

HP-71

Owner's Documentation

Addendum

This addendum contains information on rare HP-71 situations that can cause problems when using an HP-71 equipped with a version 1AAAA operating system. To determine the version of your operating system, execute `VER#` (type `VER#`, then press `END LINE`).

While you probably won't encounter the situations described below, we want you to know what they are and how to avoid the problems they can produce.

Less than 40% of the situations described below can occur with a standard HP-71. The remaining situations can occur only with a card reader, an HP-IL interface, or a Math Pac installed.

Note: A magnetic card containing a file named `CARDFIX` is included in the package with your HP-71. This magnetic card contains code designed to correct certain potential problems that can occur when operating the HP 82400A Card Reader. If you acquire an HP 82400A Card Reader, you should copy this card file into memory and keep it in memory while your card reader is installed.

Problems That Can Occur With A Standard HP-71 System

Using Random Access To Store a String in a DATA File

This caution does not apply to sequential access.

In the *HP-71 Owner's Manual*, pages 258-260. When using *random access* to store a string in a DATA file, make sure the string data does not exactly fill one record. If the string does exactly fill one record, the next record is destroyed.

To avoid this problem, make sure the record length is *greater* than the number of characters in the string plus 3 (since 3 bytes of overhead are stored with the string).

Example:

```
CREATE DATA FILE1,2,8 END LINE
```

Record length is 8 bytes.

```
ASSIGN#1 TO FILE1 END LINE
```

```
PRINT#1,1;"ABC" END LINE
```

```
PRINT#1,0;"1234" END LINE
```

This doesn't destroy record 1.

```
PRINT#1,0;"12345" END LINE
```

This *does* destroy record 1.

Calling a Subprogram That Creates a File in RAM

In the *HP-71 Owner's Manual*, pages 205-210. If a subprogram creates a file in RAM or copies a file into RAM, there are two situations that can alter memory. Each situation concerns any program in RAM that calls the subprogram (using the `CALL` statement).

Situation 1. If an `END` statement is not the last statement in the calling program, program execution can fail to terminate and memory can be altered.

To avoid this problem, ensure that `END` is the last statement in the calling program.

Situation 2. If you `RENUMBER` the calling program after the program is run, memory can be altered.

To avoid this problem, execute `EDIT filename`, where *filename* is the name of the calling program, before executing `RENUMBER`.

Example: Program `MAIN` calls a subprogram named `S1` located in another file. This subprogram, when executed, creates a `DATA` file in RAM.

Subprogram.

```
10 SUB S1
   :
50 CREATE DATA TEMP,2,8
   :
100 END SUB ! END OF SUBPROGRAM
```

Main Program.

```
10 ! PROGRAM NAME: MAIN
   :
50 CALL S1
   :
100 END ! END OF MAIN PROGRAM
```

- Make sure `END` is the last statement in `MAIN` to avoid possible memory alteration.
- After running `MAIN`, execute `EDIT MAIN` before executing `RENUMBER` to avoid possible memory alteration.

Single-Stepping Through . . . GOSUB ELSE . . .

This caution does not apply to a running program.

In the *HP-71 Owner's Manual*, pages 168-171. When a GOSUB immediately precedes an ELSE, single-step fails when the RETURN in the subroutine is encountered. Single-stepping through the RETURN will cause a jump to an unexpected place in the program and can result in a memory alteration.

To avoid this problem, put a harmless statement before ELSE, like BEEP 0.

Example: Single-stepping through the following program fails when the RETURN statement is encountered.

```
10 A=1
20 IF A=1 THEN GOSUB 50 ELSE DISP "HI"
30 DISP PI
40 STOP
50 RETURN
```

If you wish to single-step through line 20, replace line 20 by

```
20 IF A=1 THEN GOSUB 50 @ BEEP 0
   ELSE DISP "HI"
```

Using DATE\$ In An Expression

In the *HP-71 Owner's Manual*, pages 90-91. DATE\$ can cause a Data Type error if it is not the first operand or parameter appearing in an expression.

