RSC_Viewer

A Tool To Dump, Disassemble and Patch Macintosh Resources.

Version 4.6 (01/26/1991)

Written by François Menneteau

Written in *THINK's LightspeedC*
Portions © 1986, THINK Technologies, Incorporated.

TABLE OF CONTENTS

INTRODUCTION

This application is a tool to dump, disassemble and patch any kind of resources.

It is not a Resource Editor like *ResEdit* ™ because resource manipulations are very limitated (e.g. you cannot create, add, delete or append resources). In addtion, you cannot change any file attributes i.e. if a file is locked and you want to patch it, you must unlock it before laun-ching *RSC_Viewer*.

However resource attributes modifications are allowed.

Indeed *RSC_Viewer* is more oriented toward understanding how applications work. Thus many useful information are available into different **windows** (like the list of traps currently recognized, the low memory global variables addresses, etc), and they can be consulted at any time.

RSC_Viewer also offers the possibility to save or print dumped or disassembled resources in order to examine them (or use them) later.

THE WINDOWS

In the *RSC_Viewer* application there are nine windows. We give here their "symbolic" names and a brief description of their contents :

- **The RSRC FILE WINDOW.** This is the main window. It contains the current resource in dump or disassemblage mode.

- **The SYMBOLS WINDOW.** It contains the list of all Macsbug symbols (see MacsBug manual for more information about symbol formats).

Note : during the loading phase, the application informes you if some sym-bols are available .

- **The GLOBAL VARIABLES WINDOW.** It contains a list of most of the low memory global variables.

  It is possible to update this list, see the **RSCV_Compiler.Doc** file.

- **The TOOLBOX TRAPS WINDOW.** It contains a list of the Toolbox traps currently recognized. This list is taken from Inside Macintosh Volume I through V and from various Technical Notes.

For each trap, its hexadecimal value and its name are displayed. In addition, if the trap is a dispatch one, the symbol "/d" is appended to its na-me, and if the trap is an element of a dispatch or package trap, the symbol "/value" (where value is the value that is pushed into the stack) is append to its name (exemple : A260: PBGetFCBInfo /8).

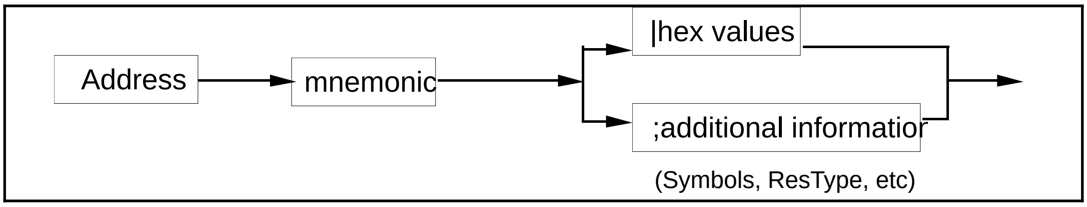
- **The OS TRAPS WINDOW.** It contains a list of the OS traps currently recognized. The syntax used for the Toolbox traps also applies here.

- **The three MC680xx WINDOWS.** They contain information about MC68000 and MC68020 format, MC68020 operations and MC68000 ins-tructions. If you want to update them (for example with some 68020 and 68030 instructions), you must modify the corresponding PICT resources.

- **The AVAILABLE MEMORY WINDOW.** It contains the value of the current free memory under finder, and the remaining room of your partition under multifinder.

THE DISASSEMBLER

A great effort has been made to produce a MC680xx code as clear as possible. The format of each line of code is as follows :



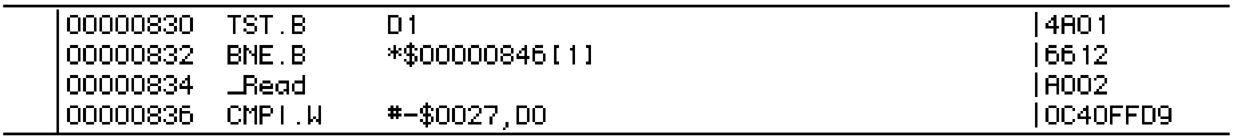
The syntax for a mnemonic symbol is as close as possible to the one used in MacsBug. Howe-ver, there are little changes (for example the address representation), but it is generally for a clarity purpose.

Note : FPU instructions are not disassembled yet.

Any kind of resources can be disassembled (see **Filter…** and **File Rsrc List…** in the File menu for more information), and the following functionalities are provided :

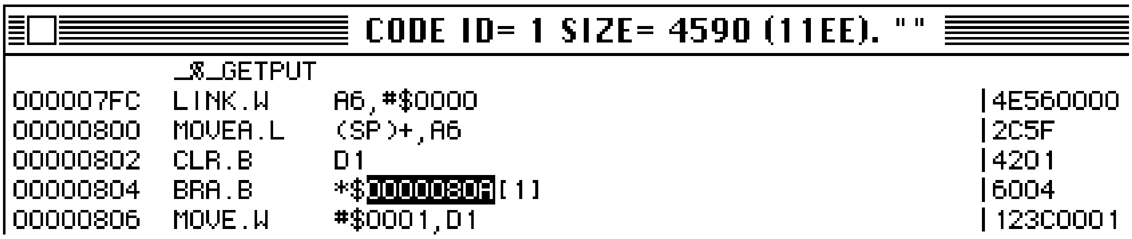
- Conversion of a d(PC) instruction into a more suitable address i.e. **Offset[resource number]** (Offset is a long hexadecimal value and resour-ce number is a decimal value).

- Conversion of a trap value into its corresponding name. Trap dispatches and Packages are also converted, but sometimes, it is impossible to give their real name because the program does not succeed in finding the value put into the stack (due to static disassemblage).



```
00000830  TST.B    D1                                      |4A01
00000832  BNE.B    *$00000846[1]                           |6612
00000834  _Read                                            |A002
00000836  CMPI.W   #-$0027,D0                              |0C40FFD9
```

- Disassemblage at a specific address (remember : address = Offset [id]), by pressing the Command key and selecting it by double-clicking.



CODE ID= 1 SIZE= 4590 (11EE). ""

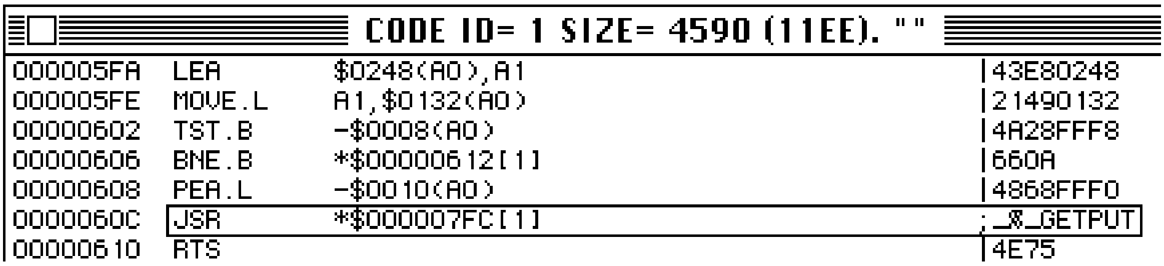
```
            _%_GETPUT
000007FC    LINK.W     A6,#$0000                               |4E560000
00000800    MOVEA.L    (SP)+,A6                                |2C5F
00000802    CLR.B      D1                                      |4201
00000804    BRA.B      *$0000080A[1]                           |6004
00000806    MOVE.W     #$0001,D1                               |123C0001
```

Some additionnal features are also available when disassembling resources of type CODE. These are :

- Resolution of external references, by scanning the jump table (CODE ID 0) and converting a d(A5) instruction into an Offset [resource number] address.

