

HELP for **all** commands
incorporated into the O.S.

Full Command and Function Reference
inside your HP 50g.

It is the manual **inside**
your calculator !

- Help for 201 built-in CAS commands.
- Help for 563 built-in non CAS commands.
- Total help for 764 built-in commands.
- Context sensitive help from the stack
- "Hyperlinked" navigation to related commands.
- Detailed access for every commands
- Supports all built in fonts.
- Detailed description of the related flags used by the command / function
- Detailed description of all flags.
- Detailed description of all keyboard shortcuts.
- Detailed description of all reserved variables.
- Detailed description of the Computer Algebra System.

The new help in detail.

PARTFRAC: ↓↑
 Performs partial fraction decomposition on a fraction
 1: $\frac{2x^2}{x^2-1}$
PARTFRAC **ENTER**
 1: $2 + \frac{1}{x-1} - \frac{1}{x+1}$
 Flags: -3CF -17SF
 -105CF
 Type: Command
 Access: **RS** **ALG** /
S **ARITH** **POLYNOMIAL**
 See: **PROPFAC**
EXIT **ECHO** **PROPF** **MAIN**

Annotations:

- Name of the command / function
- Detailed description of the operation, combined text and graphic area with unlimited length
- RPN or ALG example
- RPN or ALG result
- Flag settings affecting the operation of the function or command
- Access to the command / function
- Related commands / functions

Use the help entry for **HELP** to learn how to use the new help.

HELP: ↓↑
 Shows help for the catalog
[ECHO] push example onto stack
[MAIN] help list
[APPS] show additional help links, if any
[MODE] toggle between

shortcuts
[X] reserved variables
[EE] using the CAS
[+/-] toggle current/mini font
[_] toggle menu bar
[←↑↓→] move screen, if applicable

ALG/RPN
[VAR] toggle stack diagram
[SYM] one page up
[COS] one page down
[] top of text
[<=] bottom of text
[HIST] related flags
[x^y] all flags
[EVAL] keyboard

Used abbreviations:
 α = ALPHA key
 α+ = hold ALPHA key
 RS = right shift
 RS+ = hold right shift
 LS = left shift
 LS+ = hold left shift
 CF = clear flag
 SF = set flag

If an example errors out then please check if your flag settings are correct for it.

Type: Command
Access: [U][O][O][O] [UNX][O]
See: CASCMD

Get help for every command directly from the stack.

```
RAD XYZ HEX R= 'X' 127276
{HOME}
6:
5:
4:
3:
2:
1: "UFACT"
HELP
[EDIT VIEW STACK RCL PURGE CLEAR]
```

```
UFACT: ↓↑
Factor Unit Command
Factors the level 1
unit from the unit
expression of the
level 2 unit object.
2: 1_W
1: 1_N
UFACT [ENTER]
[EXIT ECHO CONVE UBASE UVAL MAIN]
```

```
RAD {HOME} 127276
Help on:
UFACT
UFL1+MINIF
UNASSIGN
UNASSUME
UNBIND
UNPICK
UNROT
UNTIL
[CANCL OK]
```

```
1: 1  $\frac{Nm}{s}$ 
Type: Command
Access: [U][S] [CONVERT]
UNITS TOOLS / [RS]
[UNITS] TOOLS
See: CONVERT UBASE
UVAL →UNIT
[EXIT ECHO CONVE UBASE UVAL MAIN]
```

Jump directly to any related command (Key [APPS])...

```

1: 1_ N/m
1_ s
See:
Type: CONVERT
Access: UBASE
Unit: UVAL
Unit: →UNIT
See: CONVERT UBASE
UVAL →UNIT
[CANCL] [OK]

```

```

CONVERT: ↓↑
Convert Units Command
Converts a source unit
object to the
dimensions of a target
unit.
The source and target
units must be
compatible. The number
[EXIT] [ECHO] [UBASE] [UFACT] [UVAL] [MAIN]

```

```

CON: Help on: ↓↑
Cor: CONVERT nd
Cor: CORR nit
obj: COS get
dir: COSH et
uni: COV ber
The: CR
uni: CRDIR
com: CROSS
[CANCL] [OK]

