

VOYAGER16

Extended Memory

This addendum replaces every memory-size figure in the HP 16c Collector's Edition Owner's Handbook. The way memory is shared, partitioned, addressed, and cleared is unchanged. Three numbers change: the size of the pool, the width of the register count in f MEM, and the longest program the line-number display can hold. The extended pool sits inside the calculator's ordinary 2 KB VoyagerSave image, so a backup of Continuous Memory includes it.

Disclaimer. This is unofficial, experimental firmware and documentation. It is not made, supported or endorsed by HP Inc. or its licensees, and it is provided as is, without any warranty. Installing it is at your own risk and may void the manufacturer's warranty. Flashing can fail, and a failed or interrupted flash, low batteries or an incompatible unit can leave the calculator unusable. Back up your programs and data first, because installing firmware or changing settings can erase memory. Results have not been certified; check important calculations independently. Nobody involved accepts liability for damage, data loss or any consequence of using it. HP and HP 16c are trademarks of HP Inc., used only to identify the compatible hardware.

How to tell

Reset Continuous Memory as on page 20 (turn the calculator off, hold ON, press and hold –, release ON, then –). Pr Error appears. Press any key. The calculator is then in hexadecimal, 2's complement, word size 16, with program memory clear. Press f MEM and hold MEM.

Keystrokes	Display	
	P-0 r-0616	<i>This calculator. The handbook shows r-101.</i>

To see the largest register count, set the word size to 4 and press f MEM again. The display reads P-0 r-2464. The handbook's maximum, at the same word size, is 406.

A word size of 4 cannot hold a two-digit number, so you cannot key the return directly. Key 0 f WSIZE first. That sets the word size to 64, as the note on page 32 says. You are still in hexadecimal, and hexadecimal 10 is decimal 16, so key 10 f WSIZE. If you have pressed DEC, key 16 f WSIZE instead. Keying 16 while the calculator is still in hexadecimal sets the word size to 22.

The keyboard self-test (hold ON, press ÷) of this ROM ends by displaying 16, as on the original calculator. Appendix C of this handbook is a different test (hold g and ENTER, press ON). It does not show that legend. The fMEM lines above are the check for this book. This addendum matches CE checksum B7B7 (Voyager16-ext-path-c-trim4-show-B7B7.bin). fMEM counts registers as the handbook does: r-N means N registers (R0 through R(N-1)). After a clear at word size 16 it shows r-0616, that is R0 through R615.

Where the printed figures change

Handbook	Printed	This calculator
Page 62, the pool	203 bytes	1,232 bytes
Page 64, word size 16, no program	101 registers	616 registers
Page 64, word size 4, no program	406 registers	2,464 registers
Page 65, f MEM after reset	P-0 r-101	P-0 r-0616
Page 65, f MEM after one program line	P-6 r-098	P-6 r-0612
Page 65, range of RRR	0 to 406, three digits	0 to 2,464, four digits
Page 66, indirect registers at word size 16	up to 69	up to 584
Page 70, highest 4-bit register in the example	R326	R2463
Longest program	203 lines	999 lines

f MEM counts registers as the handbook does: r-N means R0 through R(N-1). r-0616 at word size 16 means R0 through R615, and r-2464 at word size 4 means R0 through R2463.
Every other rule in Section 6 still stands, including the worked example on page 67 (doubling the word size concatenates neighboring registers) and the Index register.

Memory allocation

Data storage and program lines still share one pool. The pool is 1,232 bytes, not 203. At Continuous Memory reset the whole pool is data storage. Program lines still take it away in segments of seven bytes, from the highest numbered register downward. Data in a register is lost when that register is converted. Program memory returns to data storage only when you delete instructions, with f CLEAR PRGM or with a Continuous Memory reset.

1,232 bytes

Program memory	Data storage ... R ₂ R ₁ R ₀
----------------	---

Movable partition. Program memory grows to the right, from the highest register.

Program lines recorded*	Program memory	Storage memory
0	0 bytes	1,232 bytes
1 to 7	7	1,225
8 to 14	14	1,218
15 to 21	21	1,211
...	...	...
993 to 999	1,001	231

* One line equals one byte. Both allocations are always multiples of seven.

The calculator will not record a line numbered above 999. The line number is three digits, the same field described on page 73. Attempting to enter line 1000 gives Error 4. That is a hard cap: the firmware stops at 999 program lines.

