **EvtSysEventAvail** (ignorePenUps? -- Boolean)
- **EvtSysInit** (-- Err)
- **EvtWakeup** (-- Err)
- **ExgAccept** (&socketP. -- Err)
- **ExgConnect** (&socketP. -- Err)
- **ExgDBRead** (keepDates? &needResetP. cardNo &dbIDP. &userDataP. &deleteProcP. &readProcP. -- Err)
- **ExgDBWrite** (cardNo dbID. &nameP. &userDataP. &writeProcP. -- Err)
- **ExgDisconnect** (error &socketP. -- Err)
- **ExgGet** (&socketP. -- Err)
- **ExgInit** (-- Err)
- **ExgNotifyReceive** (&socketP. -- Err)
- **ExgPut** (&socketP. -- Err)
- **ExgReceive** (&err. bufLen. &bufP. &socketP. -- ud.)
- **ExgRegisterData** (&dataTypesP. id creatorID. -- Err)
- **ExgSend** (&err. bufLen. &bufP. &socketP. -- ud.)
- **FileClose** (stream. -- Err)
- **FileControl** (&valueLenP. &valueP. stream. op[>byte] -- Err)
- **FileDelete** (&nameP. cardNo -- Err)
- **FileOpen** (&errP. openMode. creator. type. &nameP. cardNo -- FileHand.)
- **FileReadLow** (&errP. numObj. objSize. dataStoreBased? offset. &baseP. stream. -- Long.)
- **FileSeek** (origin[>byte] offset. stream. -- Err)
- **FileTell** (&errP. &fileSizeP. stream. -- Long.)
- **FileTruncate** (newSize. stream. -- Err)
- **FileWrite** (&errP. numObj. objSize. &dataP. stream. -- Long.)
- **Find** (&goToP. --)
- **FindDrawHeader** (&title. &findParams. -- Boolean)

- **FindGetLineBounds** (*&r. &findParams. --*)
- **FindSaveMatch** (*dbID. cardNo appCustom. fieldNum pos recordNum &findParams. -- Boolean*)
- **FindStrInStr** (*&posP. &strToFind. &strToSearch. -- Boolean*)
- **FldCalcFieldHeight** (*maxWidth &chars. -- u*)
- **FldCompactText** (*&fld. --*)
- **FldCopy** (*&fld. --*)
- **FldCut** (*&fld. --*)
- **FldDelete** (*end start &fld. --*)
- **FldDirty** (*&fld. -- Boolean*)
- **FldDrawField** (*&fld. --*)
- **FldEraseField** (*&fld. --*)
- **FldFreeMemory** (*&fld. --*)
- **FldGetAttributes** (*&attrP. &fld. --*)
- **FldGetBounds** (*&rect. &fld. --*)
- **FldGetFont** (*&fld. -- FontID*)
- **FldGetInsPtPosition** (*&fld. -- u*)
- **FldGetMaxChars** (*&fld. -- u*)
- **FldGetNumberOfBlankLines** (*&fld. -- u*)
- **FldGetScrollPosition** (*&fld. -- u*)
- **FldGetScrollValues** (*&fieldHeightP. &textHeightP. &scrollPosP. &fld. --*)
- **FldGetSelection** (*&endPosition. &startPosition. &fld. --*)
- **FldGetTextAllocatedSize** (*&fld. -- u*)
- **FldGetTextHandle** (*&fld. -- Handle.*)
- **FldGetTextHeight** (*&fld. -- u*)
- **FldGetTextLength** (*&fld. -- u*)
- **FldGetTextPtr** (*&fld. -- &CharPtr.*)
- **FldGetVisibleLines** (*&fld. -- u*)
- **FldGrabFocus** (*&fld. --*)
- **FldHandleEvent** (*&pEvent. &fld. -- Boolean*)
- **FldInsert** (*insertLen &insertChars. &fld. -- Boolean*)
- **FldMakeFullyVisible** (*&fld. -- Boolean*)
- **FldNewField** (*numeric? hasScrollBar? autoShift? justification[>byte] dynamicSize? singleLine? underlined? editable? maxChars. font[>byte] height width y x id &formPP. -- &FieldPtr.*)
- **FldPaste** (*&fld. --*)
- **FldRecalculateField** (*redraw? &fld. --*)
- **FldReleaseFocus** (*&fld. --*)

- **FldScrollField** (*direction[>byte] linesToScroll &fld. --*)
- **FldScrollable** (*direction[>byte] &fld. -- Boolean*)
- **FldSendChangeNotification** (*&fld. --*)
- **FldSendHeightChangeNotification** (*numLines pos &fld. --*)
- **FldSetAttributes** (*&attrP. &fld. --*)
- **FldSetBounds** (*&rect. &fld. --*)
- **FldSetDirty** (*dirty? &fld. --*)
- **FldSetFont** (*fontID[>byte] &fld. --*)
- **FldSetInsPtPosition** (*pos &fld. --*)
- **FldSetInsertionPoint** (*pos &fld. --*)
- **FldSetMaxChars** (*maxChars &fld. --*)
- **FldSetScrollPosition** (*pos &fld. --*)
- **FldSetSelection** (*endPosition startPosition &fld. --*)
- **FldSetText** (*size offset textHandle. &fld. --*)
- **FldSetTextAllocatedSize** (*allocatedSize &fld. --*)
- **FldSetTextHandle** (*textHandle. &fld. --*)
- **FldSetTextPtr** (*&textP. &fld. --*)
- **FldSetUsable** (*usable? &fld. --*)
- **FldUndo** (*&fld. --*)
- **FldWordWrap** (*maxWidth &chars. -- u*)
- **FntAverageCharWidth** (*-- n*)
- **FntBaseLine** (*-- n*)
- **FntCharHeight** (*-- n*)
- **FntCharWidth** (*char[>byte] -- n*)
- **FntCharsInWidth** (*&fitWithinWidth. &stringLengthP. &stringWidthP. &string. --*)
- **FntCharsWidth** (*len &chars. -- n*)
- **FntDefineFont** (*&fontP. font[>byte] -- Err*)
- **FntDescenderHeight** (*-- n*)
- **FntGetFont** (*-- FontID*)
- **FntGetFontPtr** (*-- &FontPtr.*)
- **FntGetScrollValues** (*&topLine. &linesP. scrollPos width &chars. --*)
- **FntLineHeight** (*-- n*)
- **FntLineWidth** (*length &pChars. -- n*)
- **FntSetFont** (*font[>byte] -- FontID*)
- **FntWordWrap** (*maxWidth &chars. -- u*)
- **FntWordWrapReverseNLines** (*&scrollPosP. &linesToScrollP. maxWidth &chars. --*)
- **FontSelect** (*fontID[>byte] -- FontID*)
- **FplAToF** (*&s. &FloatType[8]. --*)

- **FplAdd** (*b[8] a[8] &FloatType[8]. --*)
- **FplBase10Info** (*&signP. &exponentP. &mantissaP. f[8] -- Err*)
- **FplDiv** (*divisor[8] dividend[8] &FloatType[8]. --*)
- **FplFToA** (*&s. f[8] -- Err*)
- **FplFloatToLong** (*f[8] -- Long.*)
- **FplFloatToULong** (*f[8] -- ud.*)
- **FplFree** (*--*)
- **FplInit** (*-- Err*)
- **FplLongToFloat** (*x. &FloatType[8]. --*)
- **FplMul** (*b[8] a[8] &FloatType[8]. --*)
- **FplSub** (*b[8] a[8] &FloatType[8]. --*)
- **FrmActiveState** (*save? &stateP. -- Err*)
- **FrmAddSpaceForObject** (*objectSize objectKind[>byte] &objectPP. &formPP. -- Err*)
- **FrmAlert** (*alertId -- u*)
- **FrmCloseAllForms** (*--*)
- **FrmCopyLabel** (*&newLable. lableID &frm. --*)
- **FrmCopyTitle** (*&newTitle. &frm. --*)
- **FrmCustomAlert** (*&s3. &s2. &s1. alertId -- u*)
- **FrmDeleteForm** (*&frm. --*)
- **FrmDispatchEvent** (*&eventP. -- Boolean*)
- **FrmDoDialog** (*&frm. -- u*)
- **FrmDrawForm** (*&frm. --*)
- **FrmEraseForm** (*&frm. --*)
- **FrmGetActiveForm** (*-- &FormPtr.*)
- **FrmGetActiveFormID** (*-- u*)
- **FrmGetControlGroupSelection** (*groupNum[>byte] &frm. -- Byte*)
- **FrmGetControlValue** (*controlID &frm. -- n*)
- **FrmGetFirstForm** (*-- &FormPtr.*)
- **FrmGetFocus** (*&frm. -- u*)
- **FrmGetFormBounds** (*&r. &frm. --*)
- **FrmGetFormId** (*&frm. -- u*)
- **FrmGetFormPtr** (*formId -- &FormPtr.*)
- **FrmGetGadgetData** (*objIndex &frm. -- &VoidPtr.*)
- **FrmGetLabel** (*lableID &frm. -- &CharPtr.*)
- **FrmGetNumberOfObjects** (*&frm. -- u*)
- **FrmGetObjectBounds** (*&r. pObjIndex &frm. --*)
- **FrmGetObjectId** (*objIndex &FormPtr. -- u*)
- **FrmGetObjectIndex** (*objID &frm. -- u*)

- **FrmGetObjectPosition** (&y. &x. *objIndex* &frm. --)
- **FrmGetObjectPtr** (*objIndex* &frm. -- &VoidPtr.)
- **FrmGetObjectType** (*objIndex* &frm. -- *FormObjectKind*)
- **FrmGetTitle** (&frm. -- &CharPtr.)
- **FrmGetUserModifiedState** (&frm. -- *Boolean*)
- **FrmGetWindowHandle** (&frm. -- *WinHandle.*)
- **FrmGotoForm** (*formId* --)
- **FrmHandleEvent** (&eventP. &frm. -- *Boolean*)
- **FrmHelp** (*helpMsgId* --)
- **FrmHideObject** (*objIndex* &frm. --)
- **FrmInitForm** (*rscID* -- &FormPtr.)
- **FrmNewBitmap** (*y x rscID iD* &formPP. -- &FormBitmapType.)
- **FrmNewForm** (*menuRscID helpRscID defaultButton modal? height width y x &titleStrP. formID* -- &FormPtr.)
- **FrmNewGadget** (*height width y x id* &formPP. -- &FormGadgetType.)
- **FrmNewLabel** (*font[>byte] y x &textP. iD* &formPP. -- &FormLabelType.)
- **FrmPointInTitle** (*y x* &frm. -- *Boolean*)
- **FrmPopupForm** (*formId* --)
- **FrmRemoveObject** (*objIndex* &formPP. -- *Err*)
- **FrmReturnToForm** (*formId* --)
- **FrmSaveAllForms** (--)
- **FrmSetActiveForm** (&frm. --)
- **FrmSetCategoryLabel** (&newLabel. *objIndex* &frm. --)
- **FrmSetControlGroupSelection** (*controlID groupNum[>byte]* &frm. --)
- **FrmSetControlValue** (*newValue controlID* &frm. --)
- **FrmSetEventHandler** (&FormEventHandlerPtr. &frm. --)
- **FrmSetFocus** (*fieldIndex* &frm. --)
- **FrmSetGadgetData** (&data. *objIndex* &frm. --)
- **FrmSetMenu** (*menuRscID* &frm. --)
- **FrmSetNotUserModified** (&frm. --)
- **FrmSetObjectBounds** (&bounds. *objIndex* &frm. --)
- **FrmSetObjectPosition** (*y x objIndex* &frm. --)
- **FrmSetTitle** (&newTitle. &frm. --)
- **FrmShowObject** (*objIndex* &frm. --)
- **FrmUpdateForm** (*updateCode formId* --)
- **FrmUpdateScrollers** (*scrollabledown? scrollableUp? downIndex upIndex* &frm. --)
- **FrmValidatePtr** (&frm. -- *Boolean*)
- **FrmVisible** (&frm. -- *Boolean*)

