

# MA65 Macro Assembler Version 1.6 For SpartaDOS

©1998-2008 by Jonathan Halliday

*Credits: MA65 is based on and includes code from the A65 Assembler by Charles Marslett, and was originally assembled using an enhanced version of A65. Thanks to Charles for providing a basic, open-source assembler without which MA65 probably wouldn't exist.*

# Contents

1	INTRODUCTION.....	1-3
1.1	FEATURES OF THE ASSEMBLER .....	1-3
1.1.1	MACROS .....	1-3
1.1.2	CONDITIONAL ASSEMBLY .....	1-3
1.1.3	MODULAR CODE .....	1-3
2	COMMAND LINE SYNTAX.....	2-4
3	COMMAND LINE EXAMPLES.....	3-5
4	THE ASSEMBLY PROCESS.....	4-6
5	SOURCE LISTING FORMAT .....	5-7
6	SYMBOL MAP FORMAT.....	6-8
7	LANGUAGE STRUCTURE .....	7-9
7.1	SYMBOLS AND MACRO NAMES .....	7-9
7.2	NUMBERS .....	7-9
7.3	STRINGS.....	7-9
7.4	OPERATORS.....	7-9
7.4.1	PRECEDENCE .....	7-10
8	PSEUDO-OPERATIONS .....	8-11
9	ERROR CODES.....	9-16

## **1 INTRODUCTION**

The MA65 Macro Assembler is specially designed for use with SpartaDOS 3.x and the SpartaDOS X Cartridge. A disk to disk assembler, it is modelled on the Atari Macro Assembler and MAC/65, and includes many features found in both these packages. MA65 is a development of the A65 Assembler by Charles Marslett, which provided the basic framework of the assembler, onto which was added the Macro pre-processor, conditional assembly routines, local variables and procedures, and SpartaDOS integration.

### **1.1 FEATURES OF THE ASSEMBLER**

#### **1.1.1 MACROS**

The MA65 Assembler allows macros to be defined, allowing large sections of code to duplicated with a single instruction. Macros can call or define other macros, and can call themselves recursively, providing recursion is eventually terminated by means of conditional assembly.

#### **1.1.2 CONDITIONAL ASSEMBLY**

Conditional Assembly, using IF ... ELSE ... ENDIF constructs, allows code to be generated selectively, according to special computed conditions.

#### **1.1.3 MODULAR CODE**

PROC ... EPROC procedure constructs allow sections of code to define and use symbols and labels invisible to the rest of the program.

## 2 COMMAND LINE SYNTAX

(Square brackets surrounding an argument mean the argument is optional. Under SpartaDOS X, kernel device names may be used on the command line, such as "CON:" for the screen and "PRN:" for the printer. In both SpartaDOS X and earlier versions, however, all filenames in the actual source code must be preceded by "D:" or "Dn:" in the usual manner.)

MA65            <source> [.ext] [dest[.ext]] [/I lib\_path] [/L [list\_path]] [/M  
[list\_path]] [/P nnn] [/C nnn] [/F] [/?]

<source>        Mandatory source filespec (default extension is "A65").

[dest]           Optional object code filespec (default extension is "OBJ"). If no object filespec is given, the object file will take the source filename with the extension "OBJ". If a drive letter/number is specified, the object file will be written to that drive, otherwise it will be written to the drive on which the source file is located.

[/I]            Path for INCLUDE files. This drive/path will precede the names of included files in the source code, unless their names already have a path of their own. Follow the "/" option with a space or comma, then the path, optionally terminated with ">". The default include path is the path of the main source file.

[/L]            Produce source listing. Follow with optional path/filespec. "/" on its own will send a listing to the screen. If a drive number is specified, the listing will be sent to a file with the same name as the source file, plus the extension "PRN". A different filename may be specified, however. "/L P:" ("/L PRN:" with SpartaDOS X) will send the listing to the printer.

[/M]            Produce symbol map listing. Include this option as well as the "/" option to send both source listing and symbol map to the listing device.

[/P]            Source listing page length. Follow with a space, then the number of lines in decimal (default=58).

[/C]            Specify columns in source listing. Follow with a space then the number of columns in decimal (default = 80 printer and disk, 38 for the screen).

[/F]            Print entire line when printing to the screen. Overrides default truncation to 38 characters.

[/?]                      Print usage help to screen then return to DOS prompt.  
Improper arguments will cause the assembler to print the command line options to the screen and abort. Assuming the arguments are acceptable, the assembler will then display the title message and assembly will begin. The names of the source files are printed on the screen as they are assembled. Fatal errors will cause the assembler to abort operation. Assembly errors are flagged to and printed to the screen during pass two.