- Translation of the MAC-OS low memory addresses (global variables) into their corresponding names (if possible). The names are those found in *Inside Macintosh* Volume I through V and in various other sources (Compilers, Technical Notes, etc).

- When Macsbug symbols are available, translation of the Offset [rsrc number] address into the corresponding symbolic name.



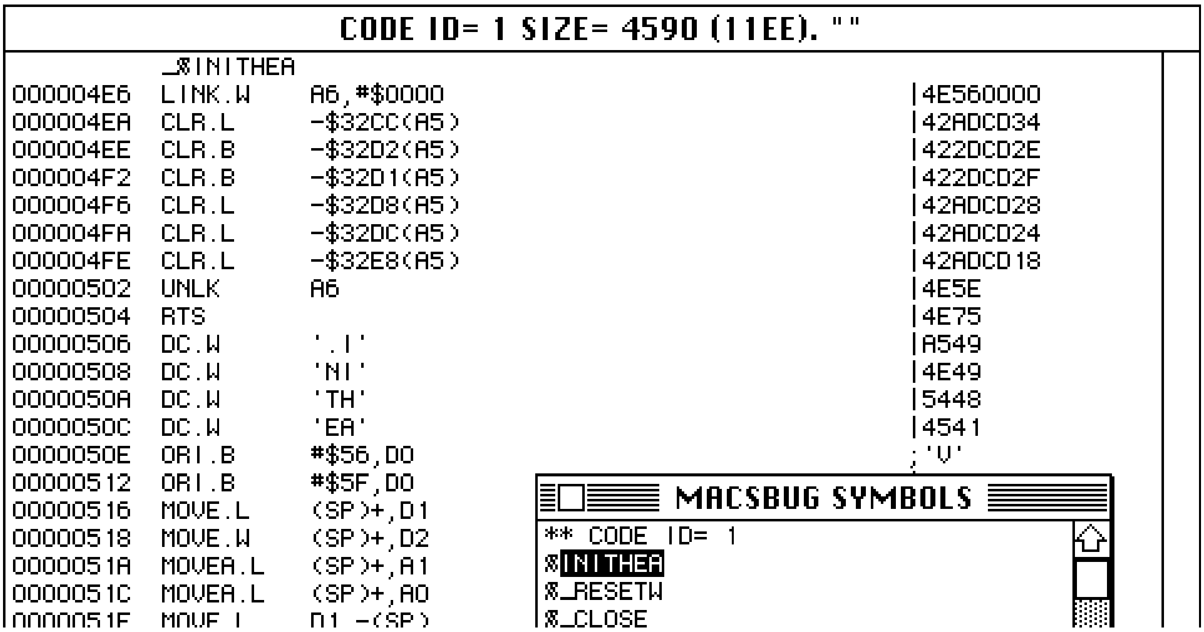
```
CODE ID= 1 SIZE= 4590 (11EE). ""
000005FA  LEA      $0248(A0),A1                              |43E80248
000005FE  MOVE.L   A1,$0132(A0)                              |21490132
00000602  TST.B    -$0008(A0)                                |4A28FFF8
00000606  BNE.B    *$00000612[1]                             |660A
00000608  PEA.L    -$0010(A0)                                |4868FFF0
0000060C  JSR      *$000007FC[1]                             ;_%_GETPUT
00000610  RTS                                                |4E75
```

- Display of the application entry point, if possible (see **Resource Table**).

- Jump of the four first bytes of any CODE resources (except for the “Jump Table” where 16 bytes are jumped), because these bytes are used by the system and so they have no interest from the disassembler point of view.

- Attempt not to disassemble the symbolic information stored in the code, in order to produce a better listing (it sometimes fails).

- Disassemblage at a specific symbol address, by pressing the Command key and double-clicking on the correspondind symbol in the SYMBOLS WINDOW.



CODE ID= 1 SIZE= 4590 (11EE). ""

```
         _%INITHEA
000004E6 LINK.W   A6,#$0000                         |4E560000
000004EA CLR.L    -$32CC(A5)                        |42ADCD34
000004EE CLR.B    -$32D2(A5)                        |422DCD2E
000004F2 CLR.B    -$32D1(A5)                        |422DCD2F
000004F6 CLR.L    -$32D8(A5)                        |42ADCD28
000004FA CLR.L    -$32DC(A5)                        |42ADCD24
000004FE CLR.L    -$32E8(A5)                        |42ADCD18
00000502 UNLK     A6                                |4E5E
00000504 RTS                                        |4E75
00000506 DC.W     '.I'                              |A549
00000508 DC.W     'NI'                              |4E49
0000050A DC.W     'TH'                              |5448
0000050C DC.W     'EA'                              |4541
0000050E ORI.B    #$56,D0                           ;'V'
00000512 ORI.B    #$5F,D0
00000516 MOVE.L   (SP)+,D1
00000518 MOVE.W   (SP)+,D2
0000051A MOVEA.L  (SP)+,A1
0000051C MOVEA.L  (SP)+,A0
0000051E MOVE.L   D1,-(SP)
```

MACSBUG SYMBOLS

```
** CODE ID= 1
%INITHEA
%_RESETW
%_CLOSE
```

USER GUIDE (VERY SHORT!)

The principle is as follows : after selecting a file, the application displays in the RSRC FILE WINDOW, the first resource available which type is the last specified (the default is CODE), and in dump or dissasemblage mode (the default is dump). If some symbols are available you can find them in the SYMBOLS WINDOW.

Then you can view the other resources by specifying their IDs (see **Resource Table** item in the Rsrc menu for the list of the available resources).

Each resource is shown by block of 512 bytes in dump mode, and by block of roughly 128 bytes otherwise. The vertical scrolling bar allows you to view the hidden part of the current block (the screen is generaly too short to display the entire block), whereas the horizontal scrolling bar allows you to view the other blocks of the current resource.

To patch into a resource, use the **Pach Mode** item in the Rsrc menu, and valid your modifications with the **Write Block** item. If you have made a mistake, the **Restore Block** item may be used.

Note : If you have **modified** the current block and you decide to quit the application or chan-ge the current block (there are many way to do this, see menu descriptions), **all the mo-difications are lost**. However if you have **saved** them, a dialog box appears asking you if you really want the changes to be permanent.

You can also save or print disassembled or dumped resources. This process can be cancelled at any time.

The other capabilities of this application are described menu by menu, in the next pages.

USEFUL FEATURES

- **Easy way to update the list of OS and Toolbox traps**.

  See the *RSCV_Compiler* documentation. However if you update these lists, please send me back the corresponding text file, or at least post it on the net, to allow every one else to update their own files.

- **Easy way to change the 'abort' character when saving or printing rsrc**.

  As there exist different keyboards (for example AZERTY and QWERTY), it is sometimes useful to modify this character. The default one is ';' but if you want to change it, simply edit the '**STR** ' resource (**ID 127**), and replace the first character of the string.

  Note : the string must be at least one-byte long or unpredictable results could arise.

- **Easy way to change the list of resources for symbols scanning.**

  When *RSC_Viewer* opens a file, first it scans all the resources of the current type to search for symbolic information. However resources which do not contain 680xx code never have symbols (MENU resources for example). So in order to avoid scanning all the resources, a list is provided.

  This list contains all the resources where symbolic information can be found (CODE, DRVR, FKEY, etc). If you want to restrict it or expand it, simply edit the '**FSYM**' resource (list of ResTypes).

  Note : if you edit this resource, be sure the resource length is a multiple of four, or the application will crash (a ResType is four-byte long).

THE PACKAGE

The *RSC_Viewer* package is distributed as **shareware**. If you find it useful, please send $25 to the following address, to become a registered user :

François Menneteau
3 Chemin de L'Eglise
38100 Grenoble
FRANCE.