Program length. *The hard cap is 999 program lines (Error 4 on attempting line 1000). As the program grows, leftover storage shrinks and STO / RCL of a register whose bytes have been taken returns Error 3, as in the handbook.*

Storage register size

The rule on page 63 does not change. A register is the smallest multiple of four bits that can hold the current word size. In Floating-Point Decimal mode the word size, and the register, are 56 bits (seven bytes). The number of registers is the bytes still in the pool, divided by the bytes in one register. A leftover half register is not usable, exactly as $203 \div 2$ gives 101 on page 64 and not 101.5.

With program memory clear the counts are:

Word size	Bytes each	Registers	f MEM
4 bits	½	2,464	P-0 r-2464
8	1	1,232	P-0 r-1232
16	2	616	P-0 r-0616
32	4	308	P-0 r-0308
64	8	154	P-0 r-0154
Floating point (56)	7	176	P-0 r-0176

The Registers column is what STO and RCL accept: whole registers that fit in the first 1,232 bytes of the pool (R0 through R615 at 16 bits, R0 through R2463 at 4 bits). f MEM matches that count at every word size: r-N means N usable registers,

the last one being R(N-1).

The f MEM column is what the calculator displays with program memory clear. With one program line recorded, word size 64 shows r-0153 (R0 through R152 usable).

The example on page 65

Initial memory, word size 16. One recorded program line converts seven bytes from the 1,232-byte pool (1,225 bytes left for storage). The calculator then displays r-0612 (R0 through R611).

Keystrokes	Display	
ON / -	0 h	Continuous Memory reset. Word size is 16.
f MEM	P-0 r-0616	Hold MEM. R0 through R615 are usable. Release returns to 0 h.
g P/R	000-	Program mode.
1	001- 1	One program line.
f MEM	P-6 r-0612	Six lines to go before the next seven bytes. R0 through R611 are usable.
f CLEAR PRGM	000-	Clears program memory.

Reading f MEM

Press f MEM. Hold MEM to keep the display. The form is still

P-B r-RRRR

B is the number of program lines you can still record before another seven bytes are taken from storage. $0 \leq B \leq 6$. RRRR is the number of storage registers available now, always in base 10, whether or not the calculator is in hexadecimal. $0 \leq RRRR \leq 2464$. The field is four digits, zero-filled (r-0616, r-0308), because 2,464 does not fit in the three-digit field on page 65. Ten display places hold it: the four-digit line fills the window.

Addressing

Naming a register has not changed. STO and RCL reach 32 registers directly: 0 through 9, A through F, .0 through .F. Every register above R31 is reached only with the Index register, STO I then STO i (and the same with RCL and x≠(i)). The f before I or i may still be omitted after STO, RCL, or STO arithmetic.

At word size 16, registers above R31 are reached only through the Index. STO and RCL work through R615; R616 returns Error 3. That is 584 unnamed registers (R32 through R615), matching f MEM r-0616 after Continuous Memory reset. At word size 4 the parallel range is R32 through R2463 (2,432 of them). The highest registers are still the first converted to program lines, so keep data in the lowest registers, as page 66 says. Addressing a register that does not exist, or whose bytes have been taken by the program, is still Error 3.

The Index register itself is still 68 bits, still not converted to program lines, and still outside the pool. Storing a number into RI and recalling it into a shorter word size still keeps the least significant bits.

Storing in the highest register

This is the example on page 70, aimed at the new top of memory. R2463 exists only at a word size of 4. The number 2463 does not fit in four bits, so it has to be parked in the Index register at a wide word size first. Assume the calculator is already at word size 4.

Keystrokes	Display	
DEC0fWSIZE	0 d	Decimal, then word size 64, so 2463 can be keyed.
2463	2463 d	
STOI	2463 d	Parks the address in RI.
4fWSIZE	-1 d	Word size 4. 2,464 registers exist (R0 through R2463). 2463 in four bits is 1111, which is -1 in 2's complement.
3STOi	3 d	Stores 3 in R2463. R2464 returns Error 3.
CLx	0 d	Clears the display.
RCLi	3 d	Recalls R2463.

The display after 4 f WSIZE is the previous X truncated to four bits, not a confirmation of the address. The address is the value that stayed in RI. The same two changes of word size are required for any register above 15 when the word size is 4, which is already how R326 is reached on page 70.

