

**binary\_dtc.doc**

COLLABORATORS
---------------

	TITLE : binary_dtc.doc		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		December 8, 2024	

REVISION HISTORY
------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>binary_dtc.doc</b>	<b>1</b>
1.1	binary_dtc.doc . . . . .	1
1.2	binary.datatype/binary.datatype . . . . .	1

# Chapter 1

## binary\_dtc.doc

### 1.1 binary\_dtc.doc

binary.datatype

### 1.2 binary.datatype/binary.datatype

#### NAME

binary.datatype -- data type for any binary file

#### FUNCTION

The binary data type, a base-class of all binary data, is used to load any binary file and displays the contents of the file in hex format.

#### PREFS

The data type tries to load the prefs file first from "PROGDIR:Prefs/DataTypes/binary.prefs" and then "ENV:DataTypes/binary.prefs" on each OM\_NEW method to set up the attributes !  
Up from version 39.10 it uses the ReadArgs() function to parse the prefs file. The template is :

NOASCII/S, NOWRAP/S, NONE/S, BYTE/S, WORD/S, LONG/S, BPL=BYTESPERLINE/N/K

NOASCII - sets BDTA\_ShowASCII to FALSE  
NOWRAP - sets BDTA\_DisplayWrap to FALSE  
NONE - sets BDTA\_DisplayHex to BDTDH\_NONE  
BYTE - sets BDTA\_DisplayHex to BDTDH\_BYTE  
WORD - sets BDTA\_DisplayHex to BDTDH\_WORD  
LONG - sets BDTA\_DisplayHex to BDTDH\_LONG  
BYTESPERLINE <bpl> or  
BPL <bpl> - sets BDTA\_BytesPerLine to <bpl> bytes

The options can be on several lines !

#### METHODS

OM\_NEW -- Create a new text object from a binary file in hex mode.

---

OM\_DISPOSE -- dispose a object

OM\_GET -- get a attribute of the object

OM\_SET -- set attributes of the object

OM\_UPDATE -- update some attributes of the object

GM\_LAYOUT -- Method to layout the hex text

GM\_RENDER -- draw the object

DTM\_WRITE -- DTWM\_RAW mode is supported

DTM\_PRINT -- prints the hex text

DTM\_DRAW -- draws the datatype in the given RastPort (This is  
expermental, I use this method for my new text.datatype to  
embed other datatype objects ! Don't use this at the moment !)

DTM\_TRIGGER -- trigger methods to let the user input a search string  
and to search next or previous occurence of that string :  
STM\_ACTIVATE\_FIELD (return) - opens the string requester. This  
requester is spawned asynconly !  
STM\_BROWSE\_NEXT ('>') - searchs next occurence  
STM\_BROWSE\_PREV ('<') - searchs previous occurence

DTBM\_GETSTRING -- opens a requester to let the user input the  
string !

DTBM\_SEARCHNEXT -- searchs for the next occurence of the specified  
string

DTBM\_SEARCHPREV -- searchs for the previous occurance of the given  
string

TAGS

BDTA\_Buffer -- (UBYTE \*) pointer to the buffer, which should be  
displayed.  
Applicability is (ISG).

BDTA\_BufferLen -- (ULONG) length of the buffer supplied with  
BDTA\_Buffer tag. This must be given if the buffer tag is  
specified.  
Applicability is (ISG).

BDTA\_BytesPerLine -- (UWORD) number of bytes per line.  
If BDTA\_DisplayHex is BDTDH\_WORD it must be a multiply of 2,  
if it is BDTDH\_LONG it must be a multiply of 4 !  
Default is 32.  
Applicability is (ISGNU).

BDTA\_DisplayHex -- (UWORD) type of the display. The following types  
are supported : BDTDH\_NONE - displays no hex values  
BDTDH\_BYTE - displays each byte in hex ( 8 bit)  
BDTDH\_WORD - displays each word in hex (16 bit)

---

BDTDH\_LONG - displays each long in hex (32 bit)  
Default is BDTDH\_LONG.  
Applicability is (ISGNU).

BDTA\_ShowASCII -- (BOOL) display at the end of the line the  
appropriate ASCII string !  
Default is TRUE.  
Applicability is (ISGNU).

BDTA\_DisplayWrap -- (BOOL) the BDTA\_BytesPerLine are ignored and the  
byte number is retrieved from the object width !  
Default is TRUE.  
Applicability is (ISGNU).

BDTA\_Found -- (STRPTR) pointer to the buffer to highlight the line.  
This is used to display the line of a found string !  
Default is NULL  
Applicability is (ISNU).

#### BUGS

At the moment proportional fonts can't be handled.

#### SEE ALSO

datatypesclass (where ?)