

## Table of Contents

- [Information](#)
- [Background](#)
- [Hello World](#)
- [Manuals and Docs](#)
- [Tutorials](#)
- [Workshops](#)
- [Versions](#)
- [Latest Versions](#)
- [Cross Compiler](#)
- [Original CAR-Images](#)
- [Original ROM-Images](#)
- [Original ATR-Images](#)
- [Source Code](#)
- [Action Language and Editor](#)
- [Functions](#)
- [Mini-Runtime LIBs](#)
- [Examples](#)
- [Tools](#)
- [Missing Tools: Graphics Utilities Library and Shape Editor](#)
- [Articles in Magazines](#)
- [Advertisements](#)
- [Analog](#)
- [Antic](#)
- [ATARI magazin](#)
- [Atari Magazine](#)
- [CK Computer Kontakt](#)

## Information#

### Background#

Action (also Action!) is an Atari-specific programming language written by Clinton Parker and sold by Optimized Systems Software (OSS) in ROM cartridge form starting in August 1983. It is the only language other than [BASIC](#) and [assembler](#) that had real popularity on the platform and saw any significant coverage in the Atari press; type-in programs and various technical articles were found in most magazines. In comparison, languages like [Forth](#) and [Logo](#) saw much less use and almost no press coverage.

Reviewers at the time gushed about the system. They noted that practically every aspect was superior to anything available at the time; compiling was almost instantaneous, the resulting code ran almost as fast as hand-coded assembler, the full-screen editor was universally loved, and the entire system took up only 8k due to clever memory management. The only complaint, also universal, was the poor quality of the original manual set.

Action uses a greatly cut-down version of the ALGOL syntax, and thus bears strong similarities with [Pascal](#) and [C](#), which were also derived from ALGOL. Like those languages, Action is procedural, with programs essentially consisting of a large collection of functions that call each other. It lacked encapsulation or data hiding, but that is not a serious concern in the limited program sizes available on an 8-bit machine. Syntactically it looks very similar to Pascal, with the exception that it uses ALGOL 68 DO/OD style bracketing rather than Pascal's BEGIN/END.

Action included a number of features to allow it to run as fast as possible. Notably, its main data types were BYTE, INT and CARD, 8-bit and 16-bit signed and unsigned values, respectively. These map directly onto the basic 6502-types. The language also included a syntax to directly refer to these objects in memory so they could be mapped into hardware registers. For instance, one could set a variable to `BYTE RTCLOCK=20` which defined the 8-bit value at memory location 20 to be the value of the real-time clock. The user could then read or write to that register using the name `RTCLOCK`.

These design details helped increase performance, but the primary reason Action was much faster than other languages of the era was due to its memory management model. In languages like C and Pascal, procedure calls use a stack of records known as "activation records" that record the values of variables when the procedure was called. This allows a procedure to call itself, as each call can have its own values, and it is this feature that allows recursion. This concept requires the manipulation of a stack, which in the 6502 was a non-trivial prospect given the CPU stack was only 256 bytes.

Action solved this problem by simply not implementing activation records. Instead, the storage space for variables was allocated at compile time (not dissimilar to Atari BASIC's model). This meant Action could not support recursion, but also eliminated the necessity to build and manipulate a complex stack. This dramatically lowers the overhead of procedure calls, and in a language that organizes a program as a series of procedure calls, this represents a significant amount of time.

Action had a number of limitations, none of them very serious. Curiously, Action did not include support for floating-point types, although such support is built into the machine's OS ROM (see [Atari BASIC](#) for details) and available to any programming language. This is a significant limitation in some roles, although perhaps not for its target market. It also lacked most string handling routines, but made up for this somewhat with a series of PRINT commands that made formatted output easy.

Generally, Action programs had performance on-par with reasonable-quality [assembler](#), while being much easier to program. In one review, it ran Byte's Sieve of Eratosthenes 219 times faster than Atari BASIC, while its source was only a few lines longer. In comparison, the assembler version's source ran on for several pages. Such performance, combined with terse code and library functions to access much of the platform's hardware, made it suitable for action games while still having a simple source format suitable for type-in programs. It deserved to be much more popular, and may have been had it been released earlier, or by Atari itself.

