

C/65 Manual - Text Version

work in progress -- Please help by copying the OCR text from the original PDF, fix the OCR errors and append here

- [new PDF from the pure text version](#)

??
C/65 MANUAL

Optimized Systems Software
??

Table of Contents
????????????????????????????????

- 1 Abstract
- 2 PREFACE
- 3 TRADEMARKS
- 4 Introduction
 - .. 4.1 What's in it
 - .. 4.2 What's not in it
- 5 Using C/65
 - .. 5.1 Special Notes

1 Abstract
????????????

Reference Manual for C/65 a small C Language compiler for use with Atari 400, Atari 800, and Apple II Computers

The programs, disks, and manuals comprising C/65 are Copyright (c) 1982 by Optimized Systems Software, Inc and LightSpeed Software

This manual is Copyright (c) 1982 by Optimized Systems Software, Inc., of 10379 Lansdale Avenue, Cupertino, CA

All rights reserved. Reproduction or translation of any part of this work beyond that permitted by sections 107 and 108 of the United States Copyright Act without the permission of the copyright owner is unlawful

2 PREFACE
????????????

We realise that C/65 is not the most sophisticated, most complete, language on the market today, but we believe that the inherent power and flexibillity that it exhibits within its compact size are a good match for the size and features of the machines it is intended for.

C/65 was authored by Sam Dillon and John Lowry, under the company name of LightSpeed Software, based on the Small C Compiler published in

Dr. Dobb' Journal. C/65 is a hand-coded translation from C code to 6502 assembly language and is, as result, a fast and easy to use compiler.

3 TRADEMARKS

????????????

The following trademarked names are used in various places within this manual, and credit is hereby given

OS/A+, BASIC A+, MAC/65, and C/65 are trademarks of Optimized Systems Software, Inc.

Apple, Apple II, and Apple Computer(s) are trademarks of Apple Computer, Inc., Cupertino, CA

Atari, Atari 400, Atari 800, Atari Home Computers, and Atari 850 Interface Nodule are trademarks of Atari, Inc., Sunnyvale, CA.

4 Introduction

????????????

C/65 is a subset of the C programming language as defined by Kernighan and Richie in the book, "The C Programming Language", published by Prentice-Hall.

With a few clearly noted exceptions, programs written in C/65 are compilable without modification under Standard C.

The C/65 package comes with a runtime library, which includes standard-looking character input and output functions, all of which are described later in this document. The output of the C/65 compiler is MAC/65 assembly language, which must be run through the MAC/65 assembler to produce a runnable object module. It is possible for this to be done automatically. Since the output is assembly language, it is easy to write your own assembly language routines that are compatible with the code generated by this compiler.

4.1 What's in it

????????????

Very briefly, C/65 supports

- ? the basic data types CHAR and INT
- ? pointers to the basic types (*)
- ? one dimensional arrays of the basic types ()
- ? the basic arithmetic, logical, and bit operations familiar to C programmer
- ? simple source level character substitution (\$DEFINE)
- ? file inclusion of C source code
- ? file inclusion of assembly language sourcecode (not compatible with standard C)
- ? functions with parameters and Local variables
- ? an if statement, with an optional else clause
- ? a while statement
- ? break and continue statements

- ? a return statement, with an optional return value
- ? compound statements grouped by braces (although '\$(' and '\$)' must be used instead of '(' and ')')
- ? separate compilation
- ? linked external declarations

4.2 What's not in it

????????????????????

for experienced C programmers, use of the following standard C will get you in trouble:

- ? long ints
- ? unsigned ints (but note that pointers to char will do most of what you want here)
- ? floats and doubles
- ? structures, unions and bit fields
- ? multi-dimensional arrays
- ? parameters to #DEFINE macros
- ? += and his brothers -=, *=, etc
- ? for statement
- ? do-while statement
- ? switch statement
- ? &&, !!, unary !

There are other restrictions not listed here, but these seem to be the major ones. Despite this, C/65 is complete enough that one could write C/65 in itself, space considerations aside. For various reasons, C/65 is written in assembly language, which makes it extremely fast and quite small.

5 Using C/65

????????????????

Are you a C hacker already? Did the introduction (Sections 1.1 and 1.2) tell enough? Anxious to get started? Here you go!

After using your favorite text editor to create a C source file, enter the command to OS/A+:

```
?????
? C65 filename1 filename2 [-T
?????
```

where

- ? filename1 is the name of the source file
- ? filename2 is the name of the output file
- ? -T is an optional flag which tells C/65 to include the C source text as comments in the assembler output file.

5.1 Special Notes

????????????????

E: is a valid filename for either source or output files (or both). I.e., compiler output can go directly to the screen, or you can even type in your C program on the fly.

Assemble the output file using MAC/65. Consult your MAC/65 reference manual for details of this operation. A complete, start to finish, compilation and assembly is shown in the next section.

Note: MAC/65 may be used to edit C source files if TEXTMODE is selected (via the TEXT command).