

**MCC\_HexEdit**

COLLABORATORS

	TITLE : MCC_HexEdit		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		August 25, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>MCC_HexEdit</b>	<b>1</b>
1.1	MCC_HexEdit.doc . . . . .	1
1.2	HexEdit/--background-- . . . . .	1
1.3	HexEdit/MUIA_HexEdit_ActiveField . . . . .	2
1.4	HexEdit/MUIA_HexEdit_AddressChars . . . . .	2
1.5	HexEdit/MUIA_HexEdit_BaseAddressOffset . . . . .	3
1.6	HexEdit/MUIA_HexEdit_BytesPerColumn . . . . .	3
1.7	HexEdit/MUIA_HexEdit_BytesPerLine . . . . .	3
1.8	HexEdit/MUIA_HexEdit_ByteValue . . . . .	4
1.9	HexEdit/MUIA_HexEdit_ColumnsPerLine . . . . .	4
1.10	HexEdit/MUIA_HexEdit_CursorAddress . . . . .	4
1.11	HexEdit/MUIA_HexEdit_CursorNibble . . . . .	5
1.12	HexEdit/MUIA_HexEdit_CursorVisible . . . . .	5
1.13	HexEdit/MUIA_HexEdit_EditMode . . . . .	6
1.14	HexEdit/MUIA_HexEdit_First . . . . .	6
1.15	HexEdit/MUIA_HexEdit_FirstLine . . . . .	6
1.16	HexEdit/MUIA_HexEdit_FullRefresh . . . . .	7
1.17	HexEdit/MUIA_HexEdit_HighBound . . . . .	7
1.18	HexEdit/MUIA_HexEdit_LowBound . . . . .	8
1.19	HexEdit/MUIA_HexEdit_MoveCursor . . . . .	8
1.20	HexEdit/MUIA_HexEdit_NibbleValue . . . . .	9
1.21	HexEdit/MUIA_HexEdit_PropObject . . . . .	9
1.22	HexEdit/MUIA_HexEdit_SelectMode . . . . .	9
1.23	HexEdit/MUIA_HexEdit_VisibleLines . . . . .	10
1.24	HexEdit/MUIM_HexEdit_CreateDisplayAddress . . . . .	10
1.25	HexEdit/MUIM_HexEdit_FilterChar . . . . .	11
1.26	HexEdit/MUIM_HexEdit_ReadMemoryByte . . . . .	12
1.27	HexEdit/MUIM_HexEdit_Redraw . . . . .	12
1.28	HexEdit/MUIM_HexEdit_WriteMemoryByte . . . . .	13

## Chapter 1

# MCC\_HexEdit

### 1.1 MCC\_HexEdit.doc

```
--background--
MUIA_HexEdit_ActiveField()
MUIA_HexEdit_AddressChars()
MUIA_HexEdit_BaseAddressOffset()
MUIA_HexEdit_BytesPerColumn()
MUIA_HexEdit_BytesPerLine()
MUIA_HexEdit_ByteValue()
MUIA_HexEdit_ColumnsPerLine()
MUIA_HexEdit_CursorAddress()
MUIA_HexEdit_CursorNibble()
MUIA_HexEdit_CursorVisible()
MUIA_HexEdit_EditMode()
MUIA_HexEdit_First()
MUIA_HexEdit_FirstLine()
MUIA_HexEdit_FullRefresh()
MUIA_HexEdit_HighBound()
MUIA_HexEdit_LowBound()
MUIA_HexEdit_MoveCursor()
MUIA_HexEdit_NibbleValue()
MUIA_HexEdit_PropObject()
MUIA_HexEdit_SelectMode()
MUIA_HexEdit_VisibleLines()
MUIM_HexEdit_CreateDisplayAddress()
MUIM_HexEdit_FilterChar()
MUIM_HexEdit_ReadMemoryByte()
MUIM_HexEdit_Redraw()
MUIM_HexEdit_WriteMemoryByte()
```

### 1.2 HexEdit/--background--

NAME  
HexEdit -- ... (V14)

FUNCTION  
This class provides you with an easy way to present hexadecimal

---

dump of a memory region to the user. Additionally, memory contents are visible in ASCII representation. Editing in both modes is possible. HexEdit class is very flexible, with many settable parameters and some critical methods (like byte reading and writing) waiting to be overloaded.