Please send NO cheques, as banks tend to charge substantial amounts of money for cashing foreign cheques. So, send cash or money order only.

In return, I will send you the latest version of the package, which is made of the following files:

1.	**RSC_Viewer**	:	the main application;
2.	**RSC_Viewer.Help**	:	its help file;
3.	**ReadMe**	:	
	RSCV_COMPILER		
4.	**RSCV_Compiler**	:	an application to update RSC_Viewer resources;
5.	**GlobalVariables**	:	the list of Macintosh global variables;
6.	**ListOfTraps**	:	the list of Macintosh traps;
	DOCUMENTATION		
7.	**RSC_Viewer.Doc**	:	this file;
8.	**RSCV_Compiler.Doc**	:	obvious!

the *RSC_Viewer* application requires HFS, System 4.1 or later and at least 500 Kbytes of free memory.

WARRANTY

No warranty, expressed or implied, is provided with *RSC_Viewer*. If you find a bug, let me know and I'll fix it.

Since this application manipulates resources, you **can** easily corrupt them. Thus, when using it, be careful to work on files copy. Otherwise, **I am not responsible for any incidental or consequential losses resulting from its use or misuse**.

BUG REPORTS

RSC_Viewer is at least compatible with Macintosh Plus, SE, SE/30, II, IIsi and Classic, and has been successfully tested under System 4.3, 6.0.5 and 7.0b1.

It seems to be finder friendly, however, like many resource-editors, some problems may occur (see known bugs).

Any questions or bug reports can be sent to the author at the following network address :

iron@imag.fr

Known bugs :

— Under Multifinder, the first time you paste something from *RSC_Viewer* into the Album, the content is sometimes not what you expect. Then paste again and it will work fine. Under 6.0.5 many applications seems to have the same problem.

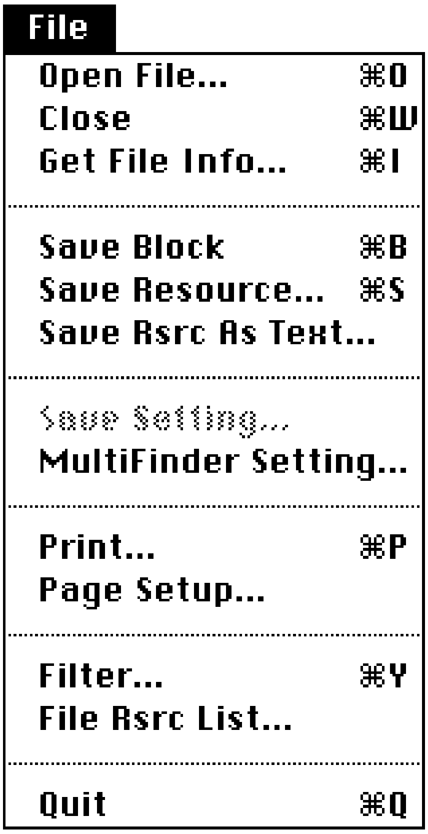
— Though it is possible to open a file which is already open (like the System file for example), it is not recommanded to do so, especially under Multifin-der. Some crashes are possible even after you quit *RSC_Viewer* .

CREDITS

Countless thanks go out to O. Taramasco, who has patiently waited for his Mac to be available. Every body should be as lucky as I am to have such a great friend.

Many thanks also to Ando Sonenblick who sent me a very interesting list of low memory global variables, and to every one on the net who helps me solving my software development problems.

FILE MENU DESCRIPTION



• Open file

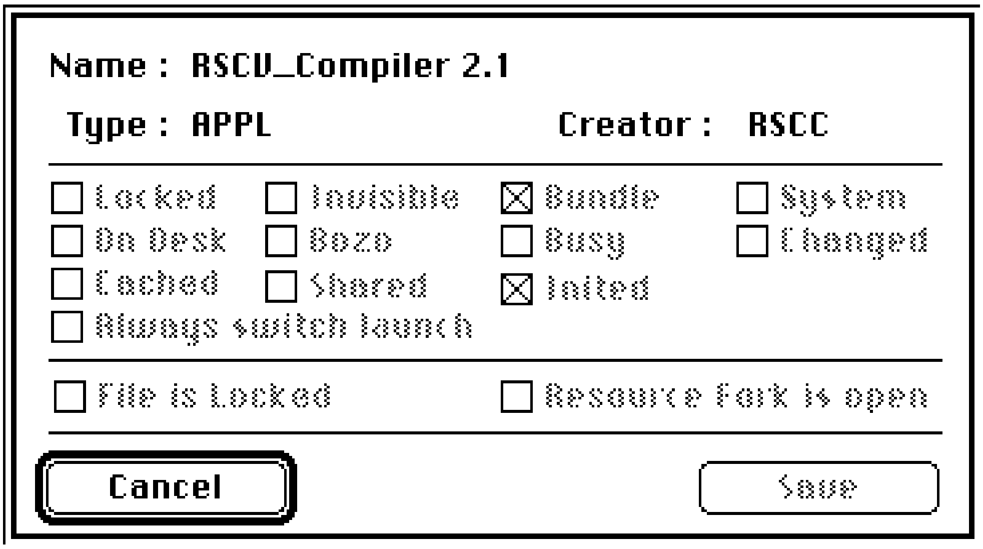
Presents a Standard File package box. The files which are listed have all a Type and a Creator specified in the filter dialog box (see **Filter** item). The default is **APPL** for Type and nothing for Creator.

• Close

Closes the front window (which can belong to the application or to a Desk Accessory under Finder).

• File Info

Shows the main attributes (file creator, file type...) about the file. Every thing is dimmed because the Save operation is not implemented (there are enough applications and DAs that do this job nicely…).



Name : RSCV_Compiler 2.1

Type : APPL Creator : RSCC

□ Locked □ Invisible ☒ Bundle □ System
□ On Desk □ Bozo □ Busy □ Changed
□ Cached □ Shared ☒ Inited
□ Always switch launch

□ File is Locked □ Resource Fork is open

• Save Block

Saves the current dumped? or disassembled block. The first time you use this item, a standard file package box proposes a file name (**<file name>.RSRC.Samp**). Then, each time you save an other block, it is appended to that file. This item may be very useful if you need only to retrieve some pieces of resource(s).

• Save Resource

Saves the dumped or disassembled resource. The default mode is save current resource. You can change this mode by selecting the **Option** item in the Misc menu.

Each time you use this item, a standard file package box asks for a file name, which will contain the dumped or disassembled resource(s) specified by the **Option** item.

The name of the save file is the concatenation of the current file and a suffix which characterize the set of resources saved :

—	**<file name>.<ResType>.<id>**	for an unique resource.
—	**<file name>.<ResType>.List**	for a list of resources of the given type.
—	**<file name>.<ResType>.All**	if all the resources of the given type are saved.

• Save Resource As Text

Saves the resource without formatting it. You can only save resources that do not belong to the '**FSYM**' resource i.e. resources that do not contain instructions. This item is really useful if you want to save the entire content of a '**STR#**' resources for example. It is an extension of the **Copy As Text** item in the menu Edit (see Edit Menu description for more information).