### **3    COMMAND LINE EXAMPLES**

#### **MA65 DEMO**

Assembles the program DEMO.MAC on the current drive, placing the resultant object code in the file DEMO.OBJ.

#### **MA65 TEST D8:/L P:/P 66**

Assembles the program TEST.MAC on the current drive, producing the object code file TEST.OBJ on drive 8, and sending a source listing to the printer with a page length of 66 lines.

#### **MA65 D2:PROJECT.SRC /I D1:>LIB> /L /M D8:LIST**

Assembles the program PROJECT.SRC, looking for included files in the LIB folder on drive 1, and sends a source listing and symbol map to the file LIST.PRN on drive 8.

## **4 THE ASSEMBLY PROCESS**

MA65 is a two pass disk to disk assembler which produces a compound object file with the default extender of "OBJ". The UNSEG utility on the distribution disk will take compound binary files and desegment them so that they take up less space on disk and load more quickly. You will probably want to run all object files that MA65 produces through the UNSEG utility and give the resulting files "COM" extenders.

Pass 1 builds the symbol table and reads macro definitions from the source file. If, during the first pass, a macro is improperly defined, or a disk error such as a missing include file occurs, the assembler will abort and provide an error message.

Pass 2 expands macros (which must be defined before being called in pass 1, i.e. at the start of the program), prints the listing file, outputs any lines with errors to the screen, and writes the object code file.

## 5 SOURCE LISTING FORMAT

Column	Description
1	Error flag or blank.
2	Blank.
3-6	Address (program counter)
6	"-" sign means not assembled owing to IF ... ELSE. Only listed if OPT CLIST is in effect.
7	Blank.
8	"+" sign means macro-generated line. Only listed if OPT MLIST is in effect.
9-18	Object code in HEX. Multiple lines will be listed with up to 6 bytes in each.
11-14	"=nnnn". Value of expression in HEX.
19-20	Blank.
21-80	Source statement.

## 6 SYMBOL MAP FORMAT

If “/M” is specified on the command line, a map of symbols is printed at the end of the listing file. Its symbol entry has the following format:

<nnnn> <f> <symbol>, where:

<nnnn>	Value of symbol in hexadecimal, or “Mac” if the symbol is a macro name.
<f>	Symbol flag: U = Undefined D = Doubly defined N = Not referenced
<symbol>	Symbol name (maximum 16 characters).

Local symbols are not listed in the symbol map.



## 7 LANGUAGE STRUCTURE

### 7.1 SYMBOLS AND MACRO NAMES

Symbols may include the following characters:

A-Z	Uppercase letters.
a-z	Lowercase letters (converted to uppercase).
:	Denotes local symbol if first character, otherwise ignored.
0-9	Digits (may not be first character).
_	Underscore; may appear anywhere and is retained.
.	Period; may appear anywhere and is retained.

Only the first 16 characters of a symbol are stored in the symbol table: any further characters are discarded.

### 7.2 NUMBERS

Numbers may have the following prefixes:

%	Binary
@	Octal
\$	Hexadecimal

No prefix implies decimal.

*	On its own as an operand denotes the value of the program counter.
---	--

### 7.3 STRINGS

Strings may contain any characters, including inverse video characters, except EOL (ATASCII 155, \$9B). If the string is surrounded by single quotes it may contain literal double quotes, and vice-versa.

### 7.4 OPERATORS

Expressions may include the following operators:

+	Sum or positive sign.
-	Difference or negative sign.
*	Multiply.
/	Divide.

NOT	Complement (negates operand).
AND	Logical AND.
&	Logical AND.
OR	Inclusive OR.
XOR	Exclusive OR.
=	Equality.
<>	Inequality.
<	Less than.
>	Greater than.
<=	Less than or equal to.
>=	Greater than or equal to.
SHL	Shift left n bits.
<<	Shift left n bits.
SHR	Shift right n bits.
>>	Shift right n bits.
HIGH	Unary, n/256.
HI	Unary, n/256.
>	Unary, n/256.
LOW	Unary, n MOD 256.
LO	Unary, n MOD 256.
<	Unary, n MOD 256.
MOD	Modulus.
DEF	Test for symbol definition.
[	Left bracket.
]	Right bracket.

### 7.4.1 PRECEDENCE

<b>Highest</b>	Brackets HIGH LOW DEF * / MOD SHL SHR + - Unary + - Binary = <> < <= > >= NOT & AND
<b>Lowest</b>	OR XOR

## 8 PSEUDO-OPERATIONS

Abbreviations:

iglab	Ignored label field.
<exp>	Required expression.
[exp]	Optional expression.