### 1.3 HexEdit/MUIA\_HexEdit\_ActiveField

#### NAME

MUIA\_HexEdit\_ActiveField, LONG [ISG] -- ... (V14)

#### SPECIAL INPUTS

MUIV\_HexEdit\_ActiveField\_HexDump

MUIV\_HexEdit\_ActiveField\_Chars

#### FUNCTION

With MUIA\_HexEdit\_ActiveField you can decide over which field of the object (hexdump or character) active cursor should be positioned.

#### NOTES

#### BUGS

No known bugs.

#### SEE ALSO

### 1.4 HexEdit/MUIA\_HexEdit\_AddressChars

#### NAME

MUIA\_HexEdit\_AddressChars, LONG [I.G] -- ... (V14)

#### SPECIAL INPUTS

MUIV\_AddressChars\_Auto

#### FUNCTION

This attribute lets you define how many characters will be devoted to representing addresses displayed by HexEdit. For example, if you specify a value of 4, all addresses will be displayed as 16 bit. Default is 8, i.e. 32 bit.

Address calculation's precision is not dependant on this attribute, it's for display purposes only. Internally all addresses are treated as LONGs.

#### NOTES

This value ranges from 1 to 8, but please see MUIM\_HexEdit\_CreateDisplayAddress().

#### BUGS

No known bugs.

---

SEE ALSO  
MUIM\_HexEdit\_CreateDisplayAddress()

## 1.5 HexEdit/MUIA\_HexEdit\_BaseAddressOffset

NAME  
MUIA\_HexEdit\_BaseAddressOffset, LONG [ISG] -- ... (V14)

FUNCTION  
The value of this attribute is added to the current address every time when it is about to be displayed. For example, if you have your data at 0x04000000 and you want the addresses to be shown as starting from 0, set this attribute to -0x04000000.

NOTES  
  
BUGS  
No known bugs.

SEE ALSO

## 1.6 HexEdit/MUIA\_HexEdit\_BytesPerColumn

NAME  
MUIA\_HexEdit\_BytesPerColumn, LONG [I.G] -- ... (V14)

FUNCTION  
With MUIA\_HexEdit\_BytesPerColumn you can define how many bytes should constitute one column. For example, default value of 4 groups memory dump in a following way:

```
01234567 01234567 01234567 01234567
```

By changing it to 3, you would get:

```
012345 670123 456701 234567
```

NOTES  
  
BUGS  
No known bugs.

SEE ALSO

## 1.7 HexEdit/MUIA\_HexEdit\_BytesPerLine

NAME  
MUIA\_HexEdit\_BytesPerLine, LONG [..G] -- ... (V14)

FUNCTION

---

With this attribute you can learn how many bytes are displayed in a single line of HexEdit object.

#### BUGS

No known bugs.

#### SEE ALSO

## 1.8 HexEdit/MUIA\_HexEdit\_ByteValue

#### NAME

MUIA\_HexEdit\_ByteValue, LONG [.SG] -- ... (V14)

#### FUNCTION

This attribute lets you read and modify the value of byte under the cursor.

#### NOTES

#### BUGS

No known bugs.

#### SEE ALSO

MUIA\_HexEdit\_NibbleValue()

## 1.9 HexEdit/MUIA\_HexEdit\_ColumnsPerLine

#### NAME

MUIA\_HexEdit\_ColumnsPerLine, LONG [I.G] -- ... (V14)

#### FUNCTION

By querying this attribute you can learn how many columns are currently displayed in one line of HexEdit object. By default, the number of columns varies with the size of the object. If, however MUIA\_HexEdit\_ColumnsPerLine is set at the creation time, the object will always use a value then specified.

#### BUGS

No known bugs.

#### SEE ALSO

## 1.10 HexEdit/MUIA\_HexEdit\_CursorAddress

#### NAME

MUIA\_HexEdit\_CursorAddress, LONG [.SG] -- ... (V14)

#### FUNCTION

Reading this attribute will return the cursor's current address.

---

Setting it will move the cursor to a new location, possibly scrolling and/or redrawing the display.

#### NOTES

Address, as usual, is specified as byte offset from `MUIA_HexEdit_LowBound()`

#### BUGS

No known bugs.

#### SEE ALSO

`MUIA_HexEdit_LowBound()`, `MUIA_HexEdit_HighBound()`,  
`MUIA_HexEdit_CursorNibble()`

## 1.11 HexEdit/MUIA\_HexEdit\_CursorNibble

#### NAME

`MUIA_HexEdit_CursorNibble`, LONG [ISG] -- ... (V14)

#### FUNCTION

With this attribute you can set or get the nibble over which the cursor is currently positioned.

