

Gadget

COLLABORATORS

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

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Gadget	1
1.1	Gadget V1.52	1
1.2	ninitgadget	3
1.3	nbuttongadget	3
1.4	ncheckboxgadget	3
1.5	nintegergadget	4
1.6	nlistviewgadget	4
1.7	nnumbergadget	5
1.8	ncyclegadget	6
1.9	noptiongadget	6
1.10	npalettegadget	7
1.11	nscrollergadget	8
1.12	nslidergadget	8
1.13	nstringgadget	9
1.14	nsetgadgetfont	10
1.15	nsetgadgetflags	10
1.16	nusegadgetlist	10
1.17	ncreategadgetlist	11
1.18	nattachgadgetlist	11
1.19	ndisablegadget	11
1.20	nactivategadget	11
1.21	nrefreshgadget	12
1.22	nrefreshgadgetlist	12
1.23	nnogadgetborder	12
1.24	nfreegadgetlist	12
1.25	nsetstringtext	12
1.26	ngetstringtext	13
1.27	ngadgetattrs	13

Chapter 1

Gadget

1.1 Gadget V1.52

Gadget V1.52 General Information:

- * Blitz Basic II library number : #179
- * Library size when linked to executable: 1784 bytes
- * Number of commands : 26
- * Ressources automatically freed at end : Yes

NInitGadget() must be put before any other Gadget functions or you will enjoy BIG crashes.

Commands summary:

NAttachGadgetList
Statement

NActivateGadget
Statement

NButtonGadget
Command (long)

NCheckBoxGadget
Command (long)

NCreateGadgetList
Function (long)

NCycleGadget
Command (long)

NDisableGadget
Statement

NGadgetAttrs
Statement

NGetStringText

Statement

NFreeGadgetList
Statement

NInitGadget
Command (long)

NIntegerGadget
Command (long)

NListViewGadget
Command (long)

NNumberGadget
Command (long)

NNoGadgetBorder
Statement

NOptionGadget
Command (long)

NPaletteGadget
Command (long)

NRefreshGadget
Statement

NRefreshGadgetList
Statement

NScrollerGadget
Command (long)

NSetGadgetFlags
Statement

NSetGadgetFont
Statement

NSetStringText
Statement

NSliderGadget
Command (long)

NStringGadget
Command (long)

NUseGadgetList
Statement

1.2 ninitgadget

SYNTAX

```
NInitGadget #GadgetLists
```

COMMAND

Try to open the gadtools.library and initialize the gadget environment. You must put this command before any use of the new gadget set.

1.3 nbuttongadget

SYNTAX

```
NButtonGadget #Gadget, x, y, Width, Height, Text$, TagList
```

COMMAND

Create a button gadget in the actual GadgetList. #Gadget will be the number returned by NEventID command. This command can return the Intuition gadget pointer for advanced programmers.

Available Tags:

```
#GA_Disabled (BOOL) - Set to TRUE to disable gadget, FALSE otherwise  
                      (defaults to FALSE). (V36)  
#GA_Immediate (BOOL) - Hear #IDCMP_GADGETDOWN events from button gadget  
                      (defaults to FALSE). (V39)
```

1.4 ncheckboxgadget

SYNTAX

```
NCheckBoxGadget #Gadget, x, y, Width, Height, Text$, TagList
```

COMMAND

Create a checkbox gadget in the actual GadgetList. #Gadget will be the number returned by NEventID command. This command can return the Intuition gadget pointer for advanced programmers.

Available Tags:

```
#GA_Disabled (BOOL) - Set to TRUE to disable gadget, FALSE otherwise  
                      (defaults to FALSE).  
#GTCB_Checked (BOOL) - Initial state of checkbox (defaults to FALSE)  
                      (V36)  
#GTCB_Scaled (BOOL) - If true, then checkbox imagery will be scaled to  
                      fit the gadget's width & height. Otherwise, a fixed size of  
                      CHECKBOXWIDTH by CHECKBOXHEIGHT will be used. (defaults to FALSE)  
                      (V39)
```

1.5 nintegergadget

SYNTAX

```
NIntegerGadget #Gadget, x, y, Width, Height, Text$, TagList
```

COMMAND

Create a Integer gadget in the actual GadgetList. #Gadget will be the number returned by NEventID command. This command can return the Intuition gadget pointer for advanced programmers.

