

**Calc**

**COLLABORATORS**

	<i>TITLE :</i> Calc		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		September 19, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Calc</b>	<b>1</b>
1.1	Index . . . . .	1
1.2	General intro . . . . .	2
1.3	Program Paradigm . . . . .	2
1.4	Installation & system requirements . . . . .	3
1.5	Using the program . . . . .	3
1.6	Legal crap . . . . .	5
1.7	Bug list . . . . .	6
1.8	History & future . . . . .	6
1.9	Political Plug . . . . .	7

---

# Chapter 1

# Calc

## 1.1 Index

\*\*\*\*\*

Calc v2.0

01/13/94

by Sean Russell

MCMXCIV Apocalypse Productions

\*\*\*\*\*

Index

- 0 - General intro
  - 1 - Paradigm
  - 2 - Installation & System requirements
- 3 - Use
- Using the program
- 4 - Legal crap
  - 5 - Bug list
  - 6 - History and future
- 7 - Political Plug

\*\*\*\*\*

## 1.2 General intro

General intro

I suppose this program is getting silly enough to require a manual...

Calc is another calculator program for public screens. I wrote this for four reasons:

- 1: I like programming, and I'm desperate for program ideas.
- 2: It was a simple project (or, at least, it started simple).
- 3: I needed to exercise my C++ programming skills.
- 4: I'm sick of the various calculator programs that exist for the Amiga. They're all normal algebraic entry, and I'm convinced that algebraic entry is for masochists.
- 5: I had just gotten MaxonC++, and have to try it out =-).

Calc is a RPN (Reverse Polish Notation) calculator, with a stack. This means that instead of the usual algebraic form of  $(a + b)$ , RPN uses the form of  $(a b +)$ , which is more intuitive to stacks, and generally easier to deal with, once you know it. All Hewlett Packard calculators use RPN, and HPs are (as everyone knows) the only serious calculators in existence =-).

You can contact me only through snailmail:

Sean Russell  
Claude-Lorraine-Str 31  
81543 Mnchen  
GERMANY

See the

Bug list  
, if nothing else.

\*\*\*\*\*

## 1.3 Program Paradigm

Program Paradigm

I tried to make this as user friendly as possible. My main concerns were that (a) it worked, (b) it was RPN, and (c) it had a stack. So far, these requirements have been met.

Calc takes input and throws it on a stack. It allows the user to preform functions on the stack (in version 2, I think most of the most popular funcs are present).

\*\*\*\*\*

---

## 1.4 Installation & system requirements

### Installation & system requirements

If I'm to believe the Maxon people, Calc should work with all versions of WorkBench. It is for this reason that I use the Maxon-supplied `rct.library` instead of my old faithful, `gadtools.library`. I haven't had a chance to test this, and if it doesn't, well: for a hobby project that I primarily wrote for myself, it is simply too much work to try to make it compatible with a version of WB I never liked in the first place.

The `rct.library` needs to go into the `Libs:` directory. The font `Topaz8` needs to be in the `Fonts:` directory. Calc needs the `calc.rct` file in the same directory from which the main program was launched; if it isn't, press "Cancel" on the requester that pops up and show Calc where it's GUI file (`calc.rct`) is with the supplied file requester. That's about it.

You can specify where the `calc rct` file is through the tooltypes of the icon, or as an argument in a CLI call. The tooltype may be in one of the following forms:

```
RCT <path>
RCT=<path>
where <path> is the full path and rct file name. EG:
  RCT sys:s/calcrct.RCT
```

As a CLI argument, the form is:

```
calc <path>
EG:
  calc sys:s/calcrct.RCT
```

I don't suggest you put the `rct` file in `s:`; actually, if you leave it in the same directory as the program you don't have to type in the path as an argument if you call Calc from a CLI. However, you always need to set the tooltype. If you don't, you'll always get a requester.

I doubt Calc will eat up much of your memory or CPU time, unless something goes very, very wrong.