• Save Setting

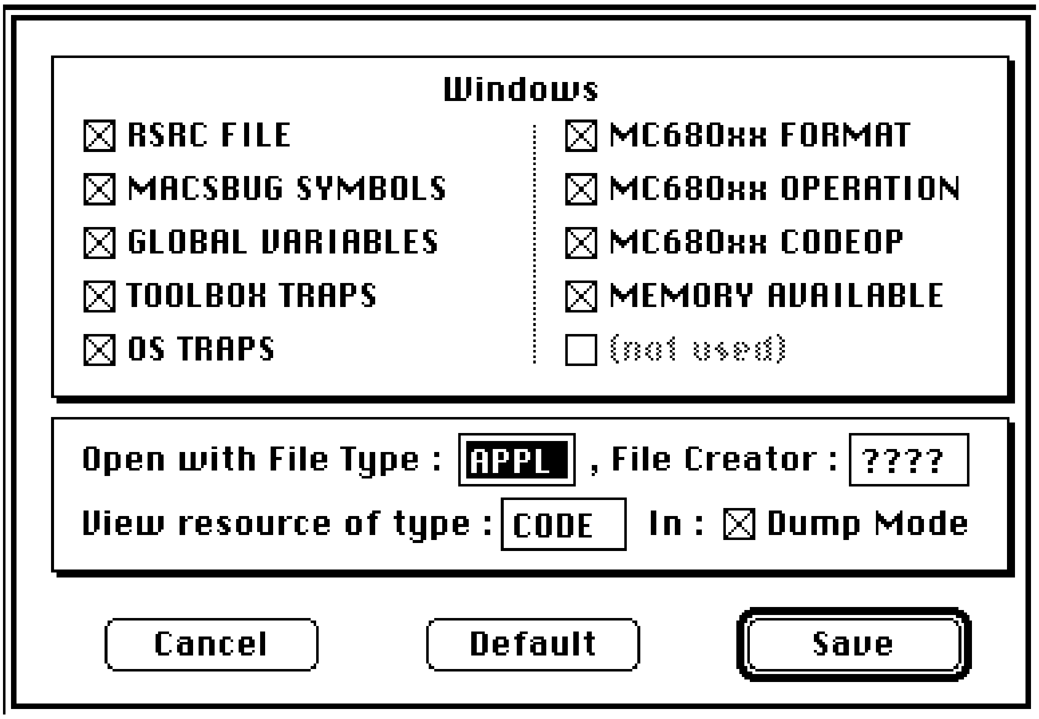
This item allows the user to save :

- The application's different windows current locations,
- The default File Type and File Creator,
- The default resource type to load and its default mode (dump or disassembly).

This setting will be take into account the next time you launch the *RSC Viewer* application.

Depending on what you set or unset the effects are very different :

> • if you activate the checkbox of a window (see dialog box below), its current location is saved when you choose **OK**. Otherwise its default location is saved. Of course, when you open again *RSC_Viewer*, each window is displayed (if visible) at its saved location.
>
> • The **Default** button restores the default position for all the windows (very useful when you switch from a Mac II to a Mac SE screen for exemple), the default File Type i.e. **APPL**, the default file Creator (none), the default resource type i.e. **CODE**, and the default mode i.e. **dump**.



Windows

☒ RSRC FILE
☒ MACSBUG SYMBOLS
☒ GLOBAL VARIABLES
☒ TOOLBOX TRAPS
☒ OS TRAPS
☒ MC680xx FORMAT
☒ MC680xx OPERATION
☒ MC680xx CODEOP
☒ MEMORY AVAILABLE
□ (not used)

Open with File Type : APPL , File Creator : ????

View resource of type : CODE In : ☒ Dump Mode

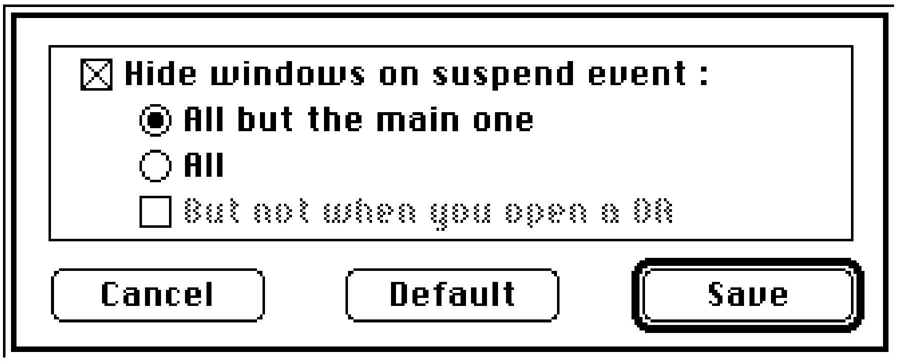
• Multifinder Setting

As you can open up to nine windows, it is sometimes desirable to close those cumbersome windows that hide your screen, when you switch from *RSC Viewer* to an other application (under

Multifinder).

However, it is also a good idea to keep at least one window visible, to quickly return to the *RSC Viewer* application.

This item allows you to configure the way your different windows will react to a context switch. It presents a dialog box which look like this :



☒ Hide windows on suspend event :

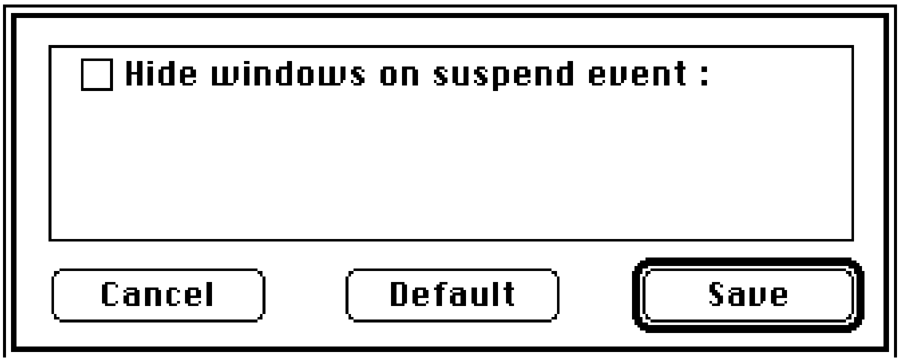
● All but the main one

○ All

□ But not when you open a DA

You can choose either to hide **“all the windows on suspend event”** or **“all but the main one”** (the RSRC FILE WINDOW).

If you do not want to hide the windows, click into the corresponding check box. The dialog will then look like this :



□ Hide windows on suspend event :

Note : This setting is the default one.

• Print

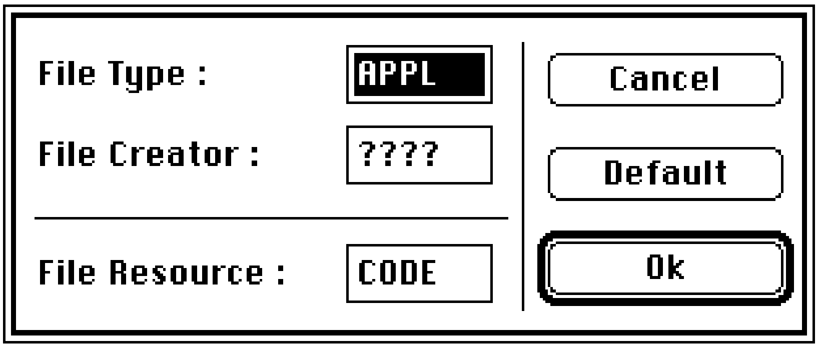
by default, prints the entire current resource. You can change this default mode by using the **Option** item which allows you to specify which resource(s) and within which addresses range you want to print. It is possible to cancel a printing operation (default is the ⌘ - ’;’ sequence, see the USEFUL FEATURES chapter if you want to change it).

• Page Setup

Classic…

• Filter

Presents a dialog box (see figure), in which you can specify a "File Type" and a "File Creator" for the filter procedure used by the Standard File Package. If you want any kind of "File Type", or "File Creator", put a blank or ???? value. In addition, you can choose the type of the resource you want to study. If the File Resource is undefined, only the **File Rsrc list** item is available.