#### RESULTS

0 for upper nibble, 1 for lower nibble.

#### NOTES

#### BUGS

No known bugs.

#### SEE ALSO

`MUIA_HexEdit_CursorAddress()`

## 1.12 HexEdit/MUIA\_HexEdit\_CursorVisible

#### NAME

`MUIA_HexEdit_CursorVisible`, BOOL [ISG] -- ... (V14)

#### FUNCTION

With this attribute you can set and get the state of cursor's visibility.

#### NOTES

Please do not attempt to switch off the cursor's while in the edit mode, as it may confuse the user.

#### BUGS

No known bugs.

#### SEE ALSO



## 1.13 HexEdit/MUIA\_HexEdit\_EditMode

### NAME

MUIA\_HexEdit\_EditMode, BOOL [ISG] -- ... (V14)

### FUNCTION

This attribute determines if HexEdit object will let the user edit a memory space that is under its control. If it is set to TRUE, user can input 0-9/a-f digits while in the hexdump part of the display and alphanumerics while in the character part.

### NOTES

### BUGS

No known bugs.

### SEE ALSO

MUIA\_HexEdit\_CursorVisible(), MUIM\_HexEdit\_WriteMemoryByte(), MUIM\_HexEdit\_FilterChar()

## 1.14 HexEdit/MUIA\_HexEdit\_First

### NAME

MUIA\_HexEdit\_First, LONG [ISG] -- ... (V14)

### FUNCTION

MUIA\_HexEdit\_First specifies offset (counting from MUIA\_HexEdit\_LowBound()) of the first byte to be displayed at the top of HexEdit object. This attribute is always rounded down to a multiply of MUIA\_HexEdit\_BytesPerLine().

### NOTES

In certain cases this attribute does not change, despite set()ting a value different from the current - for example when there are not enough data lines below to fill the entire HexEdit object. If you want to highlight certain position, please rather use MUIA\_HexEdit\_CursorAddress().

### BUGS

No known bugs.

### SEE ALSO

MUIA\_HexEdit\_CursorAddress()

## 1.15 HexEdit/MUIA\_HexEdit\_FirstLine

### NAME

MUIA\_HexEdit\_FirstLine, LONG [.S.] -- ... (V14)

### SPECIAL INPUTS

MUIV\_HexEdit\_FirstLine\_Up

MUIV\_HexEdit\_FirstLine\_Down

MUIV\_HexEdit\_FirstLine\_PageUp  
MUIV\_HexEdit\_FirstLine\_PageDown  
MUIV\_HexEdit\_FirstLine\_Top  
MUIV\_HexEdit\_FirstLine\_Bottom

#### FUNCTION

Changing this attribute lets you move around your memory space.

#### NOTES

Only values listed above are allowed as input.

#### BUGS

No known bugs.

#### SEE ALSO

## 1.16 HexEdit/MUIA\_HexEdit\_FullRefresh

#### NAME

MUIA\_HexEdit\_FullRefresh, BOOL [.S.] -- ... (V14)

#### FUNCTION

Setting this attribute to TRUE will force redraw of the entire object area next time MUIA\_HexEdit\_First() is set. Normally, only parts that are new on display would be drawn.

#### NOTES

This attribute may be helpful with some more exotic subclasses of HexEdit. Usually, you won't need to touch it, though.

#### BUGS

No known bugs.

#### SEE ALSO

MUIA\_HexEdit\_First()

## 1.17 HexEdit/MUIA\_HexEdit\_HighBound

#### NAME

MUIA\_HexEdit\_HighBound, LONG [I.G] -- ... (V14)

#### FUNCTION

With MUIA\_HexEdit\_HighBound you can define where in Amiga memory space ends the area you want HexEdit to operate on.

#### NOTES

This attribute MUST be specified, even if you overload MUIM\_HexEdit\_ReadMemoryByte() or MUIM\_HexEdit\_WriteMemoryByte()!

#### BUGS

No known bugs.

---

SEE ALSO  
MUIA\_HexEdit\_LowBound(), MUIM\_HexEdit\_ReadMemoryByte(),  
MUIM\_HexEdit\_WriteMemoryByte()

## 1.18 HexEdit/MUIA\_HexEdit\_LowBound

NAME  
MUIA\_HexEdit\_LowBound, LONG [I.G] -- ... (V14)

FUNCTION  
With MUIA\_HexEdit\_LowBound you can define where in Amiga memory space begins the area you want HexEdit to operate on.