DB, BYTE      Define Byte

LABEL: DB <exp> ... , <exp>

where: <exp> = Any legal expression, value or string.

DC, CBYTE     Define Character

LABEL: DC <exp> ... , <exp>

where: <exp> = Any legal expression, value or string.

DS             Define Space

LABEL: DS <expr16>

where: <expr16> = Any legal expression, value or string.

WORD, DBYTE  
DW             Define Word

LABEL: DW <expr16> ... , <expr16>

where: <expr16> = Any expression, value, or 1 to 2 character string.

EJECT          Eject listing page.

iglab    EJECT

ELSE           Conditional assembly clause

iglab    ELSE

END            End Program

LABEL: END [init][,run]

where: [init] is the optional initialization address of the program,  
and [run] is the optional run address.

ENDIF           End conditional assembly

iglab    ENDIF

ENDM           End Macro definition

iglab    ENDM

EPROC           End local symbol range

iglab    EPROC

EQU or =        Equate value to symbol

LABEL: EQU <expr16>  
LABEL: = <expr16>

where: <expr16> is any expression, value, or 1 to 2 character  
string.

ERROR           Force assembly error

iglab    ERROR [message]

where: [message] is an optional quoted string which will be  
printed to the screen unless the source listing is already directed  
to the screen.

FLOAT           Define Floating Point Number

LABEL: FLOAT <fpnum>

where: <fpnum> = floating point number

IF               Specify conditional assembly

iglab    IF <exp>  
          <assembly code>  
iglab    ENDIF

iglab    IF <exp>  
          <assembly code>  
iglab    ELSE  
          <assembly code>

iglab    ENDIF

where: <exp> = expression. Non-zero = true, zero = false.

IF clauses are nestable up to a depth of 255.

INCLUDE    Include source file

LABEL: INCLUDE [dev:][path]<filespec>[.ext]

where: <filespec> is the name of the file to be included. If the device is omitted or is "D:" and no path is specified, the path (if any) defined on the command line will be used. The default file extension is "A65".

LOC        Set location counter

LABEL: SET <expr16>

where: <expr16> = any value or expression.

Use LOC to set the location counter when assembling code which will be relocated at runtime. For example, if a routine begins with:

```
ORG $4000
LOC $C000
```

Code will be assembled as with the origin at \$C000, but will load at \$4000. This is useful when assembling routines which will be relocated under the OS ROM at runtime.

MACRO      Define Macro

```
MACNAME:  MACRO <param1> ... , <param9>
           <body>
           ENDM
```

where: <param1> to <param9> is a list of dummy arguments for documentation purposes only, and <body> = any desired assembly text, which may include parameters:

%0	Decimal count of parameters in macro call
%1 ... %9	Supplied parameters
%%	Percent symbol

Macros may define other macros and call other macros or themselves recursively to a depth of 127 or until parameter memory is exhausted. Macros must be defined before use.

**OPT**                    Set Assembly Options

iglab    OPT [NO] <opt> ... , [NO] <opt>

where: <opt> is one of the following:

CLIST   List skipped conditional code (default = ON)

MLIST   List macro definitions (default = OFF)

PLIST   List source listing directives

VLIST   List verbose (default = OFF)

EJECT   Eject last page of source listing (default = OFF)

TITLE   Print listing title (default = ON)

Options can be deselected by preceding them with 'NO' and a space.

**ORG, \*=**                Set origin counter

LABEL: ORG <expr16>

where: <expr16> = any value or expression.

**PROC**                    Define local symbol range

LABEL: PROC  
      <body>  
      EPROC

Symbol names within <body> which begin with ":" are local to the PROC range and do not exist outside of it. Local symbols do not appear in the symbol table listing. PROCs are nestable to a level of 127.

**SBYTE**                  Define internal screen codes

LABEL: SBYTE <expr>

where: <expr> = any legal expression, value, or string.

SBYTE works like DB, but converts values to internal Atari screen codes.

**SET**                    Define Variable Symbol

LABEL: SET <exp>

Variable symbols may be redefined throughout a program.

**TITLE**            Define Output Listing Title

          iglab    **TITLE** <string>

          where: <string> = any string of up to 40 characters.

## **9 ERROR CODES**

Error codes appear in column one of the source listing. Lines containing errors are always written to the screen during pass two, regardless of whether source listings are enabled.

A	Address error.
D	Duplicated label error.
E	Expression error.
F	Bad nesting of IF ... ELSE ... ENDIF, PROC without matching EPROC.
I	Instruction not recognized.
M	Macro definition error.
N	No free IOCB for include file.
P	Programmer forced error.
S	String length error.
U	Reference to undefined symbol.
V	Branch out of range.