Action inspired several similar languages that differ largely in syntax and various features that they do or do not support. Examples include [PL65](#) and [Quick](#).

## Hello World#

```
; Hello world in Action programming language for the Atari 8-Bit computers
PROC Hello()
    PrintE("Hello World!")
RETURN
```

The `;` is a comment marker, which was commonly used as the comment marker in assembler as well. The `PROC` is the start of a PROCedure, which ends (perhaps oddly) with `RETURN`. In Action, the last `PROC` in the program is the one that runs first, in this case "Hello". This is something of a mix between Pascal where the "global code" defines the program entry point, and C, where the function called "Main" is the entry point. The only line of code in this example is `PrintE`, which simply prints a string, while the more common `PrintF` is a formatted print similar to `printf` in C.

Like assembler, it was common for variables to be specified at a particular address that mapped onto one of the Atari's "shadow registers" that were used to communicate between the hardware and user programs. Here is a simple variation on Hello World that demonstrates this concept, as well as a basic loop:

```

; Hello world in a loop
PROC Hello()
BYTE RTCLOCK=20,          ; decimal address of system timer
    CONSOL=$D01F         ; hex address of the key-pressed register
CARD TIME

    RTCLOCK=0 ; reset the clock
    WHILE CONSOL>6 ; did the user press a key?
    DO
        PRINTE("Hello World!")
    OD
    TIME = RTCLOCK
    PRINTF("Ran for %E %U jiffies",TIME)
RETURN

```

Note that the definitions of `RTCLOCK` and `CONSOL` are not setting the values, but stating that they are at those memory locations. The syntax changes when those variables are accessed; the `RTCLOCK=0` *does* set the value of that location. Also notice the syntax of loops, which work similarly to Pascal's `BEGIN/END` but use `DO/OD`.

There is a clever trick in this code. Note that `RTCLOCK` is defined as a `BYTE` but `TIME` is defined as a `CARD`, a 16-bit value. This is because the clock value is stored in three bytes, 18, 19 and 20. By defining `TIME` as a `CARD`, when that value is read it automatically reads two bytes, thereby getting a value from both 20 and 19. This solution ignores the third byte, but since the value is from 0 to 65535 jiffies, about 36 minutes, this can safely be ignored for a program that is likely to run for a few seconds. This solution avoids the need to read two bytes and manipulate them into a 16-bit value, something that is commonly found in BASIC programs.

## Manuals and Docs#

- [Action Manual](#) - The complete latest version of the manual in English. Created and updated by GoodByteXL. There is no better version worldwide! Thank you so much GoodByteXL. We are deep in your debt! :-)))
- [Action Handbuch](#) - Das komplette, vollständige, restaurierte und überarbeitete Action Handbuch in Deutsch! Der totale Hammer, inkl. Editor, Monitor, Language, Compiler, Library, Run Time, Toolkit. Vollständig überarbeitete Version von GoodByteXL. So müssen PDF-Dateien aussehen, es gibt weltweit nichts vergleichbares. AtariWiki empfiehlt die PDF-Datei auf das Wärmste! Wer diese nicht lädt, ist selber schuld. Wir bedanken uns an dieser Stelle sehr, sehr herzlich bei GoodByteXL für seine lange andauernde und intensive Arbeit an diesem Werk, dass er hiermit der Atari-Gemeinschaft zur Verfügung stellt. GoodByteXL mega-Danke für Deine Arbeit, die Gemeinschaft steht tief in Deiner Schuld. :-)))
- [The ACTION! Toolkit](#)
- [The ACTION! Run Time Package](#)
- [ACTION! Reference Card](#)
- [ACTION! error codes](#)
- [Aktion mit ACTION](#) - Report about the ACTION! Programming Language from the German Magazin "Happy Computer"
- [Action! Bugsheet #3](#)
- [Fix for the Bugs in divide in ACTION!](#)
- [Fix for Bug in PrintF](#)
- [Optimized Systems Software, Inc. - SOFTWARE LICENSE AGREEMENT](#) ; thanks to Atarimania

## Tutorials#

- [Step-by-Step Tutorial - How to create a stand-alone ACTION! Program](#)
- [Action! and BBS Express! PRO Tutorial](#)