The programs included in this release are:

Calc	32140	(The program)
<code>calcrct.rct</code>	17632	(The GUI resource)
<code>Calc.guide</code>	13890	(This file)
icons	4162	
<code>rct.library</code>	88956	
	-----	
Total space	~156780	(156k)

\*\*\*\*\*

## 1.5 Using the program

Using the program (Rules)

Start the program by double-clicking on the icon, or launch it from a

CLI.

Calc uses Reverse Polish Notation. This means you enter the numbers you want to work with and then enter the operation you want to perform. We normally write  $2 + 2$ , and this is the order most calculators take for data entry. On a normal calculator (like the one supplied by Commodore), one would enter '2', then '+', and then '2' again, and then '=' to get the result. To do the same thing on an RPN calculator, one would press the following keys: '2', 'ENTER', '2', and then '+'. The same goes for binary operators, such as 'sin' and 'log'. Enter the number, and then the operator (EG: '2', 'sin').

I'm not going to try to explain why this is better; many people find it more intuitive than standard algebraic entry, and it has become the unofficial standard of computer science majors the world around.

Calc also has a stack. That means that when you enter numbers, they get popped onto a stack, the first 7 elements of which can be seen in the data window of Calc. This makes it especially easy to do complicated operations on more than two numbers, and to keep track of what's going on.

No stack would be complete without a swap function, something common to RPN calculators but rarely found on standard algebraic tools. This means that if you enter 2 and 3 onto the stack and then realize you need to subtract 2 from 3, you can press the swap (\$) button and reverse their order.

I'll give a brief overview of the buttons:

- « - Drop. Removes the lowest element of the stack. If you are entering a number, Drop erases it and leaves the stack unchanged.
  - \$ - Swap. Reverses the order of the two lowest elements on the stack. If a number is being entered when Swap is pressed, it is pushed onto the stack and is then swapped with the next highest number.
  - 0 - 9, . - Buttons for entering numbers.
  - \$\times\$, \$\div\$, -, + - Standard arithmetic operators. If a number is being entered, it is used as 'y' in (x op y). IE, the second number doesn't need to be entered onto the stack to be used in an operation. This applies for all operations in Calc.
  - E - Enter. Enters a number onto the stack.
- All of these buttons may be entered on the numeric pad on the right-hand side of the keyboard. Their positions in the window correspond to the keys in the same position on the keypad; IE, '[' is «, '+' is '+', 'Enter' is 'E', etc.
- ^ - Power. Raises x to the power of y ( $x^y$ ,  $x**y$  for Modula people). y must be greater than 0, although it may be a fraction.
  - rt - Root. Returns the yth root of x. IE,  $x^{(1/y)}$ .
  - lg - Natural log of x.
  - exp - Exponential of x. IE,  $e^x$ .
-

FP - Sets the number of displayed decimal places to x. x must in this case be a number from 1 to 10.

sin, cos, tan - Geometric operations on x.

\ensuremath{\pm} - Reverses the sign of x. If a number is being entered, the number is pushed onto the stack, and then it's sign is changed.

asin, acos, atan - More geometric operations on x.

dec/hex - Toggles the stack display between hexadecimal and decimal. This does not effect the floating point accuracy of the numbers; the floating point part of the numbers are simply ignored during display. IE, 3.5 E 4.5 + entered while in hex mode will still result in '7'.

CALC - Opens the info window. Very boring.

clr - Clears the entire stack, in addition to the number being entered (if any).

String Gadget - Numbers can be entered onto the stack through this gadget. My 'custom' input routine leaves a lot to be desired; this gadget allows you to enter numbers in a more 'user friendly', but slower, way.

In addition are the standard zip, close, and back gadgets.

Calc can only be quit by clicking on the close gadget.

This program uses double precision; that is, 64 bit numbers. Anything over 1e6 is automatically displayed in scientific notation. In addition, you can't display more than 10 decimal places.

It is very unlikely, but possible, that you can crash your computer by pushing too many things onto the stack. I'll put in a safety feature as soon as I find out how to check for low memory.