File Type : APPL

File Creator : ????

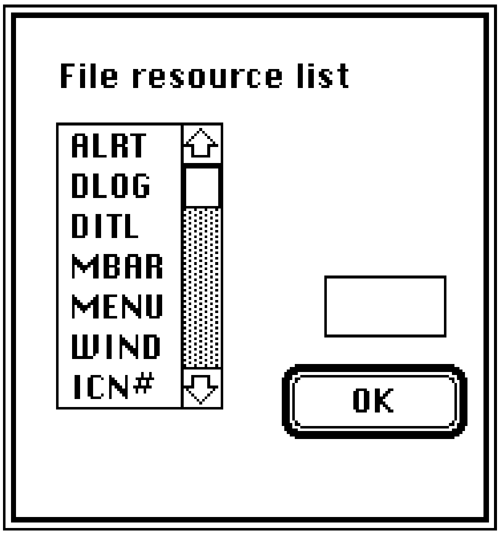
File Resource : CODE

• File Rsrc List

Presents a Standard File Package dialog box (with the same filter procedure as in **Open** item). Once you have chosen a file, a dialog box appears, displaying the list of all the resources available in that file.

You can choose a specific resource by selecting it and double-clicking in the list-window or by writing its name directly in the edit box (don't forget a resource type is four bytes long). The selected resource becomes the current File resource (see **Filter** dialog box).

Then the application opens the file and loads the required resources.



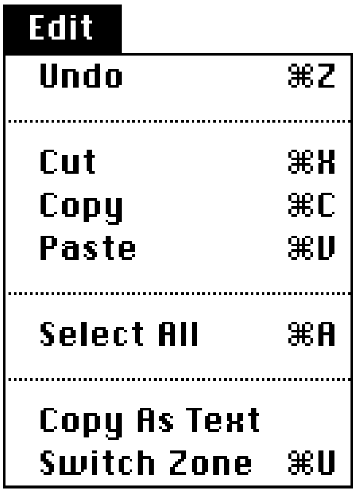
File resource list

ALRT
DLOG
DITL
MBAR
MENU
WIND
ICN#

• **Quit**

Leaves the application.

EDIT MENU DESCRIPTION



Only copy operations are possible in *RSC_Viewer* i.e. Cut and Paste operation won’t work.

However the ⌘-C/V/X mechanism is fully available in dialogs (thus, for example, you can copy a trap name from the OS TRAPS WINDOW, and paste it in the find dialog box).

• Undo

Not implemented yet ! (in fact will never be implemented…).

• Cut

You are not allowed to cut any pieces of resources.

• Copy

Nothing special to say.

• Paste

You are not allowed to paste anything into a resource. It is mostly for safety purpose (patching a resource is sufficiently dangerous not to introduce anymore problems) and data consistency pur-pose (for example it is difficult to prevent pasting some ASCII text fragments into the hexadecimal zone of a dumped resource).

• Select All

Selects the content of the entire window. For the RSRC FILE WINDOW only the content of the current block is selected.

• Copy As Text

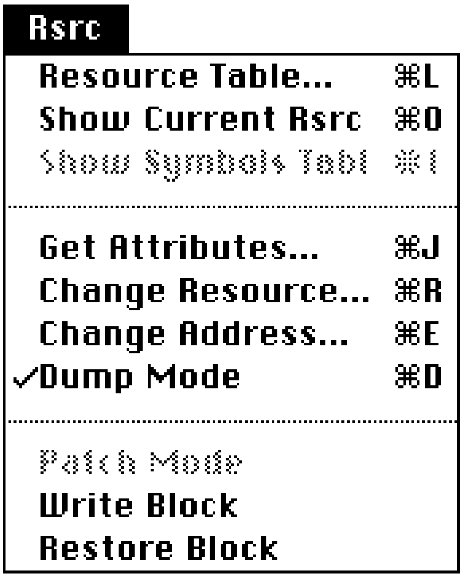
As in dump mode, you get both hexadecimal and ASCII values, it is sometimes preferable to retrieve only the ASCII part (think about '**STR#**', or '**STR** ' resources for example).

The only restriction is that, whatever you select, the minimum you can copy is 16 bytes (i.e. the ASCII part of a line). Thus, in short, selecting even any one byte of a line, selects in fact the whole ASCII part of that line.

• Switch Zone

If the cursor is in the hexadecimal part of a dumped line, then it goes to its corresponding position in the ASCII part (and vice versa in the ASCII part). Seems very useful…

RSRC MENU DESCRIPTION



• Resource Table

Shows the list of all resources of a given type present in the file. For each resource its ID, its SIZE (in decimal) and its NAME (if available) are displayed.

If you want to view a particular resource, you can select it directly by simply double-clicking (this overide the use of the **Change Resource** item).

When the resource is of type CODE, this option shows the main entry point of the application. Unfortunately, some jump tables are not standard, and in which case it is impossible to give the correct entry point. You can however find it, by disassembling the jump table at address $00000010.

• Show Current Rsrc

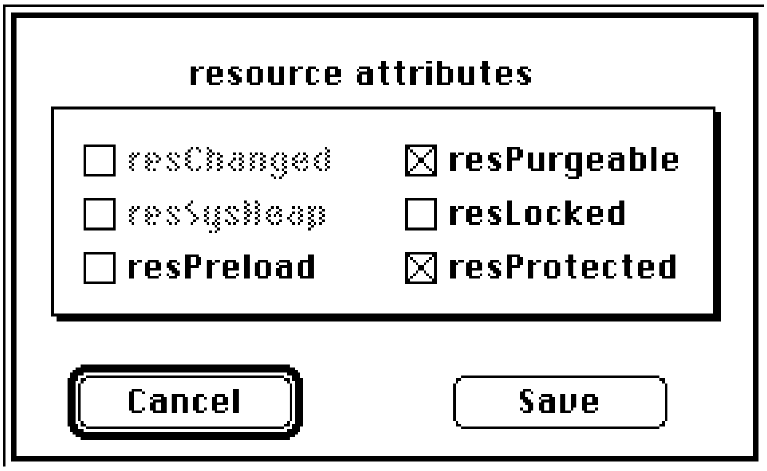
Allows the window which contains the current resource (FILE RSRC WINDOW) to become the front window.

• Show Symbol Table

Allows the window which contains the list of symbol names (SYMBOLS WINDOW) to become the front window. If the window is not visible, then it is first displayed.

• Get Attributes

Displays the attributes of the current resource. It is possible to modify all items but the **resChanged** and **resInHeap** attributes (for obvious reasons). If you set the **resProtected** bit on and if you have modified the current block, you won't be authorized to save it. In addition, if you are in **Pach Mode**, this mode becomes disabled.

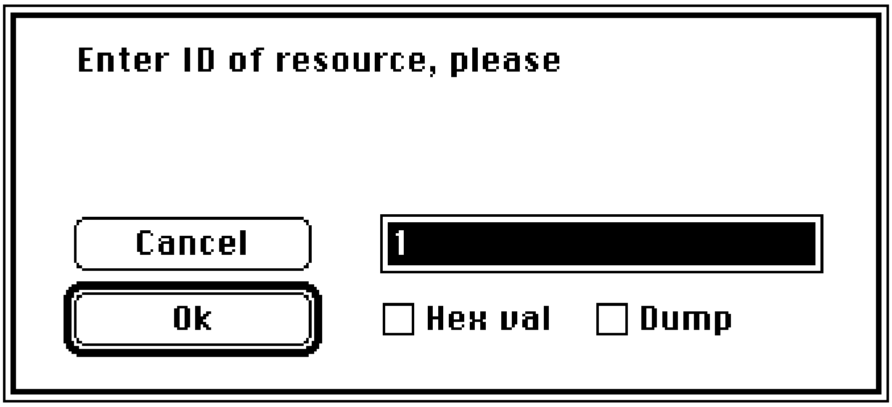


resource attributes

☐ resChanged ☒ resPurgeable
☐ resSysHeap ☐ resLocked
☐ resPreload ☒ resProtected

• Change Resource

Changes the current resource ID. You can enter the resource ID in hexadecimal (HexVal checkbox active), or in decimal (HexVal checkbox inactive). In addition, you can choose to disassemble the resource (Dump checkbox inactive) or to dump it (Dump checkbox active).