- **FtrGet** (*&valueP. featureNum creator. -- Err*)
- **FtrGetByIndex** (*&valueP. &numP. &creatorP. romTable? index -- Err*)
- **FtrInit** (*-- Err*)
- **FtrSet** (*newValue. featureNum creator. -- Err*)
- **FtrUnregister** (*featureNum creator. -- Err*)
- **GetCharAttr** (*-- &WordPtr.)*
- **GetCharCaselessValue** (*-- &BytePtr.)*
- **GetCharSortValue** (*-- &BytePtr.)*
- **GrfAddMacro** (*dataLen ¯oDataP. &nameP. -- Err*)
- **GrfAddPoint** (*&pt. -- Err*)
- **GrfCleanState** (*-- Err*)
- **GrfDeleteMacro** (*index -- Err*)
- **GrfFieldChange** (*&characterToDelete. resetState? -- Err*)
- **GrfFilterPoints** (*-- Err*)
- **GrfFindBranch** (*flags -- Err*)
- **GrfFlushPoints** (*-- Err*)
- **GrfFree** (*-- Err*)
- **GrfGetAndExpandMacro** (*&dataLenP. ¯oDataP. &nameP. -- Err*)
- **GrfGetGlyphMapping** (*&uncertainLenP. &dataLenP. &dataPtrP. &flagsP. glyphID -- Err*)
- **GrfGetMacro** (*&dataLenP. ¯oDataP. &nameP. -- Err*)
- **GrfGetMacroName** (*&nameP. index -- Err*)
- **GrfGetNumPoints** (*&numPtsP. -- Err*)
- **GrfGetPoint** (*&pointP. index -- Err*)
- **GrfGetState** (*&autoShiftedP. &tempShiftP. &numLockP. &capsLockP. -- Err*)
- **GrfInit** (*-- Err*)
- **GrfInitState** (*-- Err*)
- **GrfMatch** (*&matchInfoP. &uncertainLenP. &dataLenP. &dataPtrP. &flagsP. -- Err*)
- **GrfMatchGlyph** (*maxMatches maxUnCertainty &matchInfoP. -- Err*)
- **GrfProcessStroke** (*upShift? &endPtP. &startPtP. -- Err*)
- **GrfSetState** (*upperShift? numLock? capsLock? -- Err*)
- **GsiEnable** (*enableIt? --*)
- **GsiEnabled** (*-- Boolean*)
- **GsiInitialize** (*--*)
- **GsiSetLocation** (*y x --*)
- **GsiSetShiftState** (*tempShift lockFlags --*)
- **HwrBacklight** (*newState? set? -- Boolean*)
- **HwrBatteryLevel** (*-- u*)

- **HwrCursor** (*&blinkP. &sizeP. &whereP. &modeP. set? -- Err*)
- **HwrDelay** (*microseconds. --*)
- **HwrDisableDataWrites** (*--*)
- **HwrDockSignals** (*outputClr outputSet &inputsP. -- Err*)
- **HwrDoze** (*onlyNMI? --*)
- **HwrEnableDataWrites** (*-- Boolean*)
- **HwrGetRAMMapping** (*&logBlockInfoP. &physBlockInfoP. &numBlocksP. &cardInfoP. --*)
- **HwrGetROMToken** (*&sizeP. &dataP. token. cardNo -- Err*)
- **HwrIRQ1Handler** (*param. --*)
- **HwrIRQ2Handler** (*param. --*)
- **HwrIRQ3Handler** (*param. --*)
- **HwrIRQ4Handler** (*param. --*)
- **HwrIRQ5Handler** (*param. --*)
- **HwrIRQ6Handler** (*param. --*)
- **HwrLCDBaseAddr** (*&newAddrP. -- Err*)
- **HwrLCDDrawBitmap** (*clearFirst? &bitmapP. y x -- Err*)
- **HwrLCDGetDepth** (*request -- u*)
- **HwrLCDInit** (*depth height width &baseP. framePeriod -- Err*)
- **HwrLCDSleep** (*emergency? untilReset? -- Err*)
- **HwrLCDWake** (*-- Err*)
- **HwrMemReadable** (*&addr. -- DWord.*)
- **HwrMemWritable** (*&addr. -- DWord.*)
- **HwrPluggedIn** (*-- Boolean*)
- **HwrSetCPUDutyCycle** (*&dutyP. -- Err*)
- **HwrSetSystemClock** (*&freqP. -- Err*)
- **HwrSleep** (*emergency? untilReset? --*)
- **HwrTimerInit** (*timer -- Err*)
- **HwrTimerSleep** (*timer -- Err*)
- **HwrTimerWake** (*timer -- Err*)
- **HwrWake** (*-- Err*)
- **ImcReadFieldNoSemicolon** (*maxChars &c. inputFunc &inputStream. -- &CharPtr.*)
- **ImcReadFieldQuotablePrintable** (*maxChars quotedPrintable? stopAt &c. inputFunc &inputStream. -- &CharPtr.*)
- **ImcReadPropertyParameter** (*&valueP. &nameP. &cP. inputFunc &inputStream. --*)
- **ImcReadWhiteSpace** (*&c. &charAttrP. inputFunc &inputStream. --*)
- **ImcSkipAllPropertyParameters** (*"edPrintableP. &identifierP. &cP. inputFunc &inputStream. --*)

- **ImcStringIsAscii** (*&stringP. -- Boolean*)
- **ImcWriteNoSemicolon** (*&stringP. &outputFunc. &outputStream. --*)
- **ImcWriteQuotedPrintable** (*noSemicolons? &stringP. &outputFunc. &outputStream. --*)
- **InsPtCheckBlink** (*--*)
- **InsPtEnable** (*enableIt? --*)
- **InsPtEnabled** (*-- Boolean*)
- **InsPtGetHeight** (*-- n*)
- **InsPtGetLocation** (*&y. &x. --*)
- **InsPtInitialize** (*--*)
- **InsPtSetHeight** (*height --*)
- **InsPtSetLocation** (*y x --*)
- **KeyCurrentState** (*-- DWord.*)
- **KeyHandleInterrupt** (*status. periodic? -- ud.*)
- **KeyInit** (*-- Err*)
- **KeyRates** (*&queueAheadP. &doubleTapDelayP. &periodP. &initDelayP. set? -- Err*)
- **KeyResetDoubleTap** (*-- Err*)
- **KeySetMask** (*keyMask. -- DWord.*)
- **KeySleep** (*emergency? untilReset? -- Err*)
- **KeyWake** (*-- Err*)
- **LocGetNumberSeparators** (*&decimalSeparator. &thousandSeparator. numberFormat[>byte] --*)
- **LstDrawList** (*&list. --*)
- **LstEraseList** (*&list. --*)
- **LstGetNumberOfItems** (*&pList. -- u*)
- **LstGetSelection** (*&list. -- u*)
- **LstGetSelectionText** (*itemNum &list. -- &CharPtr.*)
- **LstGetVisibleItems** (*&pList. -- n*)
- **LstHandleEvent** (*&event. &list. -- Boolean*)
- **LstMakeItemVisible** (*itemNum &pList. --*)
- **LstNewList** (*triggerId visibleItems font[>byte] height width y x id &formPP. -- Err*)
- **LstPopupList** (*&list. -- n*)
- **LstScrollList** (*itemCount direction[>byte] &pList. -- Boolean*)
- **LstSetDrawFunction** (*&func. &pList. --*)
- **LstSetHeight** (*visibleItems &pList. --*)
- **LstSetListChoices** (*&numItems. &itemsText. &list. --*)
- **LstSetPosition** (*y x &pList. --*)
- **LstSetSelection** (*itemNum &list. --*)

- **LstSetTopItem** (*itemNum &pList. --*)
- **MdmDial** (*&phoneNumP. &userInitP. &okDialP. &modemP. -- Err*)
- **MdmHangUp** (*&modemP. -- Err*)
- **MemCardFormat** (*&ramStoreNameP. &manufNameP. &cardNameP. cardNo -- Err*)
- **MemCardInfo** (*&freeBytesP. &ramSizeP. &romSizeP. &crDateP. &versionP. &manufNameP. &cardNameP. cardNo -- Err*)
- **MemChunkFree** (*&chunkDataP. -- Err*)
- **MemChunkNew** (*attr size. heapID -- &VoidPtr.)*
- **MemCmp** (*numBytes. &s2. &s1. -- n*)
- **MemDebugMode** (*-- u*)
- **MemHandleCardNo** (*h. -- u*)
- **MemHandleDataStorage** (*h. -- Boolean*)
- **MemHandleFlags** (*h. -- u*)
- **MemHandleFree** (*h. -- Err*)
- **MemHandleHeapID** (*h. -- u*)
- **MemHandleLock** (*h. -- &VoidPtr.)*
- **MemHandleLockCount** (*h. -- u*)
- **MemHandleNew** (*size. -- VoidHand.)*
- **MemHandleOwner** (*h. -- u*)
- **MemHandleResetLock** (*h. -- Err*)
- **MemHandleResize** (*newSize. h. -- Err*)
- **MemHandleSetOwner** (*owner h. -- Err*)
- **MemHandleSize** (*h. -- ud.)*
- **MemHandleToLocalID** (*h. -- LocalID.)*
- **MemHandleUnlock** (*h. -- Err*)
- **MemHeapCheck** (*heapID -- Err*)
- **MemHeapCompact** (*heapID -- Err*)
- **MemHeapDynamic** (*heapID -- Boolean*)
- **MemHeapFlags** (*heapID -- u*)
- **MemHeapFreeByOwnerID** (*ownerID heapID -- Err*)
- **MemHeapFreeBytes** (*&maxP. &freeP. heapID -- Err*)
- **MemHeapID** (*heapIndex cardNo -- u*)
- **MemHeapInit** (*initContents? numHandles heapID -- Err*)
- **MemHeapPtr** (*heapID -- &VoidPtr.)*
- **MemHeapScramble** (*heapID -- Err*)
- **MemHeapSize** (*heapID -- ud.)*
- **MemInit** (*-- Err*)
- **MemInitHeapTable** (*cardNo -- Err*)