- [Larry's Action! Tutorial](#)
- [How to setup an ACTION! Development Disk](#)
- [Action! Programming for Atari 8-bit Computers](#) video tutorial series
- [Learning Atari Action!](#) blog

## Workshops#

- [Action! Workshop 1](#) - 24. October 2010 "Unperfekthaus" in Essen
- [Action! Workshop 2](#) - 02. October 2011 "Unperfekthaus" in Essen
- [Action! Workshop 3](#) - 28. October 2012 "Unperfekthaus" in Essen

## Versions#

### Latest Versions#

- [Action! Programming Language](#) - The latest [sources](#) and [binaries](#) for Action
- Includes CAR-Images for OSS cartridges and for standard 16K cartridges
- Includes executable language and editor "ACTION.COM" on ATR-Images for different DOS versions
- Includes executable standalone editor "ACTIONED.COM" on ATR-Images for different DOS versions You can use it under SpartaDOS and DosXL with "D1:ACTIONED MYSRC.ACT" to direct load the "MYSRC.ACT" file into the editor. Further, this editor can even be used for BASIC, PASCAL, FORTH etc.

### Cross Compiler#

- <https://gury.atari8.info/effectus/> - Thank you soooo much Gury! That is totally incredible, we now can use high end editors, eclipse and have the results in a flash! We are deep in your debt! Thank you so much, really. :-)

### Original CAR-Images#

- [ACTION Version 3.6 \(C\) 1983 ACS 034M.car](#)
- [ACTION Version 3.6 \(C\) 1983 ACS M091.car](#)

### Original ROM-Images#

- [ACTION Version 3.6 \(C\) 1983 ACS 034M.rom](#)
- [ACTION Version 3.6 \(C\) 1983 ACS M091.rom](#)

### Original ATR-Images#

- [OSS ACTION Programmers Aid Disk 100.atr](#) - Rebuild from damaged discs and files around the world
- [The ACTION! Toolkit.atr](#)
- [The Action RunTime Disk-Original.atr](#) - Protected image copy of the original disk from a good soul from AtariAge
- [The ACTION! RunTime Disk.atr](#) - Unprotected copy of the original disk from a good soul from AtariAge
- [Original Action! System Runtime Source](#)
- [Alternative Action! Runtime Source](#)
- [ACTION! Runtime von Jeff Reister](#)
- [OSS ACTION 3.6 and REAL files with DOS XL 2.30p Color.atr](#)
- [TURBO-DOS XE with ACTION Disk 1.atr](#)

- [TURBO-DOS\\_XE\\_with\\_ACTION\\_Disk\\_2.atr](#)

## Source Code #

### Action Language and Editor#

at this point AtariWiki must give the highest award possible:

- [Action! Source Code](#)

Thank you so much Mr. Parker, we can't thank you enough for what you have done for us.

Thank you so much Mr. Parker

- [Action! Editor Source Code](#)

Thank you Alfred from AtariAge for extracting the editor source code for generations to come. We are deep in your debt.

Thank you Alfred

### Functions#

- [Misc useful ACTION! Functions](#) - (DIVERS.ACT)
- [Chartest](#) - a group of routines which perform various functions and tests on characters.
- [Fast Screen IO](#)
- [Player Missile](#)
- [String Library PSC](#) - (STRING.ACT)

### Mini-Runtime LIBs#

- [\\_Intro](#) (Eine kleine Einführung zu den Mini-LIBs)
- [Simple PRINT Runtime](#) (Mini-LIB)
- [ZERO and SETBLOCK](#) (RT Part)

### Examples#

- [A pseudo Assembler in Action!](#)
- [ACTION! Logo ACS](#)
- [ATARI Rainbow effect](#)
- [Access SpartaDOS commandline parameters](#)
- [Atari Fuji Logo in ACTION!](#)
- [Atari Picture Mirror Tool](#)
- [Atari ST Mouse Driver for ACTION!](#)
- [BASIC USR Machine Language Call Simulation for ACTION!](#)
- [Backtrack in ACTION!](#)
- [Big Symbol Table for ACTION! ACS](#)
- [Binary File Load in ACTION!](#)
- [Butterfly Demo](#)
- [C Style Strings](#)
- [COM File Segment Dump](#)
- [Catch and Throw Error Handling ACS](#)
- [Catepill](#) unfinished Game with Level editor in ACTION!
- [Compile to Disk ACS](#)