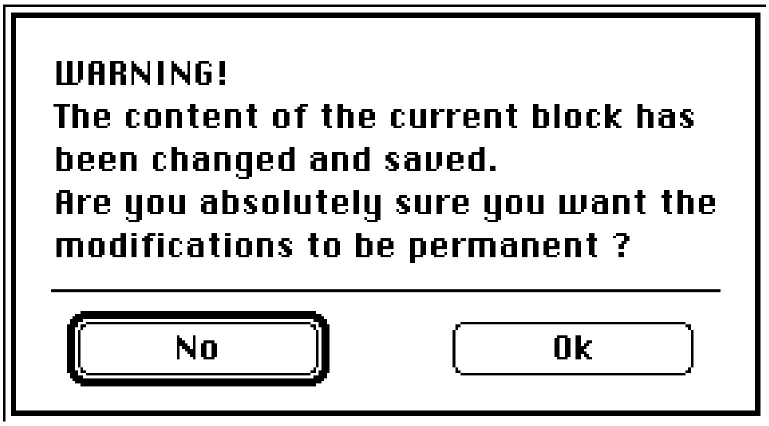


Enter ID of resource, please

1

☐ Hex val ☐ Dump

If the current block has been modified and saved, the following dialog box is displayed :



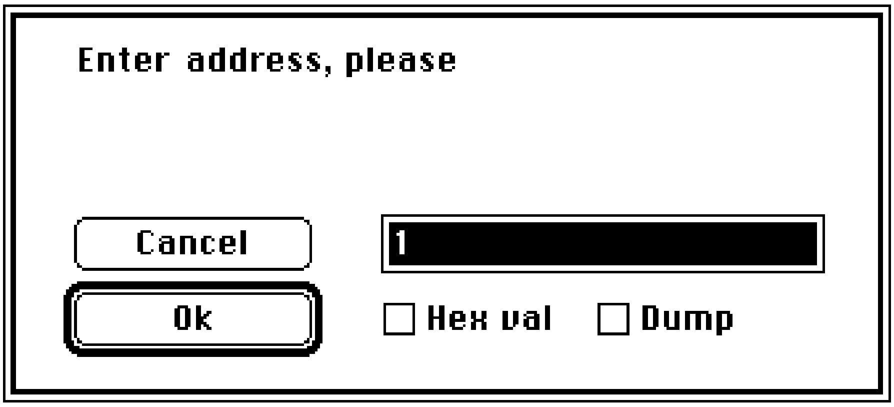
WARNING!
The content of the current block has been changed and saved.
Are you absolutely sure you want the modifications to be permanent ?

If you choose OK, you change the current resource (so the modifications are permanent, and the **Restore Block** item is disabled), otherwise you remain in the same resource, and you still have the possibility to use the **Restore Block** item.

This dialog box will appear as soon as you decide to leave a block that is modified and saved. The actions that can trigger this dialog are : **changing the current address** (menu com-mands, click in the horizontal scroll bar), **changing the current resource** (id), **switching from dump mode to disassemblage mode**, and l**eaving the application**.

• Change Address

Use it, when you want to select a new address within the current resource.



Enter address, please

1

• Dump mode

Allows you to switch between dump and disassemblage mode within the current resource.

• Patch Mode

Enters/leaves the patch mode, which allows you to write into resources. **Be careful...**

If patching is allowed, a ** Patch ** menu appears (it is an empty one and it is only for information purpose).

For instance, you can patch only if you are in dump mode. In a further release, an assembler will be available (to be used in disassembly mode).

• Write block

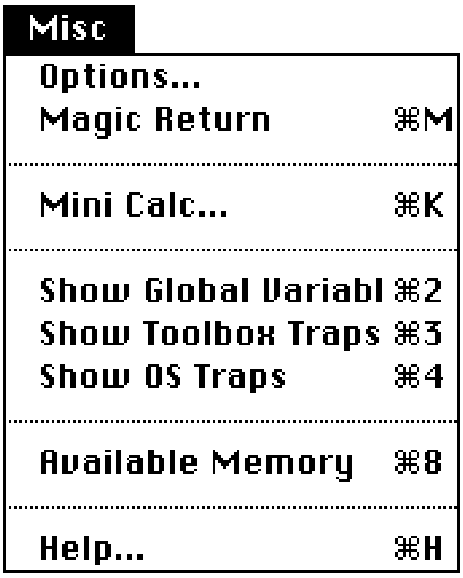
Writes the current block into the current resource.

Note : if you change the current block or the current resource before saving it, all the modi-fications are lost i.e. the resource is not changed.

• Restore Block

Restores the old values (those before the write operation).

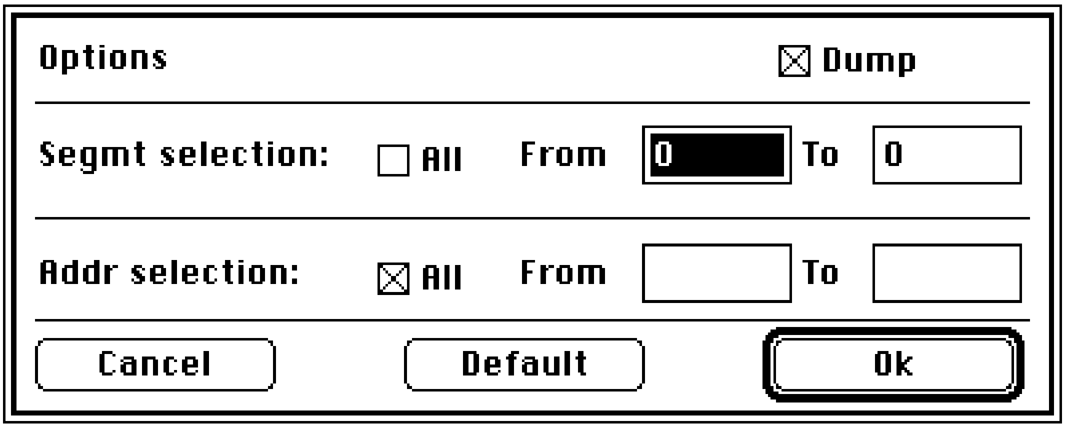
MISC MENU DESCRIPTION



• Options

Allows you to change the default mode for finding strings and printring or saving resour-ces. You can select the addresses range, and the resource IDs range (of course, resource IDs must be valid ones).

The **Default** button restores the default values i.e. actions occur within the current resource, from its beginning to its end. **Note :** Each time you select a new resource, the options are set to their default values.



Options ☒ Dump

Segmt selection: □ All From 0 To 0

Addr selection: ☒ All From To

• Magic Return

Each time you call **Change resource**, **Change address** or each time you select an address or a symbol by double-clicking, the previous address (offset[rsrc number]) is recorded in a LIFO stack.

This stack is able to store **ten** different addresses. Thus choosing **Magic Return** allow you to return to previous addresses within the limit of the ten stored adresses in the stack.