- **MemKernelInit** (*-- Err*)
- **MemLocalIDKind** (*local. -- LocalIDKind*)
- **MemLocalIDToGlobal** (*cardNo local. -- &VoidPtr.*)
- **MemLocalIDToLockedPtr** (*cardNo local. -- &VoidPtr.*)
- **MemLocalIDToPtr** (*cardNo local. -- &VoidPtr.*)
- **MemMove** (*numBytes. &sP. &dstP. -- Err*)
- **MemNVParams** (*¶msP. set? -- Err*)
- **MemNumCards** (*-- u*)
- **MemNumHeaps** (*cardNo -- u*)
- **MemNumRAMHeaps** (*cardNo -- u*)
- **MemPtrCardNo** (*&p. -- u*)
- **MemPtrDataStorage** (*&p. -- Boolean*)
- **MemPtrFlags** (*&p. -- u*)
- **MemPtrHeapID** (*&p. -- u*)
- **MemPtrNew** (*size. -- &VoidPtr.*)
- **MemPtrOwner** (*&p. -- u*)
- **MemPtrRecoverHandle** (*&p. -- VoidHand.*)
- **MemPtrResetLock** (*&p. -- Err*)
- **MemPtrResize** (*newSize. &p. -- Err*)
- **MemPtrSetOwner** (*owner &p. -- Err*)
- **MemPtrSize** (*&p. -- ud.*)
- **MemPtrToLocalID** (*&p. -- LocalID.*)
- **MemPtrUnlock** (*&p. -- Err*)
- **MemSemaphoreRelease** (*writeAccess? -- Err*)
- **MemSemaphoreReserve** (*writeAccess? -- Err*)
- **MemSet** (*value[>byte] numBytes. &dstP. -- Err*)
- **MemSetDebugMode** (*flags -- Err*)
- **MemStoreInfo** (*&databaseDirIDP. &initCodeOffset2P. &initCodeOffset1P. &heapListOffsetP. &bckUpDateP. &crDateP. &nameP. &flagsP. &versionP. storeNumber cardNo -- Err*)
- **MemStoreInit** (*numMstrPtrs &storeP. &blockInfoP. numBlocks cardNo firstHeapMstrPtrs firstHeapSize. -- Err*)
- **MemStoreSearch** (*&storePP. range. &startP. -- Err*)
- **MemStoreSetInfo** (*&databaseDirIDP. &initCodeOffset2P. &initCodeOffset1P. &heapListOffsetP. &bckUpDateP. &crDateP. &nameP. &flagsP. &versionP. storeNumber cardNo -- Err*)
- **MenuDispose** (*&pMenu. --*)
- **MenuDrawMenu** (*&pMenu. --*)
- **MenuEraseStatus** (*&pMenu. --*)

- **MenuGetActiveMenu** (-- *&MenuBarPtr.*)
- **MenuHandleEvent** (*&error. &event. &pMenu.* -- *Boolean*)
- **MenuInit** (*resourceId* -- *&MenuBarPtr.*)
- **MenuSetActiveMenu** (*&pMenu.* -- *&MenuBarPtr.*)
- **MenuSetActiveMenuRscID** (*resourceId* --)
- **PenCalibrate** (*&scrBotRightP. &scrTopLeftP. &digBotRightP. &digTopLeftP.* -- *Err*)
- **PenClose** (-- *Err*)
- **PenGetRawPen** (*&penP.* -- *Err*)
- **PenOpen** (-- *Err*)
- **PenRawToScreen** (*&penP.* -- *Err*)
- **PenResetCalibration** (-- *Err*)
- **PenScreenToRaw** (*&penP.* -- *Err*)
- **PenSleep** (-- *Err*)
- **PenWake** (-- *Err*)
- **PhoneNumberLookup** (*&fld.* --)
- **PrefGetAppPreferences** (*saved? &prefsSize. &prefs. id creator.* -- *n*)
- **PrefGetAppPreferencesV10** (*prefsSize &prefs. version type.* -- *Boolean*)
- **PrefGetPreference** (*choice[>byte]* -- *DWord.*)
- **PrefGetPreferences** (*&p.* --)
- **PrefOpenPreferenceDB** (*saved? -- DmOpenRef.*)
- **PrefOpenPreferenceDBV10** (-- *DmOpenRef.*)
- **PrefSetAppPreferences** (*saved? prefsSize &prefs. version id creator.* --)
- **PrefSetAppPreferencesV10** (*prefsSize &prefs. version creator.* --)
- **PrefSetPreference** (*value. choice[>byte]* --)
- **PrefSetPreferences** (*&p.* --)
- **PrgHandleEvent** (*&eventP. &prgGP.* -- *Boolean*)
- **PrgStartDialog** (*&textCallback. &title.* -- *&ProgressPtr.*)
- **PrgStopDialog** (*force? &prgP.* --)
- **PrgUpdateDialog** (*updateNow? &messageP. stage err &prgGP.* --)
- **PwdExists** (-- *Boolean*)
- **PwdRemove** (--)
- **PwdSet** (*&newPassword. &oldPassword.* --)
- **PwdVerify** (*&string.* -- *Boolean*)
- **RctCopyRectangle** (*&dstRect. &srcRect.* --)
- **RctGetIntersection** (*&r3. &r2. &r1.* --)
- **RctInsetRectangle** (*insetAmt &r.* --)
- **RctOffsetRectangle** (*deltaY deltaX &r.* --)
- **RctPtInRectangle** (*&r. y x* -- *Boolean*)

- **RctSetRectangle** (*height width top left &r. --*)
- **ResLoadForm** (*rscID -- &VoidPtr.*)
- **ResLoadMenu** (*rscID -- &VoidPtr.*)
- **SclDrawScrollBar** (*&bar. --*)
- **SclGetScrollBar** (*&pageSizeP. &maxP. &minP. &valueP. &bar. --*)
- **SclHandleEvent** (*&event. &bar. -- Boolean*)
- **SclSetScrollBar** (*pageSize max min value &bar. --*)
- **ScrCompressScanLine** (*firstLine? &dstParamP. width &prevLineP. &lineP. -- u*)
- **ScrCopyRectangle** (*lineCount bitCount toY toX fromY fromX &destWindow. &sourceWindow. -- Err*)
- **ScrDeCompressScanLine** (*width &dstP. &srcP. -- u*)
- **ScrDisplayMode** (*&enableColorP. &depthP. &heightP. &widthP. operation[>byte] -- Err*)
- **ScrDrawChars** (*&fontPtr. len &chars. clipRight clipBottom clipLeft clipTop yExtent xExtent yLoc xLoc &pWindow. --*)
- **ScrDrawNotify** (*updHeight updWidth updTop updLeft --*)
- **ScrInit** (*--*)
- **ScrLineRoutine** (*y2 x2 y1 x1 &pWindow. --*)
- **ScrRectangleRoutine** (*extentY extentX y x &pWindow. --*)
- **ScrScreenInfo** (*&pWindow. --*)
- **ScrSendUpdateArea** (*force? --*)
- **SelectDay** (*&title. &year. &day. &month. selectDayBy[>byte] -- Boolean*)
- **SelectDayV10** (*&title. &year. &day. &month. -- Boolean*)
- **SelectTime** (*startOfDay &title. untimed? &EndTimeP. &startTimeP. -- Boolean*)
- **SerReceiveISP** (*-- Boolean*)
- **SlkClose** (*-- Err*)
- **SlkCloseSocket** (*socket -- Err*)
- **SlkFlushSocket** (*timeout. socket -- Err*)
- **SlkOpen** (*-- Err*)
- **SlkOpenSocket** (*staticSocket? &socketP. libRefNum -- Err*)
- **SlkProcessRPC** (*&bodyP. &headerP. -- Err*)
- **SlkReceivePacket** (*timeout. bodySize &bodyP. &headerP. andOtherSockets? socket -- Err*)
- **SlkSendPacket** (*&writeList. &headerP. -- Err*)
- **SlkSetSocketListener** (*&socketP. socket -- Err*)
- **SlkSocketRefNum** (*&refNumP. socket -- Err*)
- **SlkSocketSetTimeout** (*timeout. socket -- Err*)
- **SlkSysPktDefaultResponse** (*&bodyP. &headerP. -- Err*)

- **SndCreateMidiList** (*&entHP. &wCountP. multipleDBs? creator. -- Boolean*)
- **SndDoCmd** (*noWait? &cmdP. &channelP. -- Err*)
- **SndGetDefaultVolume** (*&masterAmpP. &sysAmpP. &alarmAmpP. --*)
- **SndInit** (*-- Err*)
- **SndPlaySmf** (*bNoWait? &callbacksP. &chanRangeP. &selP. &smfP. cmd[>byte] &chanP. -- Err*)
- **SndPlaySystemSound** (*beepID[>byte] --*)
- **SndSetDefaultVolume** (*&defAmpP. &sysAmpP. &alarmAmpP. --*)
- **StrAToI** (*&str. -- Long.*)
- **StrCaselessCompare** (*&s2. &s1. -- n*)
- **StrCat** (*&src. &dst. -- &CharPtr.*)
- **StrChr** (*chr &str. -- &CharPtr.*)
- **StrCompare** (*&s2. &s1. -- n*)
- **StrCopy** (*&src. &dst. -- &CharPtr.*)
- **StrDelocalizeNumber** (*decimalSeparator thousandSeparator &s. -- &CharPtr.*)
- **StrIToA** (*i. &s. -- &CharPtr.*)
- **StrIToH** (*i. &s. -- &CharPtr.*)
- **StrLen** (*&src. -- u*)
- **StrLocalizeNumber** (*decimalSeparator thousandSeparator &s. -- &CharPtr.*)
- **StrNCaselessCompare** (*n. &s2. &s1. -- n*)
- **StrNCat** (*n &src. &dst. -- &CharPtr.*)
- **StrNCompare** (*n. &s2. &s1. -- n*)
- **StrNCopy** (*n &src. &dst. -- &CharPtr.*)
- **StrPrintf** (*... &formatStr. &s. -- n*)
- **StrStr** (*&token. &str. -- &CharPtr.*)
- **StrToLower** (*&src. &dst. -- &CharPtr.*)
- **StrVPrintf** (*&arg. &formatStr. &s. -- n*)
- **SysAppExit** (*&globalsP. &prevGlobalsP. &appInfoP. -- Err*)
- **SysAppLaunch** (*&resultP. &cmdPBP. cmd launchFlags dbID. cardNo -- Err*)
- **SysAppLauncherDialog** (*--*)
- **SysAppStartup** (*&globalsPtrP. &prevGlobalsP. &appInfoPP. -- Err*)
- **SysBatteryDialog** (*--*)
- **SysBatteryInfo** (*&percentP. &pluggedIn. &kindP. &maxTicksP. &criticalThresholdP. &warnThresholdP. set? -- u*)
- **SysBatteryInfoV20** (*&pluggedIn. &kindP. &maxTicksP. &criticalThresholdP. &warnThresholdP. set? -- u*)
- **SysBinarySearch** (*findFirst? &position. other. &searchData. &searchF. width numOfElements &baseP. -- Boolean*)