To avoid this problem, assign DATE\$ to a string variable and then use the string variable in the expression.

Example:

B\$="ABC" & DATE\$	END LINE	Can cause a Data Type error.
A\$=DATE\$ @ B\$="ABC" & A\$	END LINE	Does not cause a Data Type error.
B=30 + LEN(DATE\$)	END LINE	Can cause a Data Type error.
A\$=DATE\$ @ B=30 + LEN(A\$)	END LINE	Does not cause a Data Type error.

Changing the OPTION BASE Setting Within a User Defined Function

In the *HP-71 Owner's Manual*, pages 218-222. If a user-defined function changes the OPTION BASE setting, do not assign the result of the user-defined function to a nonexistent array element. If you attempt to assign the result to element 10 of the array, user memory can be altered. If the assignment is made to another element, the result will be assigned to the wrong element of the array.

To avoid this problem, make sure the array already exists before assigning the result of the user-defined function to any of the array's elements.

Example: This program defines function F that includes an OPTION BASE statement. Keystroke sequences follow the program showing the wrong and right way to assign the result of this function to a previously nonexistent array element.

```
10 DEF FNF
20 OPTION BASE 1
30 FNF=5
40 END DEF
```

Wrong:

```
OPTION BASE 0 [END LINE]
```

```
DESTROY A [END LINE]
```

```
A(10)=FNF [END LINE]
```

Can alter user memory.

Right:

```
OPTION BASE 0 [END LINE]
```

```
DIM A(10) [END LINE]
```

```
A(10)=FNF [END LINE]
```

Creates array A, ensuring problem-free assignment of FNF to an element of A.

Does not alter user memory.

Problems That Can Occur With the HP-IL Interface Installed

Attempting Transformation of Tape File

In the *HP 82401 HP-IL Interface Owner's Manual*, page 224. In the *HP-71 Owner's Manual*, pages 160-161. There are two situations involving transformation of a tape file that can cause problems.

Situation 1. Attempting an out-of-place transformation of a BASIC file on magnetic tape into a TEXT file in RAM can alter memory.

To avoid this problem, copy the BASIC file into RAM first and then transform it in-place.

Example: Assume FILE1 is a BASIC file on tape.

Wrong:

```
TRANSFORM FILE1:TAPE INTO TEXT    This can alter memory.  
FILE2 END LINE
```

Right:

```
COPY FILE1:TAPE TO FILE2 END LINE    Copies the file on tape (FILE1) to a file in  
main RAM (FILE2).  
TRANSFORM FILE2 INTO TEXT END LINE    Does not alter memory.
```

Situation 2. Attempting an out-of-place transformation of a file from magnetic tape into a RAM file of the same type results in an Invalid Arg error and no copy takes place.

To avoid this problem, use COPY instead of TRANSFORM in this situation.

Example: Assume FILE1 is a BASIC file on tape.

Wrong:

```
TRANSFORM FILE1:TAPE INTO BASIC    Gives an Invalid Arg error.  
FILE2 END LINE
```

Right:

```
COPY FILE1:TAPE TO FILE2 END LINE    Does not produce an error.
```

Chaining a File From Tape

In the *HP 82401 HP-IL Interface Owner's Manual*, page 79. CHAIN closes all files when a file is chained into RAM from tape.

To avoid this problem, ensure that the chained program reopens any files used by that program.

Card Reader Problems Cured By Magnetic Card Program

Note: A magnetic card containing a file named CARDFIX is included in the package with your HP-71. This magnetic card contains code designed to correct the two problems described below that can occur when operating the HP 82400A Card Reader. If you acquire an HP 82400A Card Reader, you should copy this card file into memory and keep it in memory while your card reader is installed.

Copying a Program From a Magnetic Card

In the *HP-71 Owner's Manual*, page 291. A file whose length is 0, 1, 2, or 3 bytes short of a multiple of 650 bytes can be copied to a card but cannot be copied from the card to an HP-71 that does not have `CARDFIX` in memory.