NOTES  
This attribute MUST be specified, even if you overload  
MUIM\_HexEdit\_ReadMemoryByte() or MUIM\_HexEdit\_WriteMemoryByte()!

BUGS  
No known bugs.

SEE ALSO  
MUIA\_HexEdit\_HighBound(), MUIM\_HexEdit\_ReadMemoryByte(),  
MUIM\_HexEdit\_WriteMemoryByte()

## 1.19 HexEdit/MUIA\_HexEdit\_MoveCursor

NAME  
MUIA\_HexEdit\_MoveCursor, LONG [.S.] -- ... (V14)

SPECIAL INPUTS  
MUIV\_HexEdit\_MoveCursor\_Up  
MUIV\_HexEdit\_MoveCursor\_Down  
MUIV\_HexEdit\_MoveCursor\_Left  
MUIV\_HexEdit\_MoveCursor\_Right  
MUIV\_HexEdit\_MoveCursor\_PageUp  
MUIV\_HexEdit\_MoveCursor\_PageDown  
MUIV\_HexEdit\_MoveCursor\_Top  
MUIV\_HexEdit\_MoveCursor\_Bottom  
MUIV\_HexEdit\_MoveCursor\_WordLeft  
MUIV\_HexEdit\_MoveCursor\_WordRight  
MUIV\_HexEdit\_MoveCursor\_LineStart  
MUIV\_HexEdit\_MoveCursor\_LineEnd

FUNCTION  
With MUIA\_HexEdit\_MoveCursor you can control cursor movements exactly as if you were using the keyboard. Of course, they do not make much sense if the cursor is actually turned off.

NOTES  
Only values listed above are allowed as input.

BUGS

---

No known bugs.

SEE ALSO  
MUIA\_HexEdit\_CursorVisible()

## 1.20 HexEdit/MUIA\_HexEdit\_NibbleValue

NAME  
MUIA\_HexEdit\_NibbleValue, LONG [.SG] -- ... (V14)

FUNCTION  
This attribute lets you read and modify the value of nibble under the cursor. The value is always passed in a lower four bits.

NOTES

BUGS  
No known bugs.

SEE ALSO  
MUIA\_HexEdit\_ByteValue()

## 1.21 HexEdit/MUIA\_HexEdit\_PropObject

NAME  
MUIA\_HexEdit\_PropObject, LONG [.SG] -- ... (V14)

FUNCTION  
With MUIA\_HexEdit\_PropObject you can easily connect proportional gadget with the HexEdit object. Simply pass a pointer to prop object as the value of this attribute and you're done. HexEdit will handle all house-keeping for you. How convinient! ;)

NOTES

BUGS  
No known bugs.

SEE ALSO

## 1.22 HexEdit/MUIA\_HexEdit\_SelectMode

NAME  
MUIA\_HexEdit\_SelectMode, LONG [I.G] -- ... (V14)

SPECIAL INPUTS  
MUIV\_HexEdit\_SelectMode\_Nibble  
MUIV\_HexEdit\_SelectMode\_Byte

FUNCTION

---

This attribute defines if the cursor moving over hexadecimal data should span entire bytes or just single nibbles.

NOTES

BUGS

No known bugs.

SEE ALSO

## 1.23 HexEdit/MUIA\_HexEdit\_VisibleLines

NAME

MUIA\_HexEdit\_VisibleLines, LONG [..G] -- ... (V14)

FUNCTION

With this attribute you can learn how many text lines are currently visible in the HexEdit object. It may be useful for notifications.

BUGS

No known bugs.

SEE ALSO

## 1.24 HexEdit/MUIM\_HexEdit\_CreateDisplayAddress

NAME

MUIM\_HexEdit\_CreateDisplayAddress (V14)

SYNOPSIS

DoMethod(obj, MUIM\_HexEdit\_CreateDisplayAddress, UBYTE \*\*cp, ULONG address);

FUNCTION

HexEdit class calls this method to generate ASCII representation of current address. If you want to create some fancy-looking addresses, you can overload this method in your subclass.

NOTES

If you're overloading this method, you HAVE TO set MUIA\_HexEdit\_AddressChars() to length of the largest output of your address creator, so that HexEdit knows how much space allocate for it.