- **SysBroadcastActionCode** (*&cmdPBP. cmd -- Err*)
- **SysColdBoot** (*sysCardHeaderOffset. card1Size. &card1P. card0Size. &card0P. --*)
- **SysCopyStringResource** (*theID &string. --*)
- **SysCreateDataBaseList** (*lookupName? &dbIDs. &dbCount. creator. type. -- Boolean*)
- **SysCreatePanelList** (*&panelIDs. &panelCount. -- Boolean*)
- **SysCurAppDatabase** (*&dbIDP. &cardNoP. -- Err*)
- **SysCurAppInfoPV20** (*-- &SysAppInfoPtr.*)
- **SysDisableInts** (*-- u*)
- **SysDoze** (*onlyNMI? --*)
- **SysErrString** (*maxLen &strP. err -- &CharPtr.*)
- **SysEvGroupCreate** (*init. &tagP. &evIDP. -- Err*)
- **SysEvGroupRead** (*&valueP. evID. -- Err*)
- **SysEvGroupSignal** (*type. value. mask. evID. -- Err*)
- **SysEvGroupWait** (*timeout. matchType. value. mask. evID. -- Err*)
- **SysFatalAlert** (*&msg. -- u*)
- **SysFormPointerArrayToStrings** (*stringCount &c. -- VoidHand.*)
- **SysGetAppInfo** (*&actionCodeAppPP. &uiAppPP. -- &SysAppInfoPtr.*)
- **SysGetOSVersionString** (*-- &CharPtr.*)
- **SysGetStackInfo** (*&endPP. &startPP. -- Boolean*)
- **SysGetTrapAddress** (*trapNum -- &VoidPtr.*)
- **SysGraffitiReferenceDialog** (*referenceType[>byte] --*)
- **SysGremlins** (*¶ms. selector[>byte] -- DWord.*)
- **SysHandleEvent** (*&eventP. -- Boolean*)
- **SysInit** (*--*)
- **SysInsertionSort** (*other. &comparF. width numOfElements &baseP. --*)
- **SysKernelInfo** (*¶mP. -- Err*)
- **SysKeyboardDialog** (*kbd[>byte] --*)
- **SysKeyboardDialogV10** (*--*)
- **SysLaunchConsole** (*-- Err*)
- **SysLibFind** (*&refNumP. &nameP. -- Err*)
- **SysLibInstall** (*&refNumP. &libraryP. -- Err*)
- **SysLibLoad** (*&refNumP. libCreator. libType. -- Err*)
- **SysLibRemove** (*refNum -- Err*)
- **SysLibTblEntry** (*refNum -- &SysLibTblEntryPtr.*)
- **SysMailboxCreate** (*depth. &tagP. &mbIDP. -- Err*)
- **SysMailboxDelete** (*mbID. -- Err*)
- **SysMailboxFlush** (*mbID. -- Err*)
- **SysMailboxSend** (*wAck. &msgP. mbID. -- Err*)

- **SysMailboxWait** (*timeout. priority. &msgP. mbID. -- Err*)
- **SysNewOwnerID** (*-- u*)
- **SysQSort** (*other. &comparF. width numOfElements &baseP. --*)
- **SysRandom** (*newSeed. -- n*)
- **SysResSemaphoreCreate** (*&tagP. &smIDP. -- Err*)
- **SysResSemaphoreDelete** (*smID. -- Err*)
- **SysResSemaphoreRelease** (*smID. -- Err*)
- **SysResSemaphoreReserve** (*timeout. priority. smID. -- Err*)
- **SysReset** (*--*)
- **SysRestoreStatus** (*status --*)
- **SysSemaphoreCreate** (*initValue. &tagP. &smIDP. -- Err*)
- **SysSemaphoreDelete** (*smID. -- Err*)
- **SysSemaphoreSet** (*smID. -- Err*)
- **SysSemaphoreSignal** (*smID. -- Err*)
- **SysSemaphoreWait** (*timeout. priority. smID. -- Err*)
- **SysSetA5** (*newValue. -- DWord.*)
- **SysSetAutoOffTime** (*seconds -- u*)
- **SysSetPerformance** (*&cpuDutyP. &sysClockP. -- Err*)
- **SysSetTrapAddress** (*&procP. trapNum -- Err*)
- **SysSleep** (*emergency? untilReset? --*)
- **SysStringByIndex** (*maxLen &strP. index resID -- &CharPtr.*)
- **SysTaskCreate** (*tSlice. priority. attr. stackSize. &stackP. &codeP. &creator. &taskIDP. -- Err*)
- **SysTaskDelay** (*delay. -- Err*)
- **SysTaskDelete** (*priority. taskID. -- Err*)
- **SysTaskID** (*-- ud.*)
- **SysTaskResume** (*taskID. -- Err*)
- **SysTaskSetTermProc** (*&termProcP. taskID. -- Err*)
- **SysTaskSuspend** (*taskID. -- Err*)
- **SysTaskSwitching** (*enable? -- Err*)
- **SysTaskTrigger** (*taskID. -- Err*)
- **SysTaskUserInfoPtr** (*taskID. -- &SysTCBUserInfoPtr.*)
- **SysTaskWait** (*timeout. -- Err*)
- **SysTaskWaitClr** (*-- Err*)
- **SysTaskWake** (*taskID. -- Err*)
- **SysTicksPerSecond** (*-- u*)
- **SysTimerCreate** (*param. periodicDelay. &timerProc. &tagP. &timerIDP. -- Err*)
- **SysTimerDelete** (*timerID. -- Err*)

- **SysTimerRead** (*&valueP. timerID. -- Err*)
- **SysTimerWrite** (*value. timerID. -- Err*)
- **SysTranslateKernelErr** (*err -- Err*)
- **SysUIAppSwitch** (*&cmdPBP. cmd dbID. cardNo -- Err*)
- **SysUIBusy** (*value? set? -- u*)
- **SysUILaunch** (*--*)
- **SysUnimplemented** (*--*)
- **TblDrawTable** (*&table. --*)
- **TblEditing** (*&table. -- Boolean*)
- **TblEraseTable** (*&table. --*)
- **TblFindRowData** (*&rowP. data. &table. -- Boolean*)
- **TblFindRowID** (*&rowP. id &table. -- Boolean*)
- **TblGetBounds** (*&r. &table. --*)
- **TblGetColumnSpacing** (*column &table. -- u*)
- **TblGetColumnWidth** (*column &table. -- u*)
- **TblGetCurrentField** (*&table. -- &FieldPtr.*)
- **TblGetItemBounds** (*&r. column row &table. --*)
- **TblGetItemFont** (*column row &table. -- FontID*)
- **TblGetItemInt** (*column row &table. -- u*)
- **TblGetLastUsableRow** (*&table. -- u*)
- **TblGetNumberOfRows** (*&table. -- u*)
- **TblGetRowData** (*row &table. -- DWord.*)
- **TblGetRowHeight** (*row &table. -- u*)
- **TblGetRowID** (*row &table. -- u*)
- **TblGetSelection** (*&columnP. &rowP. &table. -- Boolean*)
- **TblGrabFocus** (*column row &table. --*)
- **TblHandleEvent** (*&event. &table. -- Boolean*)
- **TblHasScrollBar** (*hasScrollBar? &table. --*)
- **TblInsertRow** (*row &table. --*)
- **TblMarkRowInvalid** (*row &table. --*)
- **TblMarkTableInvalid** (*&table. --*)
- **TblRedrawTable** (*&table. --*)
- **TblReleaseFocus** (*&table. --*)
- **TblRemoveRow** (*row &table. --*)
- **TblRowInvalid** (*row &table. -- Boolean*)
- **TblRowSelectable** (*row &table. -- Boolean*)
- **TblRowUsable** (*row &table. -- Boolean*)
- **TblSelectItem** (*column row &table. --*)

- **TblSetBounds** (*&r. &table. --*)
- **TblSetColumnEditIndicator** (*editIndicator? column &table. --*)
- **TblSetColumnSpacing** (*spacing column &table. --*)
- **TblSetColumnUsable** (*usable? row &table. --*)
- **TblSetColumnWidth** (*width column &table. --*)
- **TblSetCustomDrawProcedure** (*&drawCallback. column &table. --*)
- **TblSetItemFont** (*fontID[>byte] column row &table. --*)
- **TblSetItemInt** (*value column row &table. --*)
- **TblSetItemPtr** (*&value. column row &table. --*)
- **TblSetItemStyle** (*type[>byte] column row &table. --*)
- **TblSetLoadDataProcedure** (*&loadDataCallback. column &table. --*)
- **TblSetRowData** (*data. row &table. --*)
- **TblSetRowHeight** (*height row &table. --*)
- **TblSetRowID** (*id row &table. --*)
- **TblSetRowSelectable** (*selectable? row &table. --*)
- **TblSetRowStaticHeight** (*staticHeight? row &table. --*)
- **TblSetRowUsable** (*usable? row &table. --*)
- **TblSetSaveDataProcedure** (*&saveDataCallback. column &table. --*)
- **TblUnhighlightSelection** (*&table. --*)
- **TimAdjust** (*adjustment. &dateTimeP. --*)
- **TimDateTimeToSeconds** (*&dateTimeP. -- ud.*)
- **TimGetAlarm** (*-- ud.*)
- **TimGetSeconds** (*-- ud.*)
- **TimGetTicks** (*-- ud.*)
- **TimHandleInterrupt** (*periodicUpdate? --*)
- **TimInit** (*-- Err*)
- **TimSecondsToDateTime** (*&dateTimeP. seconds. --*)
- **TimSetAlarm** (*seconds. -- ud.*)
- **TimSetSeconds** (*seconds. --*)
- **TimSleep** (*emergency? untilReset? -- Err*)
- **TimWake** (*-- Err*)
- **TimeToAscii** (*&pString. timeFormat[>byte] minutes[>byte] hours[>byte] --*)
- **UIInitialize** (*--*)
- **UIReset** (*--*)
- **WinAddWindow** (*winHandle. --*)
- **WinClipRectangle** (*&r. --*)
- **WinCopyRectangle** (*mode[>byte] destY destX &srcRect. dstWin. srcWin. --*)
- **WinCreateOffscreenWindow** (*&error. format[>byte] height width -- WinHandle.*)

- **WinCreateWindow** (*&error. focusable? modal? frameType &bounds. -- WinHandle.*)
- **WinDeleteWindow** (*eraseIt? winHandle.* --)
- **WinDisableWindow** (*winHandle.* --)
- **WinDisplayToWindowPt** (*&extentY. &extentX.* --)
- **WinDrawBitmap** (*y x &bitmapP.* --)
- **WinDrawChars** (*y x len &chars.* --)
- **WinDrawGrayLine** (*y2 x2 y1 x1* --)
- **WinDrawGrayRectangleFrame** (*&r. frameType* --)
- **WinDrawInvertedChars** (*y x len &chars.* --)
- **WinDrawLine** (*y2 x2 y1 x1* --)
- **WinDrawRectangle** (*cornerDiam &r.* --)
- **WinDrawRectangleFrame** (*&r. frameType* --)
- **WinDrawWindowFrame** (--)
- **WinEnableWindow** (*winHandle.* --)
- **WinEraseChars** (*y x len &chars.* --)
- **WinEraseLine** (*y2 x2 y1 x1* --)
- **WinEraseRectangle** (*cornerDiam &r.* --)
- **WinEraseRectangleFrame** (*&r. frameType* --)
- **WinEraseWindow** (--)
- **WinFillLine** (*y2 x2 y1 x1* --)
- **WinFillRectangle** (*cornerDiam &r.* --)
- **WinGetActiveWindow** (-- *WinHandle.*)
- **WinGetClip** (*&r.* --)
- **WinGetDisplayExtent** (*&extentY. &extentX.* --)
- **WinGetDisplayWindow** (-- *WinHandle.*)
- **WinGetDrawWindow** (-- *WinHandle.*)
- **WinGetFirstWindow** (-- *WinHandle.*)
- **WinGetFramesRectangle** (*&obscuredRect. &r. frameType* --)
- **WinGetPattern** (*&CustomPatternType.* --)
- **WinGetWindowBounds** (*&r.* --)
- **WinGetWindowExtent** (*&extentY. &extentX.* --)
- **WinGetWindowFrameRect** (*&r. winHandle.* --)
- **WinInitializeWindow** (*winHandle.* --)
- **WinInvertChars** (*y x len &chars.* --)
- **WinInvertLine** (*y2 x2 y1 x1* --)
- **WinInvertRectangle** (*cornerDiam &r.* --)
- **WinInvertRectangleFrame** (*&r. frameType* --)
- **WinModal** (*winHandle.* -- *Boolean*)