To avoid this problem, ensure that the `CARDFIX` program is in HP-71 memory. (`CARDFIX` is on the magnetic card included in the HP-71 package.) If `CARDFIX` is not in memory, follow this procedure before copying a BASIC file to a card: Use `CAT` to determine the file's length in bytes. If this length plus 6 overhead bytes is 0, 1, 2, or 3 bytes short of a multiple of 650, make the file at least 4 bytes longer (add a comment) before copying it to a card.

Copying a TEXT File From a Magnetic Card

In the *HP-71 Owner's Manual*, page 291. When a TEXT file is copied from a magnetic card to an HP-71 that does not have `CARDFIX` in memory, memory can be altered.

To avoid this problem, ensure that the `CARDFIX` program is in HP-71 memory. (`CARDFIX` is on the magnetic card included in the HP-71 package).

Problems That Can Occur with the Card Reader Installed

Note: The two problems described below are *not* cured by the `CARDFIX` program. (The `CARDFIX` program is on the magnetic card included in the HP-71 package.)

Using `CHAIN :CARD` in a Program

In the *HP-71 Owner's Manual*, page 289. When `CHAIN :CARD` is encountered in a running program, the HP-71 displays `READ:Align then ENDLN`. If you then press `[ATTN]` and attempt to continue the program by executing `CONT` or by pressing `[f][CONT]` or `[f][SST]`, you can alter memory.

To avoid this problem, do *not* continue program execution in this situation. What you should do, after pressing `[ATTN]`, is execute `EDIT filename` to establish a current file and to turn off the `SUSP` annunciator.

Using `PROTECT` and `UNPROTECT` in a Program

In the *HP-71 Owner's Manual*, pages 289, 292. When `PROTECT` and `UNPROTECT` are encountered in a running program, they halt the program and display the `SUSP` annunciator.

To continue program execution, simply press `[f][CONT]`.

Problems That Can Occur With the Math Pac Installed

Using a Complex Subscript for an Array

In the *HP 82480A Math Pac Owner's Manual*, topic "Array Types" in "How To Use This Manual." Using a complex subscript for an array will alter memory if the Invalid Operation trap (IVL) is set to level 2. For example, if **A** is a one-dimensional array, then referencing **A((2,3))** will alter memory if the trap setting for IVL is set to 2 (TRAP(IVL,2)).

To avoid this problem, do not use complex subscripts for arrays.

Using READ# To Read Data Into a Complex Variable

This caution only applies to DATA files; there is no similar restriction for SDATA files.

In the *HP 82480 Math Pac Owner's Manual*, topic "Complex Number Operations" in section 3. In the *HP-71 Owner's Manual*, pages 254-257, 260. You can alter memory when using READ# to read data from a DATA file into a variable of complex type. This can occur if the real and imaginary parts of the complex number are in different records.

To avoid this problem, ensure that the real and imaginary parts of each complex number are in the same record.

Using an Expression Like (1,2,3,4,5,6,7,8) in CALC Mode

In the *HP 82480 Math Pac Owner's Manual*, topic "Complex Number Operations" in section 3. Using an n -tuple in an expression in CALC mode will alter memory if $7 < n < 16$. (An example of an n -tuple where $n = 2$ is (5,3)). For instance, (1,2,3,4,5,6,7,8) END LINE will alter memory if CALC mode is on.

To avoid this problem, do not use n -tuples other than those where $n = 2$. This causes no restriction on HP-71 power, since for $n \neq 2$, n -tuples have no meaning to the HP-71. The HP-71 uses n -tuples with $n = 2$ for entering and displaying complex numbers.



**HEWLETT
PACKARD**

**Portable Computer Division
1000 N.E.Circle Blvd., Corvallis, OR 97330**

**European Headquarters
150, Route Du Nant-D'Avril
P.O. Box, CH-1217 Meyrin 2
Geneva - Switzerland**

**HP-United Kingdom
(Pinewood)
GB-Nine Mile Ride, Wokingham
Berkshire RG11 3LL**