EXAMPLE

This is HexEdit's implementation of MUIM\_HexEdit\_CreateDisplayAddress method:

```
ULONG __asm _CreateDisplayAddress(REG(A0) struct IClass *cl, REG(A2) Object *obj ←
    , REG(A1) struct MUIM_HexEdit_CreateDisplayAddress *msg)
{
    struct Data *d = INST_DATA(cl,obj);
```

```

ULONG address;
UBYTE i;
UBYTE *hextable = "0123456789ABCDEF";

address = (d->base_address + msg->address) << (HE_MAX_ADDRESS_LEN - d->↵
    address_chars * 4);

for(i = 0; i < d->address_chars; i++)
{
    *(&msg->cp)++ = hextable[address >> (HE_MAX_ADDRESS_LEN - 4)];
    address <<= 4;
}

return(TRUE);
}

BUGS
No known bugs.

SEE ALSO
MUIA_HexEdit_AddressChars()

```

## 1.25 HexEdit/MUIM\_HexEdit\_FilterChar

NAME  
 MUIM\_HexEdit\_FilterChar (V14)

SYNOPSIS  
 DoMethod(obj, MUIM\_HexEdit\_FilterChar, ULONG value, UBYTE \*buffer);

FUNCTION  
 HexEdit class calls this method to generate ASCII representation of character it is currently processing. You can overload it to e.g. show only characters you want and replace others with dots.

EXAMPLE  
 This is example implementation of MUIM\_HexEdit\_FilterChar:

```

ULONG __asm _FilterChar(REG(A0) struct IClass *cl, REG(A2) Object *obj, REG(A1) ↵
    struct MUIP_HexEdit_FilterChar *msg)
{
    if(isascii(msg->value))
        *msg->buffer = msg->value;
    else
        *msg->buffer = '.';

    return(TRUE);
}

```

NOTES

BUGS  
 No known bugs.

SEE ALSO

## 1.26 HexEdit/MUIM\_HexEdit\_ReadMemoryByte

NAME

MUIM\_HexEdit\_ReadMemoryByte (V14)

SYNOPSIS

DoMethod(obj, MUIM\_HexEdit\_ReadMemoryByte, UBYTE \*value, ULONG address);

FUNCTION

HexEdit class calls this method to read a byte from memory. You can overload it in your subclass, so that operation on areas which are not present in Amiga's memory space is possible.

NOTES

Address, as usual, is specified as byte offset from MUIM\_HexEdit\_LowBound()

EXAMPLE

This is HexEdit's implementation of MUIM\_HexEdit\_ReadMemoryByte method:

```
ULONG __asm _ReadMemoryByte(REG(A0) struct IClass *cl, REG(A2) Object *obj, REG(←
    A1) struct MUIM_HexEdit_ReadMemoryByte *msg)
{
    struct Data *d = INST_DATA(cl, obj);

    *msg->value = *(UBYTE *) (d->bound_low + msg->address);

    return(TRUE);
}
```

BUGS

No known bugs.

SEE ALSO

MUIM\_HexEdit\_WriteMemoryByte()

## 1.27 HexEdit/MUIM\_HexEdit\_Redraw

NAME

MUIM\_HexEdit\_Redraw (V14)

SYNOPSIS

DoMethod(obj, MUIM\_HexEdit\_Redraw);

FUNCTION

Redraws the entire object's contents.

NOTES

BUGS  
No known bugs.

SEE ALSO

## 1.28 HexEdit/MUIM\_HexEdit\_WriteMemoryByte

NAME  
MUIM\_HexEdit\_WriteMemoryByte (V14)

SYNOPSIS  
DoMethod(obj, MUIM\_HexEdit\_ReadMemoryByte, ULONG value, ULONG address);

FUNCTION  
HexEdit class calls this method to write a byte to memory. You can overload it in your subclass, so that operation on areas which are not present in Amiga's memory space is possible.

NOTES  
Address, as usual, is specified as byte offset from  
MUIM\_HexEdit\_LowBound()

EXAMPLE  
This is HexEdit's implementation of MUIM\_HexEdit\_WriteMemoryByte method:

```
ULONG __asm _WriteMemoryByte(REG(A0) struct IClass *cl, REG(A2) Object *obj, REG(A1) struct MUIM_HexEdit_WriteMemoryByte *msg)
{
    struct Data *d = INST_DATA(cl,obj);

    *(UBYTE *) (d->bound_low + msg->address) = msg->value;

    return(TRUE);
}
```

BUGS  
No known bugs.

SEE ALSO  
MUIM\_HexEdit\_ReadMemoryByte()