- [DLI in ACTION!](#)
- [DOS Setup](#) - A small tool to copy some files from disk to ramdisk. can be configured by a textfile.
- [Data Entry Routines](#)
- [Date Routines](#) - Library of routines supporting the input, storage and manipulation of dates.
- [Delete EOL Char in Textfiles](#)
- [Displaylist in ACTION!](#)
- [END Procedure](#) - Call procedure to leave an Action Program
- [ERROR](#) (Converts SpartaDOS, BeWeDOS and RealDOS error # to readable text)
- [Fast Graphics 15 Routines](#)
- [Fast Graphics 8 Routines](#)
- [File Compare](#)
- [File IO Routines](#)
- [File Select Box](#)
- [File Select Shell](#)
- [Game AMAZING in ACTION!](#)
- [Grep for Sparta DOS](#)
- [HexDump](#) - Dump - Print the contents of binary files in hexadecimal and ATASCII
- [Jump to DOS DUP](#)
- [Kermit in ACTION!](#)
- [Load APL Display-List Files](#)
- [Load Font Files in ACTION!](#)
- [Load Koala Pictures](#)
- [MiniDOS](#)
- [Multi Player Animation](#)
- [OS Vectors](#)
- [PERCOM Block Manipulation](#)
- [PERCOM Service](#) -- Disk Format Configuration
- [Printing Routine for Epson Printer](#)
- [Query Console Keys](#)
- [SIO CIO Routine](#)
- [SourceCodeDisk1](#) ; SpartaDOS X disk image with Action source code
- [Starburst](#)
- [Symbol table lister ACS](#)
- [Timer Programming](#)
- [Trackball](#)
- [Using the RAM Under the OS ROM on XL and XE Computers](#)
- [VT52 Terminal Emulator](#)
- [VTEulator](#)
- [Windowing Routines](#)
- [XFD Disk Transfer tool](#) XFormer Filetransfere
- [XModem Filetransfer](#)

## Tools#

- [Action Source Code Formatter](#)
- [Infoline](#) for ACTION! and BASIC
- [ACTION OBJECT CODE RELOCATION PROGRAM](#) ; Thank you so much Alfred from AtariAge, we all really appreciate your help here again.
- [Relocator](#) for Action; relocates Action code to run independent from the code location
- [acsterm.txt](#) ; ACSTERM is a terminal emulator for the Atari 800, 800XL, 1200XL and 130XE
- [How to find the revision number of ACTION](#)

## Missing Tools: Graphics Utilities Library and Shape Editor#

As of 2020, just these two programs are still missing. They were published via OSS-BBS only! The number was: +1 (408) 438 - 6775. Any hint, any help is welcome at any time. We would really appreciate your help in that case.

Graphics Utilities Library for ACTION! screen 1

Graphics Utilities Library for ACTION! screen 2

Shape Editor and Animator for ACTION!

OSS-BBS with the number: +1 (408) 438 - 6775 where the two missing programs were published only!

## Articles in Magazines#

### Advertisements#

First Action ad in Compute July, 1983 ; please take into account: 128-column screen and for Apple II & Commodore 64. Thanks to GoodByteXL!