```

```

CON: Help on: ↓↑
Cor: UNROT nd
Cor: UNTIL nit
obj: UPDIR get
dir: UTPC et
uni: UTPF ber
The: UTPD
uni: UTPT
com: UVAL
[CANCL] [OK]

```

```

UVAL: ↓↑
Unit Value Function
Returns the numerical
part of a unit object.
1: 2.300000_N
UVAL [ENTER] 1: 2.30
Flags: -3
Type: Function
[EXIT] [ECHO] [CONVE] [UBASE] [UFACT] [MAIN]

```

```

UVAL [ENTER] 1: 2.30
Flags: -3
Type: Function
Access: [ES] [CONVERT]
UNITS TOOLS / [RS]
[UNITS] TOOLS
See: CONVERT UBASE
UFACT →UNIT
[EXIT] [ECHO] [CONVE] [UBASE] [UFACT] [MAIN]

```

... or move to any other help entry.

Get stack diagram for every command. (Key [VAR])

UVAL:

X_U → X
'sym' → 'UVAL(sym)'

EXIT ECHO CONVE UBASE UFACT MAIN

UBASE:

X_U → y_base-units
'sym' → 'UBASE(symb)'

EXIT ECHO CONVE UFACT UVAL MAIN

Use [VAR] to toggle between the help and the stack diagram.

Push example on the stack. (Key [ECHO])

Converts a unit object
to SI base units.

1: 1_N

UBASE [ENTER]

1: $1 \frac{\text{kg} \cdot \text{m}}{\text{s}^2}$

Flags: -3

Type: Function

EXIT ECHO CONVE UFACT UVAL MAIN

RAD XYZ HEX R= 'X' 113855
{HOME}

6:

5:

4:

3:

2:

1:

1_N

UBASE↵

EDIT VIEW STACK RCL PURGE CLEAR

RAD XYZ HEX R= 'X' 112659
{HOME}

5:

4:

3:

2:

1:

$1 \frac{\text{kg} \cdot \text{m}}{\text{s}^2}$

EDIT VIEW STACK RCL PURGE CLEAR

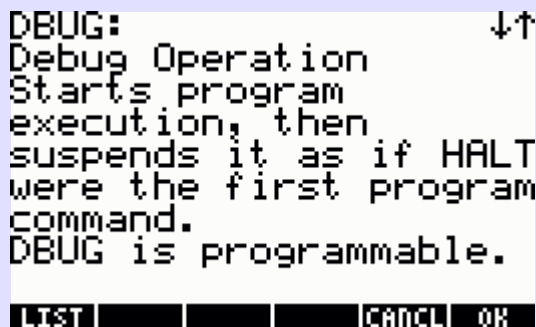
Help is available in every Choose Box menu ...



... and from the catalog of all functions.

Help is available from every environment where the catalog of all functions is available.

For example in the programming environment ...



... so you can look up every command while you are programming.

And then paste the command you need into the programming environment.

Supports all built in fonts.

```

ABS:                                     +†
Absolute Value Function
Returns the absolute value of its
argument
ABS has a derivative (SIGN) but
not an inverse.
In the case of an array, ABS
returns the Frobenius (Euclidean)
norm of the array, defined as the
square root of the sum of the
squares of the absolute values of
all n elements. That is:
EXIT ECHO NEG SIGN MAIN

```

```


$$\sqrt{\sum_{i=1}^n |z_i|^2}$$

1: -1
ABS [ECHO] 1: 1
Flags: -3
Type: Function
Access: [ECHO] [MATH] / [ECHO] [MATH]
EXIT ECHO NEG SIGN MAIN

```

See detailed description of the related flags (Key [HIST]) used by the command or the function ...

```

TAN:                                     ↓↑
Tangent Analytic
Function
Returns the tangent of
the argument.
For real arguments,
the current angle mode
determines the
number's
EXIT ACOS ASIN ATAN MAIN

```

```

differentiate expressions
containing TAN with a unit
object, the angle mode must be
set to radians (since this is a
"neutral" mode).

Flags: -3 -17 -18 -22
Type: Analytic Function
Access: [ECHO]
See: ACOS ASIN ATAN ACOSH ASINH
ATANH COS SIN COSH SINH TANH ISOL
EXIT ACOS ASIN ATAN MAIN

```

```

System Flag -3
Numerical Results
Clear:
Functions with
symbolic arguments,
including symbolic
constants, evaluate to
symbolic results.
Set:
Functions with

```

```

symbolic arguments,
including symbolic
constants, evaluate to
numbers.

