

MEMACCLIB

Conversion program

COLLABORATORS

	<i>TITLE :</i> MEMACCLIB		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Conversion program	October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	MEMACCLIB	1
1.1	Overview of MEMACCLIB	1
1.2	MEMACCLIB	1
1.3	MEMACCLIB	2
1.4	MEMACCLIB	2
1.5	MEMACCLIB	2
1.6	MEMACCLIB	3

Chapter 1

MEMACCLIB

1.1 Overview of MEMACCLIB

Overview

An Acid Software Library

Converted to AmigaGuide by

Red When Excited Ltd

Used with the permission of Acid Software

Edited, fixed and cleaned by Toby Zuijdveld 27/02/1999.
mailto:hotcakes@abacus.net.au

1.2 MEMACCLIB

Statement: Poke

Modes :

Syntax : Poke [.Type] Address,Data

The Poke command will place the specified Data into the absolute memory location specified by Address. The size of the Poke may be specified by the optional Type parameter. For example, to poke a byte into memory, you would use Poke.b; to poke a word into memory you would use Poke.w; and to poke a long word into memory you would use Poke.l

In addition, strings may be poked into memory by use of Poke\$. This will cause the ascii code of all characters in the string specified by Data to be poked, byte by byte, into consecutive memory locations. An extra 0 is also poked past the end of the string.

1.3 MEMACCLIB

Statement: Peek

Modes :

Syntax : Peek [Type] (Address)

The Peek function returns the contents of the absolute memory location specified by Address. The optional Type parameter allows peeking of different sizes. For example, to peek a byte, you would use Peek.b; to peek a word, you would use Peek.w; and to peek a long, you would use Peek.l

It is also possible to peek a string using Peek\$. This will return a string of characters read from consecutive memory locations until a byte of 0 is found.

1.4 MEMACCLIB

Function: Peeks\$

Modes :

Syntax : =Peeks\$ (Address,Length)

Peeks\$ will return a string of characters corresponding to bytes peeked from consecutive memory locations starting at Address, and Length characters in length.

1.5 MEMACCLIB

Statement: Call

Modes :

Syntax : Call Address

Call will cause program flow to be transferred to the absolute memory location specified by Address.

PLEASE NOTE! Call is for advanced programmers only, as incorrect use of Call can lead to severe problems - GURUS etc!

A 68000 JSR instruction is used to transfer program flow, so an RTS may be used to transfer back to the Blitz 2 program.

Please refer to the 'Assembler' section of the reference guide for the rules machine code programs must follow to operate correctly within the Blitz 2 environment.

1.6 MEMACCLIB

MEMACCLIB

Overview

Command Index

Call

Peek

Peeks\$

Poke