Continuous Memory and VoyagerSave

The extended registers occupy Nut addresses 20₁₆ through CF₁₆ inside the same 2 KB image that VoyagerSave reads and writes over USB. A dump of Continuous Memory therefore includes them. Soft reset usually keeps the pool; a Continuous Memory reset clears it and returns f MEM to r-0616 at word size 16.

The rest of the handbook

Section 6 is where the numbers are worked. Four other sentences in the Collector's Edition handbook still quote the old size. Read them as follows.

Where	Printed	Read instead
Introduction, page 2	203 bytes, up to 203 program lines	1,232 bytes, up to 999 program lines
Section 7, page 77	203 bytes available for storage and program lines	1,232 bytes. Hard cap 999 lines
Section 8, page 82	GTO . nnn, range 000 to 203	Still three digits. Range 000 to 999
Appendix A, page 97, Error 4	Attempted to load more than 203 program lines	Attempted to load more than 999 program lines

GTO . nnn is still not programmable, and nnn is still three digits. It moves to an existing line. The top of that range is now 999, not 203. A line number that does not exist is still Error 4, whether the missing line is inside that range or past it.

Error 4 keeps its other sentence: a label or line number called is nonexistent. The load limit is 999 program lines (Error 4 on attempting line 1000). Error 3 is unchanged. A named register that does not exist, or whose bytes have been taken by the program, is still Error 3.

Nothing else in the handbook changes its meaning. Word size, complement modes, the stack, flags, labels 0 through 9 and A through F, and the four-deep subroutine limit are as printed. The Index register is still 68 bits and still not part of the pool.

What does not change

- Changing the word size still reslices the same bits. The example on page 67, where $R_0 = 1234$ h and $R_1 = 5678$ h become one 32-bit R_0 of 56781234 h, is still correct. Restore the original word size and the original registers are there.
- Switching between Integer and Floating-Point Decimal mode still does not clear the registers. The bits stay. Their meaning changes until you return to the mode you stored them in.
- f CLEAR REG still clears every storage register and does not clear the stack or LAST X. A Continuous Memory reset still clears the registers, the stack, and the program, and still returns the word size to 16. After that reset, f MEM shows r-0616, not r-2464.
- Labels are still 0 through 9 and A through F. GTO . nnn is still a three-digit line number. Nothing in this memory map adds a fourth program-address digit.

Service menu

Turn the calculator off. Hold g and ENTER, press ON, and release. The display shows 1.L 2.C 3.H. Press a number to open an item. For items 5 to 8, the first press shows the current value and each further press of the same key steps to the next value. ON leaves the menu. Items 5 to 8 keep their values while the batteries are in. If the batteries are removed, they return to the defaults shown below. (f ENTER ON is the USB backup mode, not this menu.)

Key	Item	Values
2	Checksum	B7B7, then the date 2026-09-25
5	Speed	4, 12, 48 MHz (default 4)
6	Same-key lockout	50, 80, 100, 120 ms (default 120)
7	Key press confirmation	10, 20, 30 ms (default 10)
8	Key release confirmation	10, 20, 30, 50 ms (default 30)
9	Stopwatch	Opens the stopwatch
E	Battery reading	One reading, for example bAtt 3.02 (volts)

How the keyboard is read

The keyboard is read every 10 ms. A key counts as pressed only after it has read closed for the press confirmation time (item 7), and as released only after it has read open for the release confirmation time (item 8). A brief bounce or dropout of the contacts while a key is held is therefore ignored. In addition, the same key pressed again within the lockout time (item 6) is ignored. Longer confirmation times reject more bounce, but a key must then be held a little longer, or released a little more clearly before it is pressed again.

Battery reading

In the service menu, press E. The calculator takes one reading of the battery voltage and shows it, for example bAtt 3.02. Press E again for a new reading. If a reading fails, the display shows bAtt - - -. The value is measured under load and may differ by about 0.1 V from a multimeter.

The low-battery indicator (*) uses fixed thresholds: it turns on after five low readings in a row (below 2.5 V, one reading per keypress) and off again after 80 good readings in a row.

Stopwatch speed factor

The stopwatch (item 9) multiplies the elapsed time by a speed factor with four decimals, from 0.5000 to 2.0000 (default 1.0000). In the stopwatch, + and – change the factor by 0.0001 (0.01 %). ENTER shows the factor for about a second without changing it, for example SCF1.0885; digits you have typed are kept. To set a value directly, type the difference from 1 in hundredths of a percent and press + or –: 885 then + gives 1.0885, 885 then – gives 0.9115. The factor is kept while the batteries are in.