```

```

System Flag -17
Trigonometric Angle
Mode
Clear:
Degrees or Grads

```

```

constants, evaluate to numbers.

System Flag -17
Trigonometric Angle Mode
Clear:
Degrees or Grads
Set:
Radians

System Flag -18
Angle Mode
Clear:
Degrees (Flag -17 clear)

```

```

Degrees (Flag -17 clear)
Set:
Grads (Flag -17 clear)

System Flag -22
Infinite Result Exception
Clear:
Infinite result exception treated
as an error.
Set:
Infinite result exception returns
± 9.999999999999999E499 and sets Flag
-26.

```

... or view all flags. (Key [y^x])

ABOUT FLAGS

Flags are Mode settings and Mode indicators. One can set, clear or test a system or user flag by specifying its number as argument of the UserRPL Flag commands.

SF: Sets the specified flag.
CF: Clears the specified flag.
FS?: Tests whether the flag on level 1 is set. Returns 1 if true, 0 otherwise.

User flags are identified by positive numbers 1 to 128. You can set, clear, and test all flags, although certain flags are used for specific purposes by the CAS and should not be altered. The default state of the flags is clear - except for flags -17, -27, -34, -90, -95 and -128 and the binary integer math flags (flags -5 through -12).

...

All variables are real variables.

USER FLAGS

User Flag +60

Units Type

Clear:

CONST returns values in SI units.

FS?C: Tests whether the flag on level 1 is set, returning a corresponding test result, then clears it.

FC?: Tests whether the flag on level 1 is clear. Returns 1 if true, 0 otherwise.

FC?C: Tests whether the flag on level 1 is clear, returning a corresponding test result, then clears it.

System flags are identified by negative numbers -1 to -128.

SYSTEM FLAGS

System Flag -1

Principal Solution

Clear:

Symbolic commands return a result representing all possible solutions.

Set:

Symbolic commands return only the principal solution.

System Flag -2

Set:

CONST returns values in English units.

User Flag +61

Units Usage

Clear:

CONST uses units.

Set:

CONST uses no units.

Detailed description of all keyboard shortcuts. (Key [EVAL])

```
Keyboard Shortcuts

[<=] If no command
line, performs DROP in
RPN mode. If hold dur-
ing system reset it
prevents all added
libraries from being
attached AND prevents
EXIT ECHO COME GANNA PERM MAIN
```

noted that keystrokes also cause a system interrupt, so be sure NOT to press any key between pressing [ON]+[9] and the next repeating alarm, or the alarm will come due and continue repeating.

[ON]+[F1]+[F6] Clears the calculator's RAM, returning everything to the factory state. Anything in port 2 (FLASH ROM) Anything in port 2 (FLASH ROM)

Detailed description of all reserved variables. (Key [VX])

```
System Reserved Variables

ENTER
is the vectored ENTER
pre-processor. It is
active when flags -62
and -63 are set. When
ENTER is pressed, the
command line is placed
on the stack as a
```

A list of the names of variables that some CAS operations treat as real numbers when complex Mode is set. If additional assumptions are made on any variables, these are included here. By default the list is {X, Y, t, S1, S2}.

VX
A name or list of names of the current CAS variable or variables. Default value is X.

EXIT ECHO COME GANNA PERM MAIN

Detailed description of the Computer Algebra System.
(Key [EEX])

```
Computer Algebra System

CAS settings are
selected using the
"CAS MODES" input
form.
Selecting a mode is
equivalent to setting
or clearing one of the
EXIT ECHO COME GANNA PERM MAIN
```

If a CAS operation gives an unexpected result or message, check the list of points given in the section on CAS settings. Some problems can be caused by unexpected settings, so it can be helpful to reset all CAS settings to their default values, with the CASCFG command, or with the [RESET] key in the CAS settings menu.

EXIT ECHO COME GANNA PERM MAIN

Suggestions, criticism and/or improvements ?

<mailto:Software49G@gmx.de>

Please report any bugs
and/or spelling mistakes
and/or anything else
notable to me.
Thanks.