

binary_dtc.doc

COLLABORATORS

	TITLE : binary_dtc.doc		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		December 7, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	binary_dtc.doc	1
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 asynchronly !
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 BDTH_DisplayHex is BDTH_WORD it must be a multiply of 2,
if it is BDTH_LONG it must be a multiply of 4 !
Default is 32.
Applicability is (ISGNU).

BDTH_DisplayHex -- (UWORD) type of the display. The following types
are supported : BDTH_NONE - displays no hex values
BDTH_BYTE - displays each byte in hex (8 bit)
BDTH_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 ?)