- **WinMoveWindowAddr** (*&newLocationP. &oldLocationP. --*)
- **WinRemoveWindow** (*winHandle. --*)
- **WinResetClip** (*--*)
- **WinRestoreBits** (*destY destX winHandle. --*)
- **WinSaveBits** (*&error. &source. -- WinHandle.*)
- **WinScrollRectangle** (*&vacated. distance direction[>byte] &r. --*)
- **WinSetActiveWindow** (*winHandle. --*)
- **WinSetClip** (*&r. --*)
- **WinSetColors** (*&oldBackColorP. &newBackColorP. &oldForeColorP. &newForeColorP. --*)
- **WinSetDrawWindow** (*winHandle. -- WinHandle.*)
- **WinSetPattern** (*&CustomPatternType. --*)
- **WinSetUnderlineMode** (*mode[>byte] -- UnderlineModeType*)
- **WinSetWindowBounds** (*&r. winHandle. --*)
- **WinValidateHandle** (*winHandle. -- Boolean*)
- **WinWindowToDisplayPt** (*&extentY. &extentX. --*)

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Online Resources

Forth Resources:

- [Quartus Forth Home Page](#)
 - Where it all happens!
- [Quartus Forth Discussion Forum](#)
 - Talk with a whole community of developers.
- [ISO/ANSI Forth Standard](#) in a variety of electronic formats
- [ISO/ANSI Forth Standard](#) in [Doc](#) format for reading on-board the PalmPilot
- [The Evolution of Forth](#) (overview and background)
- [Simple Forth](#) (an introduction by Leo Wong)
- [A Beginning Forth Tutorial](#)
- Article: [Lost at C? Forth May Be the Answer](#)
- [Forth Information on Taygeta](#) (compilers and literature)
- [FTP link to Forth Directories on Taygeta](#)
- [Forth Research Page](#)
- [Forth on the Web](#)
- [Journal of Forth Application and Research](#)
- [Forth Frequently Asked Questions](#)
- [Newsgroup comp.lang.forth](#) (via DejaNews)
- [Leo Brodie's Starting Forth examples, converted to ANS Forth by Benjamin Hoyt](#)

PalmPilot Programming Resources:

- [Adobe Acrobat Reader](#) (for viewing .PDF files)
- [Palm Computing® platform Development Zone](#)
 - [Palm OS Documentation](#)
 - [Palm OS Emulator](#)
 - [Creator ID Database](#)
- [Motorola](#) References
 - [Documentation and specifications for the Motorola](#)





- [MC68000 Documentation and Programmer's Reference Manual](#)
- [Newsgroup pilot.programmer](#) (via DejaNews)
- [RoadCoders HandHeld Software Development Resources](#)
- [RsrcEdit by Roger Lawrence](#) -- an on-board resource creation/editing tool for the PalmOS (versions 2 and later)
- [PilRC](#) (Pilot Resource Compiler)
- [Pila](#) (Pilot assembler)

Pilot Software Resources:

- [AportisDoc Reader](#) (for reading Doc-format documents on the PalmPilot)
- [PilotGear H.Q.](#) (Palm/Pilot software archive)
- [PilotZone](#) (TUCOWS repository of Palm software)
- [Quartus Home Page](#) for shareware HackMaster hacks that make editing source on the PalmPilot a more pleasant experience:
 - [LeftHack](#) for left-handed users -- moves most scrollbars over to the left edge of the Pilot screen.
 - [ShiftHack](#) disables Graffiti auto-shifting.
 - [ClearHack](#) removes the annoying dotted underlining from the MemoPad and other apps.
- Also of interest:
 - [CharHack](#) makes entering special symbols easy.
 - [PopUp Note](#) allows editing MemoPad notes while in Quartus Forth!
 - [Pilot-Xfer](#) is free software for Win32 and Linux that lets you take a full byte-for-byte backup of your Palm/Pilot.

[Table of Contents](#) - [Quartus Forth Home Page](#)

\ unique, registered creator ID.

When you next HotSync your PalmPilot, the generated PRC is automatically downloaded to your PC or Mac, and can be distributed and uploaded to any Palm device. It's completely stand-alone, and requires no run-time module.

Using External Resources

CopyRsrc (*resource# resourceID. --*)

DelRsrc (*resource# resourceID. -- success*)

You can create external PalmOS GUI resources using standard tools such as [PilRC](#), or the on-board resource compiler, [RsrcEdit](#). These resources can be used by your Quartus Forth applications, and compiled directly into stand-alone executables you create.

After **MakePRC**, you can use **CopyRsrc** and **DelRsrc** to copy resources into or delete resources from the generated target PRC file.

CopyRsrc will search for a specified resource by number and ID, and create a copy of that resource in the target PRC.

From the [sample application](#) *Year*:

```
3000 constant AboutBox
3001 constant HelpString
...
1001 constant YearForm
...
\ Add required resources:
2000 (ID) MBAR CopyRsrc \ menu
YearForm (ID) tFRM CopyRsrc
AboutBox (ID) Talt CopyRsrc
HelpString (ID) tSTR CopyRsrc

\ Copy icons:
1000 (ID) tAIB CopyRsrc \ normal
1001 (ID) tAIB CopyRsrc \ small
```

DelRsrc will delete a specified resource from the target PRC.

From [Year](#):

```
\ Delete unneeded forms:
MainFormID (ID) tFRM DelRsrc drop
TitledFormID (ID) tFRM DelRsrc drop
```

See the [Sample Applications](#) for more details and examples of the use of these words.

Caveats for Stand-Alone Applications

Bear in mind that generated stand-alone executables contain only the code required by your application. In a generated executable, there is no dictionary; dictionary headers on words are removed, and the interpreter is not present.

Also, take care not to rely on the value returned by ' (tick) in a stand-alone executable. An *xt* returned by ' during compilation will not match the *xt* of a word in a generated executable. Should you need to initialize a vector with an *xt*, use [] from within a word to do so; *xt*'s returned by [] are position-independent.

For example:

This technique would not produce working code in a stand-alone executable:

```
: myword ." Hello" cr ;
```

```
variable vector  
' myword vector !
```

```
: go vector @ execute ;
```

This is the correct method:

```
: myword ." Hello" cr ;
```

```
variable vector
```

```
: go  
[] myword vector !  
vector @ execute ;
```

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Glossary of Terms

PRC

The EXE of the Palm world. Executable programs for the Palm end in .PRC, and are installed using the Palm Desktop Software that ships with the Palm device.

ISO/ANSI Forth Standard

The [ISO/ANSI Forth Standard](#) was released in 1994. As of 1998, it is up for review.

From section 1.1, "Purpose":

"The purpose of this Standard is to promote the portability of Forth programs for use on a wide variety of computing systems, to facilitate the communication of programs, programming techniques, and ideas among Forth programmers, and to serve as a basis for the future evolution of the Forth language."

native code

Machine language; binary instructions that can be directly executed by the CPU (in the case of the Palm, the CPU is a [Motorola MC68328 DragonBall](#) processor, a member of the [Motorola 68000 family](#) of microprocessors).

source code

The human-readable text of a computer program.

object code

The machine-readable, native-code instruction sequences for a specific program.

interpreter

Reads commands input as text, and takes action depending upon the commands received.

pre-processor

A component of compilers for certain compiled languages (such as C). In those languages, as a first step before compilation, the source is passed through a pre-processor that performs certain text replacements, removes comments, and/or takes other actions based on specially-coded directives entered as part of the source.

compiler

Reads the human readable text of a program, written in programming language, and outputs object code.

optimization

Additional processing during the compilation process that ensures that generated object code is as small and fast as possible.

linker

Links object code and library code into a single, invokable form. Linking is one of the

final stages of producing an executable program.

debugger

A system component that provides tools for the developer to use in finding and removing bugs from a program.

assembler

Reads human-readable instructions that have a one-to-one correspondence to native-code machine instructions, and produces object code for a specific CPU.

word

In Forth, definitions are known as *words*.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

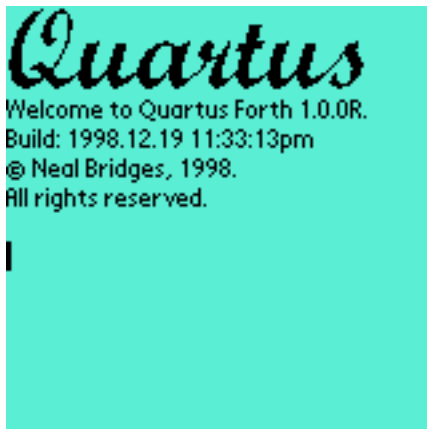
Quick Start

Installing Quartus Forth

To load the Quartus .PRC executable file into your PalmPilot/WorkPad, first unzip the distribution .zip file. Be sure and choose the 'make directories' option in your unzipping software. Use the *Install* option in the HotSync Desktop software that ships with the Palm device to transfer the Q4THxxxx.PRC file into the PDA, just as you would install any other Palm program.

Starting Quartus Forth

To start Quartus Forth, tap on its icon from the Application Launcher. After an initial "Quartus initializing workspace..." message, you will see a welcome screen similar to this:



The blinking cursor indicates that the system is waiting for your input.

Writing your First Quartus Forth Program

To test your new compiler, type this (where *<Return>* is a Graffiti return, a single pen stroke from upper-right to lower-left in the Graffiti area):

```
3 5 + . <Return>
```

The system will add 3 and 5, and display the result:

```
3 5 + . 8 ok
```

Now go to the MemoPad application, and create a new memo. It should look just as shown below. Note that for the memo to be recognized by Quartus Forth as a source file, the first two characters at the top of the memo must be backslash '\' and a single space, followed by the word

'hello'. Check your Palm documentation for how to use Graffiti to make any unfamiliar symbols.

Note also that the spaces between the various words are important. There's a space after the colon, before the period, after the first quote, and before the semi-colon.

Each line must end with a *<Return>*.

```
\ hello  
: go page ." Hello, World!" ;  
go
```

Compiling and Running your Program

Re-start Quartus Forth by tapping its icon from the Application Launcher. At the prompt, type the following:

```
include hello <Return>
```

Quartus Forth will find your file, load and compile it. The screen will clear, and you will see:

Hello, World! ok

Generating a Stand-alone Executable

If you have [registered Quartus Forth](#), you can create a stand-alone executable of your new program. For a new program that you want to distribute to others, you'll need to register a [creator ID](#) with Palm. Since this is just a simple demo, and not a program you'll be distributing, for the moment you can use the dummy creator ID (12345.) shown here.

Update your 'hello' memo so that it looks like this (note that new code is shown in *italics*). The character at the start of the MakePRC line is an apostrophe ('). The period at the end of 12345. is required.

```
\ hello  
: go page ." Hello, World!" begin key drop again ;  
' go 12345. MakePRC Hello
```

Now start Quartus Forth and include hello again, just as you did before. This time, you'll see

..... *ok*

A new program will now be in your Application Launcher, called **Hello**. This is a stand-alone executable, no run-time required. The next time you HotSync, Hello.PRC will be copied to your `c:\pilot\username\backup\` directory.

Tap on the 'Hello' icon and see what it does!

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Testing Procedures:

Quartus Forth undergoes rigorous release testing to ensure that new features integrate tightly with existing code. Both regression and unit tests are performed.

- Tested under the [PalmOS Emulator](#), against PalmOS version 1 (using Copilot), version 2, version 3 (both debug and real), and the EZ DragonBall (3.1, from the Palm IIIx and V) debug ROM images
- Regression and unit tests are performed on a PalmPilot Professional™ and a Palm III™ connected organizer

While Quartus Forth hasn't been formally submitted for [Palm Platinum certification](#), informally it meets all applicable requirements outlined in the testing guidelines. It successfully passes 1,000,000 Gremlin events in the PalmOS emulator with all debugging options switched on, and conforms with all described interface recommendations.