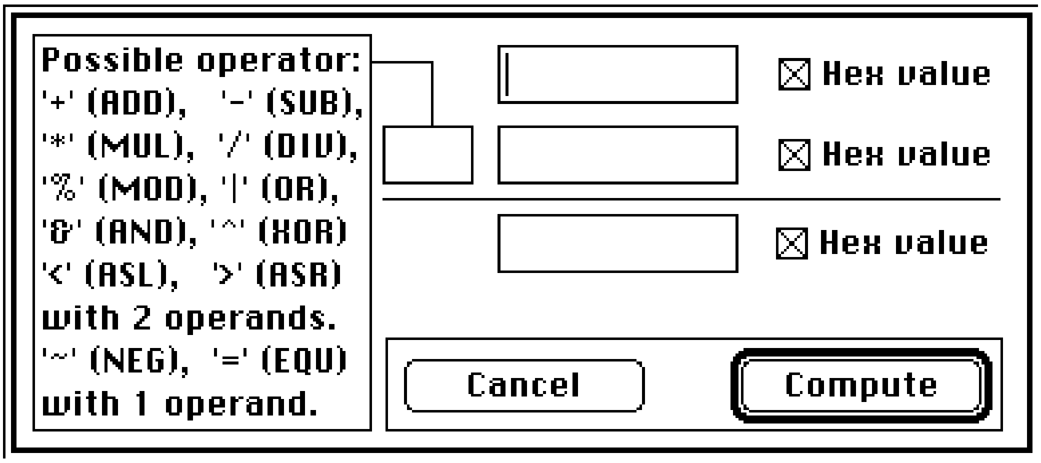
This item is very useful when there is a JSR instruction somewhere in a resource and you want to see the corresponding code, but you don't want to loose the address where the JSR was.

• Mini Calc

Emulates a mini calculator in hexadecimal and/or decimal mode. This calculator is word or long-word oriented (a byte is interpreted as a word). The available operators are :

- [two operands] : + (ADD), **-** (SUB), * (MUL), / (DIV), **%** (MOD),
- [two operands] : | (OR), **&** (AND), > (ASR), < (ASL), ^ (XOR),
- [one operand (the top one)] : ~ (NOT), = (EQU).

The '=' operator is useful to convert number (decimal to hexadecimal or hexadecimal to decimal conversion).



Possible operator:
'+' (ADD), '-' (SUB),
'*' (MUL), '/' (DIV),
'%' (MOD), '|' (OR),
'&' (AND), '^' (XOR)
'<' (ASL), '>' (ASR)
with 2 operands.
'~' (NEG), '=' (EQU)
with 1 operand.

☒ Hex value

☒ Hex value

☒ Hex value

• Show Global Variables

Allows the window which contains the list of global variables (GLOBAL VARIABLES WINDOW) to become the front window. If the window is not visible, then it is first displayed.

If you want to view a low memory address, you can do it simply by pressing the command key and double-clicking on the address part ($xxx:size).

• Show Toolbox Traps

Allows the window which contains the list of toolbox traps (TOOLBOX TRAPS WINDOW) to become the front window. If the window is not visible, then it is first displayed.

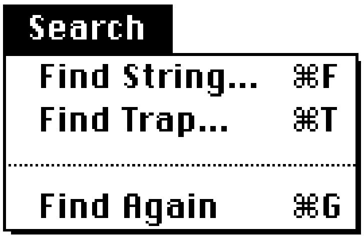
• Show OS Traps

Allows the window which contains the list of OS traps (OS TRAPS WINDOW) to become the front window. If the window is not visible, then it is first displayed.

• Available Memory

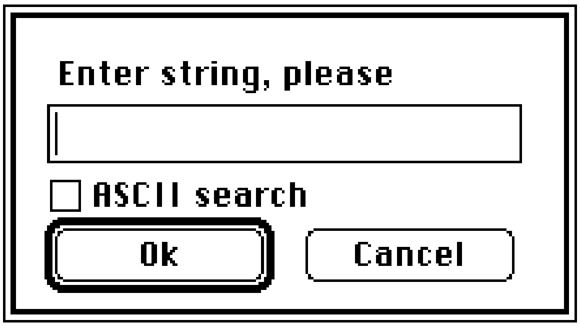
Allows the window which contains the value (in Kbytes) of the free memory (AVAI-LABLE MEMORY WINDOW) to become the front window. If the window is not visible, then it is first displayed.

SEARCH MENU DESCRIPTION



• Find String

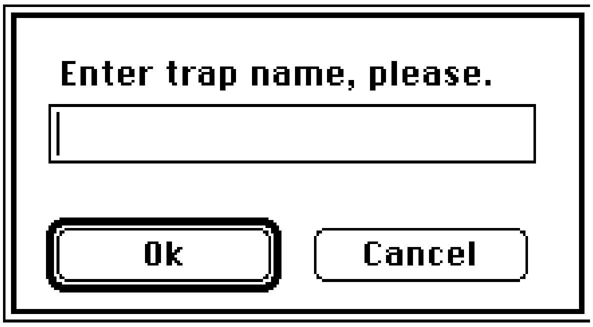
If you choose the ASCII checkbox, the string is interpreted as an ASCII one, otherwise it is interpreted as an hexadecimal value, and in that case, you can find a word (two bytes) or a long (four bytes).



• Find Trap

Trap name search (always in ASCII mode). It is possible to search for a trap with some bits set (see examples).

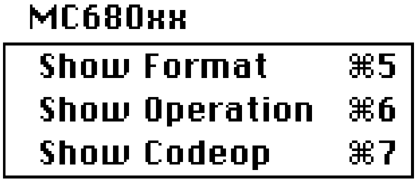
To search for \x(NewHandle,CLEAR), enter \x(NewHandle,9) (CLEAR bit is bit 9).
For \x(GetTrapAddress,NEWTOOL), enter \x(GetTrapAddress,9,10).



• Find Again

As soon as a search string or a trap string is defined, this item becomes available. But if you change the current window, it is disabled since the search parameters are no longer valid.

MC680XX MENU DESCRIPTION

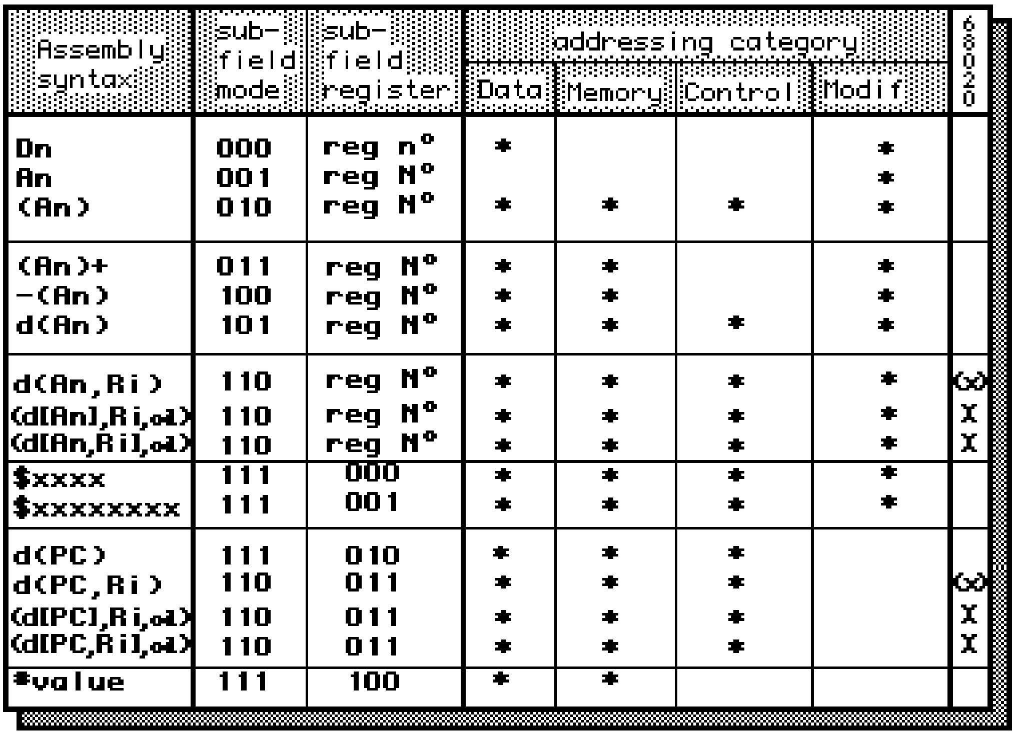


MC680xx

Show Format ⌘5
Show Operation ⌘6
Show Codeop ⌘7

• Show Format

Allows the window which contains the 680xx format to become the front window. If the window is not visible, then it is first displayed.