### Analog#

| Title                                       | Issue                  | Language | Comment                                |
|---|------------------------|----------|--|
| <a href="#">Review Action!</a>              | #16 (02/ 84)           | en       | Review                                 |
| <a href="#">An Introduction to ACTION!</a>  | #17 + #18 (03+ 04/ 84) | en       | Tutorial                               |
| <a href="#">Stars in 3D</a>                 | #20 (07/ 84)           | en       | Demo                                   |
| <a href="#">Bounce in ACTION!</a>           | #20 (07/ 84)           | en       | Game                                   |
| <a href="#">Pulse in ACTION!</a>            | #26 (01/ 85)           | en       | Demo                                   |
| <a href="#">More Fun with Bounce!</a>       | #27 (02/ 85)           | en       | Game                                   |
| <a href="#">Demon Birds</a>                 | #28 (03/ 85)           | en       | Game                                   |
| <a href="#">Roto</a>                        | #31 (06/ 85)           | en       | Game                                   |
| <a href="#">Color the shapes</a>            | #32 (07/ 85)           | en       | game                                   |
| <a href="#">Getting in on the Action! 1</a> | #32 (07/ 85)           | en       | Tutorial                               |
| <a href="#">Getting in on the Action! 2</a> | #35 (10/ 85)           | en       | Tutorial                               |
| <a href="#">Sneak attack</a>                | #36 (11/ 85)           | en       | Game                                   |
| <a href="#">Air hockey</a>                  | #38 (01/ 86)           | en       | Game                                   |
| <a href="#">D-Check</a>                     | #44 (07/ 86)           | en       | Tool                                   |
| <a href="#">Trails</a>                      | #50 (01/ 87)           | en       | Tool for using the KoalaPad in ACTION! |
| <a href="#">ACTION! Zero Free</a>           | #54 (05/ 87)           | en       | Tool                                   |

## Antic#

| Title                                   | Issue               | Language | Comment  |
|---|---------------------|----------|--|
| <a href="#">Interrupts in ACTION!</a>   | Vol. 3 #3 (07/ 84)  | en       |  |
| <a href="#">Demo Pretty</a>             | Vol. 3 #7 (11/ 84)  | en       | Demo from Antic I/O-Board                            |
| <a href="#">SPLASH in ACTION!</a>       | Vol. 3 #12 (04/ 85) | en       | Demo   |
| <a href="#">Game AMAZING in ACTION!</a> | Vol. 4 #1 (05/ 85)  | en       | Game   |
| <a href="#">View 3D</a>                 | Vol. 4 #2 (06/ 85)  | en       | Tool   |
| <a href="#">Dark Star</a>               | Vol. 4 #3 (07/ 85)  | en       | Game: Zapping Aliens With Radioactive Waste          |
| <a href="#">Display Master</a>          | Vol. 4 #4 (08/ 85)  | en       |  |
| <a href="#">Eight Queens</a>            | Vol. 4 #5 (09/ 85)  | en       | 92 chess solutions in 40 seconds                     |
| <a href="#">Video Stretch</a>           | Vol. 5 #6 (10/ 86)  | en       | Tool   |
| <a href="#">Killer Chess</a>            | Vol. 6 #10 (02/ 88) | en       | Game   |
| <a href="#">Reardoor</a>                | Vol. 6 #10 (02/ 88) | en       | Game   |
| <a href="#">Frog</a>                    | Vol. 6 #10 (02/ 88) | en       | Game   |
| <a href="#">ACTION! Toolbox</a>         | Vol. 7 #6 (10/ 88)  | en       | Lightning-fast command finder (Wordfind and Matchup) |

## ATARI magazin#

| Title  | Issue        | Language | Comment                         |
|--|--------------|----------|---------------------------------|
| <a href="#">Schnelle Vektoren in ACTION!</a> | #1 (1-2/ 87) | de       | Tutorial: Action!-Center Teil 1 |
| <a href="#">Schnelle Umwege in ACTION!</a>   | #2 (3-4/ 87) | de       | Tutorial: Action!-Center Teil 2 |
| <a href="#">Interne Variablen</a>            | #3 (5-6/ 87) | de       | Tutorial: Action!-Center Teil 3 |
| <a href="#">Was ist dran an Action!?</a>     | #4 (7-8/ 87) | de       | Tutorial: Action!-Center Teil 4 |

## Atari Magazine#

| Title                        | Issue | Language | Comment                         |
|------------------------------|-------|----------|---------------------------------|
| <a href="#">ACTION! Deel</a> |       | nl       | A collection of Action Articles |

## CK Computer Kontakt#

| Title                                  | Issue   | Language | Comment  |
|--|---------|----------|----------|
| <a href="#">Musik in ACTION</a>        | #10/85  | ge       | Tutorial |
| <a href="#">ACTION! noch schneller</a> | #6-7/86 | ge       | Tutorial |