There is a single exception worthy of mention in section 7.2.4 of the testing guidelines: Quartus Forth, when re-started, doesn't restore the Forth system to its state when last exited, but rather starts up with a clean, kernel-only dictionary. This is a deliberate design decision; the Quartus Forth system state can be intricately involved with the PalmOS GUI state, and there is no reasonable way to preserve an unknown PalmOS GUI state between sessions.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Quartus Forth License Agreement and Limited Warranty

IMPORTANT: Read Before Using the Accompanying Software.

QUARTUS FORTH SOFTWARE DEVELOPMENT ENVIRONMENT SOFTWARE LICENSE AGREEMENT

CAREFULLY READ THE FOLLOWING CONDITIONS AND TERMS OF THIS AGREEMENT BEFORE USING THE ACCOMPANYING SOFTWARE, THE USE OF WHICH IS LICENSED FOR USE ONLY AS SET FORTH BELOW. IF YOU DO NOT AGREE TO THE CONDITIONS AND TERMS OF THIS AGREEMENT, DO NOT USE THE ACCOMPANYING SOFTWARE. USING ANY PART OF THE ACCOMPANYING SOFTWARE INDICATES THAT YOU ACCEPT THESE CONDITIONS AND TERMS.

Neal Bridges (the "Developer"), 65 Scadding Avenue #809, Toronto, Ontario, Canada, M5A 4L1 and the user of Quartus Forth (the "Licensee"), hereby agree as follows:

EVALUATION: The Developer grants to the Licensee a non-exclusive royalty-free license to use the "evaluation" version of the Quartus Forth software free of charge for an unlimited time for the purpose of evaluation for suitability to purchase a registered copy of the Quartus Forth software. All other terms and restrictions set forth in this license agreement are to remain in effect.

LICENSE: The Developer grants to the Licensee purchasing the [registered Quartus Forth software](#) (the "Software") a non-exclusive, royalty-free, nontransferable license to use the Software and its documentation in accordance with the terms and conditions of this License Agreement on any Palm Computing® platform product and/or any personal computer, provided that the Software is used only in connection with the development of products by the Licensee for use with the Palm Computing® platform products. This license provides for the use of the Software by a single individual only. Each individual who desires to use the Software must purchase a [registered copy](#).

Except as explicitly set forth below, (i) no license is granted to any rights to copyrights, patents, trademarks, trade secrets, or any other rights in respect to the Software; (ii) no license is granted to the human-readable code of the Software (source code).

The Developer retains title, all rights, interest, and ownership of, in and to the Software and accompanying documentation. The Software and its documentation are owned by the Developer, are protected by Canadian copyright laws and international treaty provisions, and may also be protected by other laws.

The Software contains certain sample source code in the form of library code, example applications and code fragments (both in the source code files and documentation provided hereunder), and may include tutorial applications (collectively, "Sample Source Code"). The Licensee may use the Sample Source Code internally to develop products for Palm Computing® platform products. The Licensee may distribute any such products built with the Sample Source Code, provided that the following copyright notice is included within the source code and in the location of any such product's copyright notice: "Portions copyright © 1998 Neal Bridges. All rights reserved."

The Developer grants the Licensee the right to develop and distribute products created using the facilities provided by the Software. Such products may be distributed on a royalty-free basis.

The Developer may furnish the Software to the Licensee electronically or on media in machine-readable object code form.

LICENSE RESTRICTIONS: Modification, decompilation, reverse engineering, reverse compiling, or disassembly of the Software is expressly prohibited. Any information obtained during such unlawful reverse engineering and/or decompilation activities, including but not limited to the logic, organization, algorithms and processes of the Software, shall be deemed confidential and proprietary information of the Developer.

Except as explicitly set forth in this License Agreement, the Licensee is not permitted to rent, lease, network, distribute, sublicense, loan, or create derivative works based upon the Software or its documentation in whole or in part, or to use the Software in a time-sharing arrangement or in any other unauthorized manner, unless with the written permission of the Developer.

The Licensee is not permitted to transfer or assign the Software or the rights under this license, without prior written permission of the Developer.

This License Agreement does not grant the Licensee any right to receive enhancements or updates to the Software or accompanying documentation. Updates and enhancements, if available, may be obtained by the Licensee at the Developer's then-current standard pricing, terms, and conditions. Title, ownership rights, and intellectual property rights to the content accessed through the Software are the property of the applicable content owner and may be protected by applicable copyright or other law. This License Agreement gives the Licensee no rights to such content.

TECHNICAL SUPPORT: For individual Licensees for a period of sixty (60) days from the date of purchase, ONLY LICENSED PURCHASERS OF THE FULL [REGISTERED QUARTUS FORTH SOFTWARE PRODUCT](#) will have access to technical support via electronic mail. Upgrades to the Software and its documentation, if any, are not included and may be sold separately.

LIMITED WARRANTY: If the Developer provides the Software on diskettes, he warrants the Software diskettes for a period of sixty (60) days from the date of original purchase from the Developer or an authorized retailer. To obtain warranty service, proof of date of purchase will be required. Any updates to the Software provided by the Developer (which may be provided at the Developer's sole discretion) shall be governed by the terms of the License Agreement. The Developer's entire liability and Licensee's exclusive remedy will be limited to the replacement of the defective diskettes upon return to the place of purchase within the warranty period. The Developer will not be responsible for replacement of any diskette damaged by accident, abuse, or misapplication.

The Developer makes no warranty or representation that the Software will meet the Licensee's requirements or that it will work in combination with any hardware or software products provided by third parties, or that the operation of the Software will be uninterrupted or error free, or that all defects in the Software will be corrected.

WARRANTIES EXCLUSIVE: TO THE MAXIMUM EXTENT PERMITTED BY LAW, THE FORGOING WARRANTIES AND REMEDIES ARE EXCLUSIVE AND ARE IN LIEU OF ALL OTHER WARRANTIES, TERMS, OR CONDITIONS, EXPRESS OR IMPLIED, EITHER IN FACT OR BY OPERATION OF LAW, STATUTORY OR OTHERWISE, INCLUDING WARRANTIES, TERMS, OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND SATISFACTORY QUALITY. THE DEVELOPER NEITHER ASSUMES NOR AUTHORIZES ANY OTHER PERSON TO ASSUME ANY OTHER LIABILITY IN CONNECTION WITH THE SALE, INSTALLATION, MAINTENANCE OR USE OF HIS PRODUCTS.

LIMITATION OF LIABILITY: TO THE MAXIMUM EXTENT PERMITTED BY LAW THE DEVELOPER ALSO EXCLUDES FOR HIMSELF AND HIS RESELLERS ANY LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL LOSSES OF ANY KIND OR FOR ANY REASON, OR FOR COMMERCIAL

LOSSES OF ANY KIND, OR OTHER FINANCIAL LOSS ARISING OUT OF OR IN CONNECTION WITH THE USE, SALE, MAINTENANCE, INSTALLATION, FAILURE, PERFORMANCE, OR INTERRUPTION OF THIS PRODUCT, EVEN IF THE DEVELOPER OR HIS AUTHORIZED RESELLER HAS BEEN ADVISED OF THE POSSIBILITY OR CERTAINTY OF SUCH DAMAGES, AND LIMITS HIS LIABILITY TO THE LICENSEE FOR ALL DAMAGES, LOSSES AND CAUSES OF ACTION (WHETHER BASED IN CONTRACT, TORT, INCLUDING NEGLIGENCE, OR OTHERWISE) TO REPLACEMENT OF THE SOFTWARE ON NEW MEDIA, OR REFUND OF THE PURCHASE PRICE PAID, AT THE SOLE OPTION OF THE DEVELOPER. THIS DISCLAIMER OF LIABILITY FOR DAMAGES WILL NOT BE AFFECTED IF ANY REMEDY PROVIDED HEREIN SHALL FAIL OF ITS ESSENTIAL PURPOSE.

SEVERABILITY: In the event any provision of this License Agreement is found to be unenforceable for any reason, the provision will be reformed only to the extent necessary to make it enforceable, and enforceability of the remaining provisions shall not in any way be affected or impaired.

TERM AND TERMINATION: This License Agreement is effective until terminated. The Licensee may terminate it any time by destroying the Software and documentation together with all copies. This License Agreement will also terminate immediately in the event of the Licensee's non-compliance with any condition or term of this License Agreement. Upon such termination, the Licensee must promptly destroy the Software and its documentation.

ENTIRE AGREEMENT: This License Agreement represents the entire understanding and agreement between the parties and supersedes all prior agreements or representations, whether written or oral, and may be amended only in a writing signed by both parties.

Neal Bridges
Toronto, CANADA
e-mail: info@quartus.net
<http://www.quartus.net/>

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Support Contacts

Please address your support questions to [Quartus Forth Support](#) (support@quartus.net).

Registered users, please provide your registration code and HotSync user name in your e-mail.

General questions and comments may be posted in the [Quartus Forth Discussion Forum](#); this is also where new releases and updates are announced.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Typographical Conventions

Through this documentation, typefaces are used in the following manner:

- This font is used for text, with *italic* used for stack comments, filenames, symbols, and the first appearance of new terms.
- Headings are shown in a sanserif **Bold** font.
- Forth keywords are shown in **bold** or `fixed-font`, depending on context.
- Code snippets and interactive text are shown in a fixed-font.
- <enclosed> indicates replaceable text in program source.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

About Forth

Forth is an incredibly interesting and powerful programming language. It's said that Forth gives a 10-times speed increase in development time over languages such as C.

For a thorough overview of the Forth programming language and its history and origins, I recommend reading [The Evolution of Forth](#).

Here are a few of Quartus Forth's advantages:

- Quartus is a full development environment: pre-processor, interpreter, optimizing compiler, debugger, and linker all in one [19K kernel plus associated resources](#)
- Compile Forth directly on-board the PalmPilot™ connected organizer; interactively test and debug your applications on the train, in meetings, anywhere!
- [Compliant with ISO/ANSI Standard Forth](#)
- Generates fast, compact, native-code applications
- Quick compilation times (approximately 1500 bytes of source/second)
- Elegant, expressive syntax
- Fully supports structured programming - Forth has no 'goto' instruction!
- Implicit parameter-passing via the Forth data stack
- Optimal approach to function calls and parameter passing encourages modularity and code re-use
- [Seamless integration with the underlying operating system](#)
- [Integrated inline 68000 assembler](#)
- Extracts application code to small-footprint stand-alone Palm-format (PRC) executables (no run-time required) ([Registered version only](#))

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Quartus Forth Internal Structure

Description	Resource	Approximate Size
Quartus Forth kernel and startup code	code 0000 code 0001	19K
Globals	data 0000	150 bytes
Icons	tAIB 1000 tAIB 1001	178 bytes
Preferences	pref 0001	10 bytes
Version	tver 0001	7 bytes
MakePRC scaffolding	imag 1000	1K
ISO/ANSI Standard Forth exception messages	thrw 0001	1.5K
PalmOS Systrap database	syst 0001	17K
Main form	tFRM 1000	124 bytes
Untitled form	tFRM 1001	114 bytes
Titled form	tFRM 1002	134 bytes
Quartus main menu	MBAR 1000	323 bytes
"Quartus" bitmap	Tbmp 1005	382 bytes
About alert	Talt 1000	108 bytes
Exception report alert	Talt 1005	50 bytes

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

System Requirements

When Quartus Forth is first run, it allocates a workspace of approximately 50K in size. This makes the total memory footprint of Quartus Forth approximately 90K. Ensure that you have at least 90K of storage available on your Palm/Pilot connected organizer before installation.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Setup

These files are provided in the distribution .ZIP file:

- *LICENSE.TXT* : The Quartus Forth end-user license and limited warranty.
- *INSTALL.TXT* : Quick installation instructions.
- *q4th####.prc* (where #### is the version number): The Quartus Forth Palm executable.
 - Install using the Palm Desktop software.
- *library.mpa* : Quartus Forth library modules, in MemoPad archive format.
 - Access this using the Palm Desktop software. Copy the library source to a category in your own Memopad, and then HotSync.
- *libtxt.zip* : All memo files from *library.mpa*, each as a .txt file; for import under Linux and on the Mac.
- *docs/* : Documents in Doc and text formats (ISO/ANSI Forth Standard, PalmOS stack diagrams)
 - View the Doc versions using a [PalmPilot Doc reader](#).
- *manual/* : Quartus Forth manual, in HTML format.
 - Contains the page you're currently viewing. View using your favourite web browser.
- *manual/sample/* : Sample source to production-quality Quartus Forth applications.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

The Quartus Forth Console

When you start Quartus Forth, you are presented with a welcome screen and a blinking cursor. This is where the Forth system accepts instructions. Each input line must be terminated by a Graffiti <Return>, which is a single stylus-stroke from upper-right to lower-left in the Graffiti area.