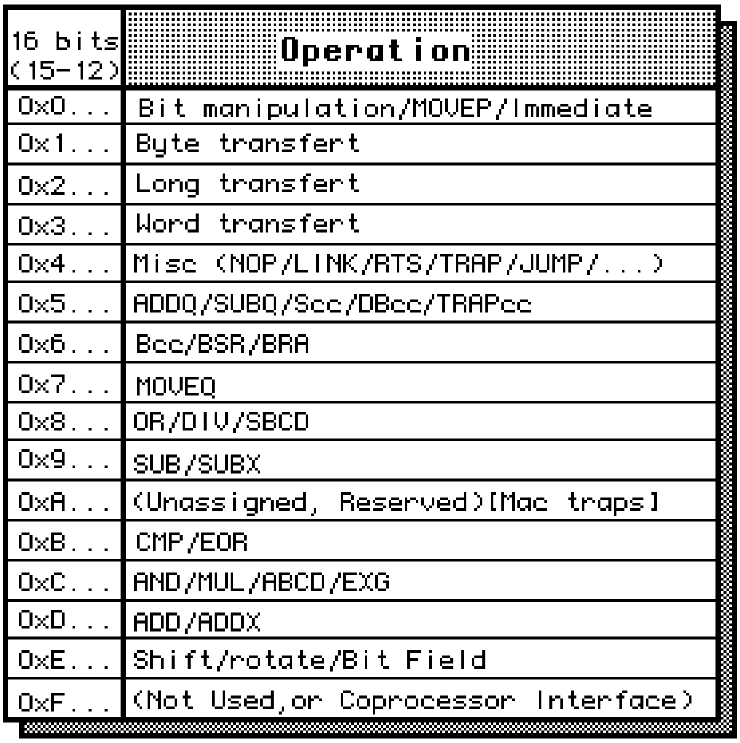


Assembly syntax	sub-field mode	sub-field register	addressing category				68020
			Data	Memory	Control	Modif	
Dn	000	reg n°	*			*	
An	001	reg N°				*	
(An)	010	reg N°	*	*	*	*	
(An)+	011	reg N°	*	*		*	
-(An)	100	reg N°	*	*		*	
d(An)	101	reg N°	*	*	*	*	
d(An,Ri)	110	reg N°	*	*	*	*	(x)
(d[An],Ri,od)	110	reg N°	*	*	*	*	X
(d[An,Ri],od)	110	reg N°	*	*	*	*	X
$xxxx	111	000	*	*	*	*	
$xxxxxxxx	111	001	*	*	*	*	
d(PC)	111	010	*	*	*		
d(PC,Ri)	110	011	*	*	*		(x)
(d[PC],Ri,od)	110	011	*	*	*		X
(d[PC,Ri],od)	110	011	*	*	*		X
#value	111	100	*	*			

This panel is very useful, if you want to build an instruction on the fly (good luck!). Assuming you know its internal representation, see Codeop window for some examples.

• Show Operation

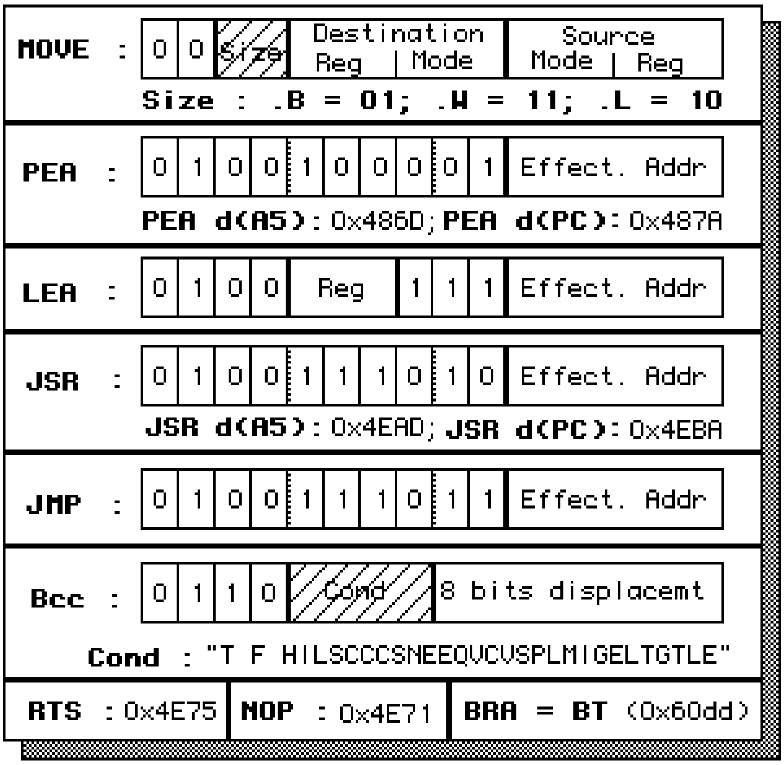
Allows the window which contains the 680xx operation to become the front window. If the window is not visible, then it is first displayed.



16 bits (15-12)	Operation
0x0...	Bit manipulation/MOVEP/Immediate
0x1...	Byte transfert
0x2...	Long transfert
0x3...	Word transfert
0x4...	Misc (NOP/LINK/RTS/TRAP/JUMP/...)
0x5...	ADDQ/SUBQ/Scc/DBcc/TRAPcc
0x6...	Bcc/BSR/BRA
0x7...	MOVEQ
0x8...	OR/DIV/SBCD
0x9...	SUB/SUBX
0xA...	(Unassigned, Reserved)[Mac traps]
0xB...	CMP/EOR
0xC...	AND/MUL/ABCD/EXG
0xD...	ADD/ADDX
0xE...	Shift/rotate/Bit Field
0xF...	(Not Used,or Coprocessor Interface)

• Show Codeop

Allows the window which contains some 68000 instructions to become the front window. If the window is not visible, then it is first displayed. Update it if you need.



MOVE :	0	0	Size	Destination Reg | Mode	Source Mode | Reg	

Size : .B = 01; .W = 11; .L = 10

PEA :	0	1	0	0	1	0	0	0	0	1	Effect. Addr

PEA d(A5): 0x486D; **PEA d(PC)**: 0x487A

LEA :	0	1	0	0	Reg	1	1	1	Effect. Addr

JSR :	0	1	0	0	1	1	1	0	1	0	Effect. Addr

JSR d(A5): 0x4EAD; **JSR d(PC)**: 0x4EBA

JMP :	0	1	0	0	1	1	1	0	1	1	Effect. Addr

Bcc :	0	1	1	0	Cond	8 bits displacemt

Cond : "T F HILSCCCSNEEQVCVSPLMIGELTGTLE"

RTS : 0x4E75	**NOP** : 0x4E71	**BRA = BT** (0x60dd)