\*\*\*\*\*

## 1.6 Legal crap

Legal crap

This program may not be sold or included in any commercial packages without my permission. It sucks that us PD programmers have to make this statement. It sucks that there are people out there who would abuse PD authors' good will. Er, um... This program is freely distributable, provided all of the files are kept together (especially this Guide file... I get a warm fuzzy thinking that random people out there are reading my mental spewings). This program may be included in PD collections, such as Fred Fish, SAAR-AG PD, and the TIME series.

Oh, uh... I will not be held responsible if Calc reformats your harddrive or blows up your monitor, or if the very existence of Calc on



your HD causes your friends to avoid you. I am not liable if the Mafia come and snuff you while Calc is running, or if you use Calc to help you break into National Security Network or rob a bank. Since Calc probably only runs under 2.0 or higher, you can't blame me if you get an inferiority complex about having an earlier version of the OS and spend the rest of your life in a metal institution.

But if any of this stuff happens, let me know so I can brag about it in the next version.

Real programmers code for the love of programming.  
Anything else is prostitution

(Don't let that stop you from sending me your recent inheritance in gratitude ==)

"This little thingy is ofcourse FreeWare, but if you REALLY like it, you can send me your sister as payment."

-Arthur Hagen

\*\*\*\*\*

## 1.7 Bug list

Bug list

None that I know of. I've been writing this empty bug list so long... if any of you ever write me about bugs you find, maybe someday I'll have something to put in this section.

\*\*\*\*\*

## 1.8 History & future

History and future

Ok. So I wanted an RPN calculator for the workbench and wasn't satisfied with the one supplied by Commodore. Since I hadn't found one that used RPN in the public domain, I decided that, well, hell! I'm a programmer! I can do this myself.

The original program was done in half a day. Version 2 (the first one that really worked) took about another 2 days of sporadic coding. Most of the problems I had were in dealing with the double floats. For a long time I was getting undesirable imprecisions in the calculations for no apparant reason, and I'm still not sure why they kept cropping up. I don't like the solution I found (it feels like a hammer-and-nails solution), but it works. I have yet to figure out an easy way to display only the significant digits in the floating point results; it's possible, but the answer escapes me. And I still don't like the input routine, but it will be a while before I get around that one. The number of things that I am ignorant about is immeasurable.

---

I have to give credit where credit is due. Although it crashes my computer (A4000/030) routinely upon exit, Maxon MakeAPP was a big help in generating the GUI. It's not quite as flexible as GadToolsBox, a bit slower in the draw, and is more difficult to work with, but for that you get an externally loaded GUI, a bunch of nice built-in requesters, font-sensitivity, and (supposedly) WB1.3 compatability. This program is not font sensitive, but only because I'm not sure what effect that would have on my custom text-drawing routines.

I'm not releasing the code, because it's ugly. There are almost no comments, it's not terribly efficient, and, frankly, I'm embarassed about some of it. Calc was a hack-job which turned out to be longer than I expected.

I have no idea how to program the Amiga. I have no books on how to write programs for the Amiga, and what I know, I learned through looking at other sourcecodes and from Maxon's nice HotHelp program. As a result, things like checking for low memory take me a while to figure out since I have no handy reference on how to do them. As a result, my programs stand on uncertain legs as far a Commodore programming guidelines go.

Everything up till version 2 had bugs that I'm not going to take the time to detail.

⌘ Version 2 - It works! (Almost the way I want it to.)

Future:

- ⌘ Font sensitive (?)
- ⌘ Improved input routine
- ⌘ Improved number output routine (optional automatic detection of significant digits).
- ⌘ More stack operations
- ⌘ MORE BUTTONS!

\*\*\*\*\*

## 1.9 Political Plug

Although I'm living in Germany right now, it would be nice if you people in the States would do something about having Tipper Gore in the White House before I get back. Perhaps it's just paranoia, but there's something unnerving about having America's leading force for censorship as the vice-president's wife.

Hail Eris,

Sean Russell

---