Main Menu Functions

If you press the Menu silkscreen button, you'll see the Quartus Forth main menu.

This menu presents four pulldowns: *Program*, *Edit*, *Go*, and *Options*:

Under *Program*:

- *Abort* sends a **-1 THROW** to the currently running program, clearing the stack and returning you to the input prompt.
- *Cold* restarts the Quartus Forth system, clearing the dictionary and returning it to its default state. The effect is the same as if you restarted Quartus Forth from the Application Launcher.

Edit provides standard Palm OS editing functions for the input line.

Go has one menu option: *Last Error*, which will take you to the Memo Pad app and highlight the exact place that the last error occurred while compiling your code.

Options has one menu option: *About*, which displays the Quartus Forth "About Box".

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Structure of a Quartus Forth Application

Forth applications are written bottom-up; no word may be referenced until it is defined. Definitions early in a source file are used to build the definitions that follow, leading up to the application's main entry point.

Here is the recommended high-level source layout of a Forth application (see the [sample application sources](#) for more examples):

```
\ <filename> <timestamp> <initials>
\ Comments about the program's purpose and design.

\ Required library code:
needs <libraryfile>
...

\ Opening of any external resource databases:
\ (via use-resources from the library file resources):
(ID) p4ap (ID) <creatorID> use-resources

\ Constant definitions:
<value> constant <constantname>
...

\ Variable declarations (globals):
variable <variablename>
...

\ Variable default initializations:
<value> <variablename> !
...

\ System-dependent words (preferences, etc.):
: <name> ... ;
...

\ Application-specific words:
: <name> ... ;
...

\ Program main entry point:
: go ( -- ) ... ;

\ Stand-alone executable generation (registered version only)
```

\ (Often this is placed in a separate file for convenience):

'[go <creatorID> **MakePRC** <ProgramName>](#)

\ Deletion of unneeded forms from the target PRC:

<id> <resourceType> **delsrc drop**

...

\ Addition of needed forms from an open resource database:

<id> <resourceType> **copysrc**

...

\ Done!

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Common Programming Errors

The following programming errors are commonly encountered when developing a Forth application. Many of them are caught by the *safe.txt* module in the [library](#)

- Division by zero
- Stack overflow -- leaving unconsumed cells on the data stack
- Stack underflow -- insufficient arguments on the data stack
- Unmatched IF ELSE THEN
- Unmatched BEGIN WHILE REPEAT (missing WHILE, or using UNTIL or AGAIN instead of REPEAT)
- Unmatched DO LOOP
- Missing or superfluous ; (semi-colon)
- Return stack imbalances -- unbalanced >R R>
- Cell-sized reads/writes to unaligned (odd) addresses
- Attempts to write to storage memory without using [DmWrite](#)
- LEAVE used outside of a DO LOOP
- Input-buffer overrun (source line longer than 80 characters)



Frequently Asked Questions

1. **Q:** When I exit Quartus and restart it, why is the Forth codespace/dataspace reset to its initial state?

A: This is a design decision. The state of Quartus Forth codespace/dataspace can be intricately involved with the PalmOS GUI state; since there's no reasonable way to preserve an unknown GUI state between sessions, the codespace/dataspace is not preserved either. The feedback I've received from developers indicates that most are pleased with this arrangement; the time lost in re-loading source during incremental development is more than made up for by the time saved in not trying to track down a problem arising from the previous system state.

2. **Q:** How is memory managed within the Palm OS?

A: Memory in the Palm device is divided into two types: *dynamic*, and *storage*.

Dynamic memory can be freely read and written to, and is limited to an amount between 12K and 64K, depending on the version of the Palm ROM. Dynamic memory is used as working memory for the currently-running application. The Palm OS also uses dynamic memory for networking, and house-keeping.

PalmOS databases (PRC and PDB files) are kept in storage memory. Storage memory can be freely read, but cannot be written to without the use of special PalmOS system calls (e.g. [DmWrite](#)). This affords protection to both applications and application data.

3. **Q:** Any chance of blowing away other applications or databases if I mess up in Forth?

A: No chance. Because of the way the Palm memory model is designed, other applications and data are protected from being accidentally corrupted or overwritten.

Barring the explicit use of [DmWrite](#), you cannot accidentally modify applications or data kept in storage memory.

4. **Q:** While developing Quartus applications, I find myself going back and forth from Quartus to the MemoPad. Is there anything that makes this simpler?

A: I recommend [SwitchHack](#), a HackMaster-compatible extension. Among other things, it allows you to switch back and forth between two applications with one pen-stroke.

I also recommend creating a Graffiti shortcut named 'i' that types 'include '. This speeds

up loading source during testing.

5. **Q:** What's the word on case-sensitivity?

A: Quartus is not case-sensitive as regards definition names and filenames. Accented characters are recognized as their equivalent letters without accents (a matches A matches ä). Between words in Forth programs, spaces and tabs are interchangeable.

6. **Q:** Quartus is a 16-bit Standard Forth. How do I perform 32-bit memory accesses?

A: The words @a !a 2@a 2!a c@a c!a are provided for 32-bit (absolute) memory accesses. See the [Words Specific to Quartus Forth](#) for more details.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

C Typedefs

In Palm's documentation, arguments for system functions are described using C typedefs. Many of these typedefs refer to exactly the same underlying data type. The table below lists all of the typedefs Palm defines, with their representation and examples of those values in Quartus Forth.

C typedef	Representing	Example values in Quartus Forth	Quartus Forth Stack Diagram Notation (<i>argument -- result</i>)
Fixed Size Data Types			
typedef char SByte typedef char Char	8 bits {-128..127}	-128 char A	(<i>name[>byte] -- name</i>)
typedef unsigned char Byte typedef unsigned char UChar	8 bits {0..255}	64	(<i>name[>byte] -- name</i>)
typedef unsigned char Boolean	8 bits {0=false, non-zero=true}	TRUE	(<i>name?</i>)
typedef unsigned short UInt typedef unsigned short UInt16 typedef unsigned short Word typedef unsigned short UShort	16 bits {0..65535}	41288	(<i>u</i>)
typedef short SWord typedef short Short typedef short Int16 typedef short Int typedef short Err	16 bits {-32768..32767}	-15000	(<i>n</i>)

typedef unsigned long UInt32 typedef unsigned long DWord typedef unsigned long ULong	32 bits {0..4294967295}	8675309.	(<i>u.</i>)
typedef DWord LocalID	32 bits representing a database ID	returned by DmFindDatabase, DmGetDatabase, etc.	(<i>localID.</i>)
typedef long Long typedef long SDWord typedef long Int32	32 bits {-2147483648..2147483647}	-12345678.	(<i>d.</i>)
Pointer Types			
typedef void* VoidPtr typedef VoidPtr* VoidHand typedef SByte* SBytePtr typedef Byte* BytePtr typedef SWord* SWordPtr typedef Word* WordPtr typedef unsigned short* UInt16Ptr typedef SDWord* SDWordPtr typedef DWord* DWordPtr typedef Boolean* BooleanPtr typedef Char* CharPtr typedef UChar* UCharPtr typedef Short* ShortPtr typedef UShort* UShortPtr typedef Int* IntPtr typedef UInt* UIntPtr	32 bits representing an address in memory	here >abs 32. MemHandleNew	(<i>&name.</i>)

typedef Long* LongPtr typedef ULong* ULongPtr typedef char* Ptr typedef Ptr* Handle			
typedef Long (* ProcPtr)()	32 bits representing the address of a function	' type xt>abs	(<i>&funcname.</i>)

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Pronunciation of Forth Words

A number of [Standard Forth](#) words have common pronunciations; the following table lists them. [Specific Quartus Forth words](#) follow the same general pattern as regards pronunciation.

!	<i>store</i>
#	<i>number-sign</i>
#>	<i>number-sign-greater</i>
#S	<i>number-sign-s</i>
#TIB	<i>number-t-i-b</i>
'	<i>tick</i>
(<i>paren</i>
(LOCAL)	<i>paren-local-paren</i>
*	<i>star</i>
*/	<i>star-slash</i>
*/MOD	<i>star-slash-mod</i>
+	<i>plus</i>
+!	<i>plus-store</i>
+LOOP	<i>plus-loop</i>
,	<i>comma</i>
-	<i>minus</i>
-TRAILING	<i>dash-trailing</i>
.	<i>dot</i>
."	<i>dot-quote</i>
.(<i>dot-paren</i>
.R	<i>dot-r</i>
.S	<i>dot-s</i>
/	<i>slash</i>
/MOD	<i>slash-mod</i>
/STRING	<i>slash-string</i>
0<	<i>zero-less</i>
0<>	<i>zero-not-equals</i>

0=	<i>zero-equals</i>
0>	<i>zero-greater</i>
1+	<i>one-plus</i>
1-	<i>one-minus</i>
2!	<i>two-store</i>
2*	<i>two-star</i>
2/	<i>two-slash</i>
2>R	<i>two-to-r</i>
2@	<i>two-fetch</i>
2CONSTANT	<i>two-constant</i>
2DROP	<i>two-drop</i>
2DUP	<i>two-dupe</i>
2LITERAL	<i>two-literal</i>
2OVER	<i>two-over</i>
2R>	<i>two-r-from</i>
2R@	<i>two-r-fetch</i>
2ROT	<i>two-rote</i>
2SWAP	<i>two-swap</i>
2VARIABLE	<i>two-variable</i>
:	<i>colon</i>
:NONAME	<i>colon-no-name</i>
;	<i>semicolon</i>
;CODE	<i>semicolon-code</i>
<	<i>less-than</i>
<#	<i>less-number-sign</i>
<>	<i>not-equals</i>
=	<i>equals</i>
>	<i>greater-than</i>
>BODY	<i>to-body</i>
>FLOAT	<i>to-float</i>
>IN	<i>to-in</i>
>NUMBER	<i>to-number</i>
>R	<i>to-r</i>
?	<i>question</i>
?DO	<i>question-do</i>
?DUP	<i>question-dupe</i>

@	<i>fetch</i>
ABORT"	<i>abort-quote</i>
ABS	<i>abs</i>
AT-XY	<i>at-x-y</i>
BL	<i>b-l</i>
BLK	<i>b-l-k</i>
C!	<i>c-store</i>
C"	<i>c-quote</i>
C,	<i>c-comma</i>
C@	<i>c-fetch</i>
CELL+	<i>cell-plus</i>
CHAR	<i>char</i>
CHAR+	<i>char-plus</i>
CHARS	<i>chars</i>
CMOVE	<i>c-move</i>
CMOVE>	<i>c-move-up</i>
COMPILE,	<i>compile-comma</i>
CR	<i>c-r</i>
CS-PICK	<i>c-s-pick</i>
CS-ROLL	<i>c-s-roll</i>
D+	<i>d-plus</i>
D-	<i>d-minus</i>
D.	<i>d-dot</i>
D.R	<i>d-dot-r</i>
D0<	<i>d-zero-less</i>
D0=	<i>d-zero-equals</i>
D2*	<i>d-two-star</i>
D2/	<i>d-two-slash</i>
D<	<i>d-less-than</i>
D=	<i>d-equals</i>
D>F	<i>d-to-f</i>
D>S	<i>d-to-s</i>
DABS	<i>d-abs</i>
DF!	<i>d-f-store</i>
DF@	<i>d-f-fetch</i>
DFALIGN	<i>d-f-align</i>

DFALIGNED	<i>d-f-aligned</i>
DFLOAT+	<i>d-float-plus</i>
DFLOATS	<i>d-floats</i>
DMAX	<i>d-max</i>
DMIN	<i>d-min</i>
DNEGATE	<i>d-negate</i>
DOES>	<i>does</i>
DU<	<i>d-u-less</i>
DUP	<i>dupe</i>
EKEY	<i>e-key</i>
EKEY>CHAR	<i>e-key-to-char</i>
EKEY?	<i>e-key-question</i>
EMIT?	<i>emit-question</i>
ENDCASE	<i>end-case</i>
ENDOF	<i>end-of</i>
ENVIRONMENT?	<i>environment-query</i>
F!	<i>f-store</i>
F*	<i>f-star</i>
F**	<i>f-star-star</i>
F+	<i>f-plus</i>
F-	<i>f-minus</i>
F.	<i>f-dot</i>
F/	<i>f-slash</i>
F0<	<i>f-zero-less-than</i>
F0=	<i>f-zero-equals</i>
F<	<i>f-less-than</i>
F>D	<i>f-to-d</i>
F@	<i>f-fetch</i>
FABS	<i>f-abs</i>
FACOS	<i>f-a-cos</i>
FACOSH	<i>f-a-cosh</i>
FALIGN	<i>f-align</i>
FALIGNED	<i>f-aligned</i>
FALOG	<i>f-a-log</i>
FASIN	<i>f-a-sine</i>
FASINH	<i>f-a-cinch</i>

FATAN	<i>f-a-tan</i>
FATAN2	<i>f-a-tan-two</i>
FATANH	<i>f-a-tan-h</i>
FCONSTANT	<i>f-constant</i>
FCOS	<i>f-cos</i>
FCOSH	<i>f-cosh</i>
FDEPTH	<i>f-depth</i>
FDROP	<i>f-drop</i>
FDUP	<i>f-dupe</i>
FE.	<i>f-e-dot</i>
FEXP	<i>f-e-x-p</i>
FEXPM1	<i>f-e-x-p-m-one</i>
FLITERAL	<i>f-literal</i>
FLN	<i>f-l-n</i>
FLNP1	<i>f-l-n-p-one</i>
FLOAT+	<i>float-plus</i>
FLOG	<i>f-log</i>
FM/MOD	<i>f-m-slash-mod</i>
FMAX	<i>f-max</i>
FMIN	<i>f-min</i>
FNEGATE	<i>f-negate</i>
FOVER	<i>f-over</i>
FROT	<i>f-rote</i>
FROUND	<i>f-round</i>
FS.	<i>f-s-dot</i>
FSIN	<i>f-sine</i>
FSINCOS	<i>f-sine-cos</i>
FSINH	<i>f-cinch</i>
FSQRT	<i>f-square-root</i>
FSWAP	<i>f-swap</i>
FTAN	<i>f-tan</i>
FTANH	<i>f-tan-h</i>
FVARIABLE	<i>f-variable</i>
F~	<i>f-proximate</i>
KEY?	<i>key-question</i>
LOCALS 	<i>locals-bar</i>

LSHIFT	<i>l-shift</i>
M*	<i>m-star</i>
M*/	<i>m-star-slash</i>
M+	<i>m-plus</i>
R/O	<i>r-o</i>
R/W	<i>r-w</i>
R>	<i>r-from</i>
R@	<i>r-fetch</i>
ROT	<i>rote</i>
RSHIFT	<i>r-shift</i>
S"	<i>s-quote</i>
S>D	<i>s-to-d</i>
SCR	<i>s-c-r</i>
SF!	<i>s-f-store</i>
SF@	<i>s-f-fetch</i>
SFALIGN	<i>s-f-align</i>
SFALIGNED	<i>s-f-aligned</i>
SFLOAT+	<i>s-float-plus</i>
SFLOATS	<i>s-floats</i>
SM/REM	<i>s-m-slash-rem</i>
SOURCE-ID	<i>source-i-d</i>
TIB	<i>t-i-b</i>
TIME&DATE	<i>time-and-date</i>
U.	<i>u-dot</i>
U.R	<i>u-dot-r</i>
U<	<i>u-less-than</i>
U>	<i>u-greater-than</i>
UM*	<i>u-m-star</i>
UM/MOD	<i>u-m-slash-mod</i>
W/O	<i>w-o</i>
XOR	<i>x-or</i>
[<i>left-bracket</i>
[']	<i>bracket-tick</i>
[CHAR]	<i>bracket-char</i>
[COMPILE]	<i>bracket-compile</i>
[ELSE]	<i>bracket-else</i>

[IF]	<i>bracket-if</i>
[THEN]	<i>bracket-then</i>
\	<i>backslash</i>
]	<i>right-bracket</i>

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Data Types

Forth has been called an 'untyped' language. This is incorrect; there are a number of data types in Forth, but only limited type-checking. Stack-diagrams used to annotate Forth programs make frequent reference to data types.

The following table lists the data types commonly encountered in Quartus Forth:

Symbol	Data Type	Cells on Stack
<i>? or flag</i>	Boolean flag	1
<i>true</i>	True flag	1
<i>false</i>	False flag	1
<i>char</i>	Character	1
<i>n</i>	Signed number	1
<i>+n</i>	Non-negative number	1
<i>u</i>	Unsigned number	1
<i>n/u</i>	Number	1
<i>x</i>	Unspecified cell	1
<i>xt</i>	Execution token	1
<i>addr</i>	Address in dataspace	1
<i>a-addr</i>	Aligned address in dataspace	1
<i>c-addr</i>	Character-aligned address in dataspace	1
<i>d</i>	Double-cell signed number	2
<i>+d</i>	Double-cell non-negative number	2
<i>ud</i>	Double-cell unsigned number	2
<i>d/ud</i>	Double-cell number	2
<i>xd</i>	Unspecified pair	2
<i>colon-sys</i>	Definition compilation	1
<i>do-sys</i>	Do-loop structures	1
<i>case-sys</i>	CASE structures	1
<i>of-sys</i>	OF structures	1

<i>orig</i>	Control-flow origins	1
<i>dest</i>	Control-flow destinations	1
<i>loop-sys</i>	Loop-control parameters	3 (Return stack)
<i>nest-sys</i>	Definition calls	2 (Return stack)
<i>i*x, j*x, k*x</i>	Any data type	0 or more
Examples of additional punctuation used in stack diagrams in Quartus Forth applications:		
<i>name.</i>	Double-cell value (called 'name')	2
<i>&name</i>	Pointer to a data structure (called 'name')	1
<i>&name.</i>	32-bit (double-cell) pointer to a data structure (called 'name')	2
<i>L.&name</i>	Pointer to a double-cell signed number (called 'name')	1
<i>U&name</i>	Pointer to a single-cell unsigned number (called 'name')	1
<i>name[>byte]</i>	Parameter (called 'name') must be left-shifted 8 bits before passing	1

A Note on the values of TRUE and FALSE

A properly-formed TRUE flag consists of a cell with all bits set to 1, which can be represented in Quartus as the number -1. A FALSE flag is a cell with all bits set to 0. For the purposes of IF, WHILE, and UNTIL, any non-zero value found on the stack is considered true.

To convert a cell to a properly formed TRUE flag, use the code sequence 0= 0=:

The value of a properly-formed TRUE flag is binary 1111111111111111, which is the same as -1 in decimal representation.

The first 0= will convert a 0 to TRUE, non-zero to FALSE (0000000000000000). The second 0= inverts the resulting flag.

This means that

```
1 0= 0= -> TRUE (1111111111111111)
2 0= 0= -> TRUE (1111111111111111)
-17 0= 0= -> TRUE (1111111111111111)
432 0= 0= -> TRUE (1111111111111111) (you get the idea)

0 0= 0= -> FALSE (0000000000000000)
```

Therefore 0= 0= will convert any value to a properly formed TRUE flag.

[Table of Contents](#) - [Quartus Forth Home Page](#)

Startup and Termination Sequences

When Quartus Forth starts, the following actions occur:

1. If launch code is non-zero, exit immediately.
2. Allocate Forth data stack.
3. Initialize data stack pointer.
4. Initialize Forth return stack pointer.
5. Establish dataspace.
6. Initialize dataspace index pointer (**HERE**).
7. Display MainForm & establish Quartus Forth menu.
8. Display 'creating workspace...' message.
9. (Re-)create workspace database.
10. Create image of Quartus kernel in workspace.
11. Initialize codespace index pointer (**CSHERE**).
12. Transfer execution into workspace kernel image.
13. Initialize Forth dictionary with kernel words.
14. Display welcome message.
15. Reset system states (search-order, etc.)
16. Open and interpret *startup.quartus* memo, if present.
17. Call **QUIT**, which starts the interactive session.

On program termination (via (**bye**) or received appStopEvent):

1. Terminate normally. Palm OS closes any open databases and frees dynamic RAM.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

Development Environment:

Quartus Forth is [developed](#) using:

- Pila 1.0 Beta 3 Fluff 6 (68000 assembler); source is pre-processed with M4
- PilRC 2.0a (PalmPilot™ resource compiler)
- Quartus Forth itself; [library](#) modules are written in high-level Forth!

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.

Quartus Forth Manual



[Table of Contents](#) - [Quartus Forth Home Page](#) - [How to Register](#)

History:

1997.7.18	Quartus Forth prototype development begins under the name 'PilotFORTH'.
1997.8.7	First beta test version of PilotFORTH released (0.1.0).
1997.8.26	PilotFORTH (0.4.0) successfully passes John Hayes' <i>coretest</i> .
1998.1.25	Last version of PilotFORTH (0.6.1) released.
1998.2.1	PilotFORTH development stopped.
1998.2.1	Quartus Forth development begins from the ground up.
1998.3.2	Quartus Forth (0.1.1) successfully passes John Hayes' <i>coretest</i> .
1998.6.10	First beta test version of Quartus Forth released (0.2.1B).
1998.12.28	Quartus Forth 1.0.0 released in both evaluation and registered versions.
1999.3.2	Quartus Forth 1.2.0 released.

[Table of Contents](#) - [Quartus Forth Home Page](#)

© 1998, 1999 [Neal Bridges](#). All rights reserved.