Available Tags:

- #GA_Disabled (BOOL) - Set to TRUE to disable gadget, FALSE otherwise (defaults to FALSE). (V36)
- #GA_Immediate (BOOL) - Hear #IDCMP_GADGETDOWN events from integer gadget (defaults to FALSE). (V39)
- #GA_TabCycle (BOOL) - Set to TRUE so that pressing <TAB> or <Shift-TAB> will activate the next or previous such gadget. (defaults to TRUE, unlike regular Intuition string gadgets which default to FALSE). (V37)
- #GTIN_Number (LONG) - The initial contents of the integer gadget (defaults to 0). (V36)
- #GTIN_MaxChars (UWORD) - The maximum number of digits that the integer gadget is to hold (defaults to 10). (V36)
- #GTIN_EditHook (struct Hook *) - Hook to use as a custom integer gadget edit hook (StringExtend->EditHook) for this gadget. GadTools will allocate the StringExtend->WorkBuffer for you. (defaults to NULL). (V37)
- #STRINGA_ExitHelp (BOOL) - Set to TRUE to have the help-key cause an exit from the integer gadget. You will then receive an #IDCMP_GADGETUP event with Code = 0x5F (rawkey for help). (defaults to FALSE) (V37)
- #STRINGA_Justification - Controls the justification of the contents of an integer gadget. Choose one of STRINGLEFT, STRINGRIGHT, or STRINGCENTER (defaults to STRINGLEFT). (V37)
- #STRINGA_ReplaceMode (BOOL) - If TRUE, this integer gadget is in replace-mode (defaults to FALSE (insert-mode)). (V37)

1.6 nlistviewgadget

SYNTAX

```
NListViewGadget #Gadget, x, y, Width, Height, Text$, TagList
```

COMMAND

Create a ListView gadget in the actual GadgetList. #Gadget will be the number returned by NEventID command. This command can return the Intuition gadget pointer for advanced programmers.

Available Tags:

- #GA_Disabled (BOOL) - Set to TRUE to disable gadget, FALSE otherwise (defaults to FALSE). (V39)
-

#GTLV_Top (WORD) - Top item visible in the listview. This value will be made reasonable if out-of-range (defaults to 0). (V36)

#GTLV_MakeVisible (WORD) - Number of an item that should be forced within the visible area of the listview by doing minimal scrolling. This tag overrides #GTLV_Top. (V39)

#GTLV_Labels (struct List *) - List of nodes whose ln_Name fields are to be displayed in the listview. (V36)

#GTLV_ReadOnly (BOOL) - If TRUE, then listview is read-only (defaults to FALSE). (V36)

#GTLV_ScrollWidth (UWORD) - Width of scroll bar for listview. Must be greater than zero (defaults to 16). (V36)

#GTLV_ShowSelected (struct Gadget *) - NULL to have the currently selected item displayed beneath the listview under V37 or with a highlight bar in V39. If not NULL, this is a pointer to an already-created GadTools #STRING_KIND gadget to have an editable display of the currently selected item. If the tag is not present, the currently selected item will not be displayed. (V36)

#GTLV_Selected (UWORD) - Ordinal number of currently selected item, or ~0 to have no current selection (defaults to ~0). (V36)

#LAYOUTA_Spacing (UWORD) - Extra space to place between lines of listview (defaults to 0). (V36)

#GTLV_ItemHeight (UWORD) - The exact height of an item. This is normally useful for listviews that use the #GTLV_Callback rendering hook (defaults to ng->ng_TextAttr->ta_YSize). (V39)

#GTLV_Callback (struct Hook *) - Callback hook for various listview operations. As of V39, the only callback supported is for custom rendering of individual items in the listview. The call back hook is called with:

- A0 - struct Hook *
- A1 - struct LVDrawMsg *
- A2 - struct Node *

The callback hook *must* check the lvdm_MethodID field of the message and only do processing if it equals LV_DRAW. If any other value is passed, the callback hook must return LVCB_UNKNOWN

#GTLV_MaxPen (UWORD) - The maximum pen number used by rendering in a custom rendering callback hook. This is used to optimize the rendering and scrolling of the listview display (default is the maximum pen number used by all of TEXTPEN, BACKGROUNDPEN, FILLPEN, TEXTFILLPEN, and BLOCKPEN. (V39)

1.7 nnumbergadget

SYNTAX

NNumberGadget #Gadget, x, y, Width, Height, Text\$, TagList

COMMAND

Create a Number gadget in the actual GadgetList. #Gadget will be the number returned by NEventID command. This command can return the Intuition gadget pointer for advanced programmers.

Available Tags:

#GTNM_Number (LONG) - A signed long integer to be displayed as a read-only

number (defaults to 0). (V36)

#GTNM_Border (BOOL) - If TRUE, this flag asks for a recessed border to be placed around the gadget. (V36)

#GTNM_FrontPen (UBYTE) - The pen to use when rendering the number (defaults to DrawInfo->dri_Pens[TEXTPEN]). (V39)

#GTNM_BackPen (UBYTE) - The pen to use when rendering the background of the number (defaults to leaving the background untouched). (V39)

#GTNM_Justification (UBYTE) - Determines how the number is rendered within the gadget box. GTJ_LEFT will make the rendering be flush with the left side of the gadget, GTJ_RIGHT will make it flush with the right side, and GTJ_CENTER will center the number within the gadget box. Under V39, using this tag also required using {#GTNM_Clipped, TRUE}, otherwise the text would not show up in the gadget. This has been fixed in V40. (defaults to GTJ_LEFT). (V39)

#GTNM_Format (STRPTR) - C-Style formatting string to apply on the number before display. Be sure to use the 'l' (long) modifier. This string is processed using exec.library/RawDoFmt(), so refer to that function for details. (defaults to "%ld") (V39)

#GTNM_MaxNumberLen (ULONG) - Maximum number of bytes that can be generated by applying the #GTNM_Format formatting string to the number (excluding the NULL terminator). (defaults to 10). (V39)

#GTNM_Clipped (BOOL) - Determine whether text should be clipped to the gadget dimensions (defaults to FALSE for gadgets without borders, TRUE for gadgets with borders). (V39)

1.8 ncyclegadget

SYNTAX

NCycleGadget #Gadget, x, y, Width, Height, Text\$, TagList

COMMAND

Create a Cycle gadget in the actual GadgetList. #Gadget will be the number returned by NEventID command. This command can return the Intuition gadget pointer for advanced programmers.

Available Tags:

#GA_Disabled (BOOL) - Set to TRUE to disable gadget, FALSE otherwise (defaults to FALSE). (V37)

#GTCY_Labels (STRPTR *) - Pointer to NULL-terminated array of strings that are the choices offered by the cycle gadget. This tag is required. (V36)

#GTCY_Active (UWORD) - The ordinal number (counting from zero) of the initially active choice of a cycle gadget (defaults to zero). (V36)

1.9 noptiongadget

SYNTAX

NOptionGadget #Gadget, x, y, Width, Height, Text\$, TagList

COMMAND

Create a exclusive option gadget in the actual GadgetList. #Gadget will be the number returned by NEventID command. This command can return the Intuition gadget pointer for advanced programmers.

Available Tags:

- #GA_Disabled (BOOL) - Set to TRUE to disable gadget, FALSE otherwise (defaults to FALSE). (V39)
- #GTMX_Labels (STRPTR *) - Pointer to a NULL-terminated array of strings which are to be the labels beside each choice in a set of mutually exclusive gadgets. This tag is required. (V36)
- #GTMX_Active (UWORD) - The ordinal number (counting from zero) of the initially active choice of an mx gadget (defaults to 0). (V36)
- #GTMX_Spacing (UWORD) - The amount of space between each choice of a set of mutually exclusive gadgets. This amount is added to the font height to produce the vertical shift between choices (defaults to 1). (V36)
- #GTMX_Scaled (BOOL) - If true, then mx gadget imagery will be scaled to fit the gadget's width & height. Otherwise, a fixed size of MXWIDTH by MXHEIGHT will be used. When setting this tag to TRUE, you should typically set the height of the gadget to be (ng.ng_TextAttr->ta_YSize + 1). (defaults to FALSE.) (V39)
- #GTMX_TitlePlace - One of PLACETEXT_LEFT, PLACETEXT_RIGHT, PLACETEXT_ABOVE, or PLACETEXT_BELOW, indicating where the title of the gadget is to be displayed. Without this tag, the NewGadget.ng_GadgetText field is ignored for MX_KIND gadgets. (V39)

1.10 npalettegadget

SYNTAX

NPaletteGadget #Gadget, x, y, Width, Height, Text\$, TagList

COMMAND

Create a Palette gadget in the actual GadgetList. #Gadget will be the number returned by NEventID command. This command can return the Intuition gadget pointer for advanced programmers.

Available Tags:

- #GA_Disabled (BOOL) - Set to TRUE to disable gadget, FALSE otherwise (defaults to FALSE). (V36)
- #GTPA_Depth (UWORD) - Number of bitplanes in the palette (defaults to 1). (V36)
- #GTPA_Color (UBYTE) - Initially selected color of the palette. This number is a pen number, and not the ordinal color number within the palette gadget itself. (defaults to 1). (V36)
- #GTPA_ColorOffset (UBYTE) - First color to use in palette

(defaults to 0). (V36)

#GTPA_IndicatorWidth (UWORD) - The desired width of the current-color indicator, if you want one to the left of the palette. (V36)

#GTPA_IndicatorHeight (UWORD) - The desired height of the current-color indicator, if you want one above the palette. (V36)

#GTPA_ColorTable (UBYTE *) - Pointer to a table of pen numbers indicating which colors should be used and edited by the palette gadget. This array must contain as many entries as there are colors displayed in the palette gadget. The array provided with this tag must remain valid for the life of the gadget or until a new table is provided. (default is NULL, which causes a 1-to-1 mapping of pen numbers). (V39)

#GTPA_NumColors (UWORD) - Number of colors to display in the palette gadget. This override #GTPA_Depth and allows numbers which aren't powers of 2. (defaults to 2) (V39)

1.11 nscrollergadget

SYNTAX

NScrollerGadget #Gadget, x, y, Width, Height, Text\$, TagList

COMMAND

Create a Scroller gadget in the actual GadgetList. #Gadget will be the number returned by NEventID command. This command can return the Intuition gadget pointer for advanced programmers.

Available Tags:

#GA_Disabled (BOOL) - Set to TRUE to disable gadget, FALSE otherwise (defaults to FALSE). (V36)

#GA_RelVerify (BOOL) - Hear every #IDCMP_GADGETUP event from scroller (defaults to FALSE). (V36)

#GA_Immediate (BOOL) - Hear every #IDCMP_GADGETDOWN event from scroller (defaults to FALSE). (V36)

#GTSC_Top (WORD) - Top visible in area scroller represents (defaults to 0). (V36)

#GTSC_Total (WORD) - Total in area scroller represents (defaults to 0). (V36)

#GTSC_Visible (WORD) - Number visible in scroller (defaults to 2). (V36)

#GTSC_Arrows (UWORD) - Asks for arrows to be attached to the scroller. The value supplied will be taken as the width of each arrow button for a horizontal scroller, or the height of each button for a vertical scroller (the other dimension will match the whole scroller). (V36)

#PGA_Freedom - Whether scroller is horizontal or vertical. Choose LORIENT_VERT or LORIENT_HORIZ (defaults to LORIENT_HORIZ). (V36)

1.12 nslidergadget

SYNTAX

NSliderGadget #Gadget, x, y, Width, Height, Text\$, TagList

COMMAND

Create a Slider gadget in the actual GadgetList. #Gadget will be the number returned by NEventID command. This command can return the Intuition gadget pointer for advanced programmers.

Available Tags:

#GA_Disabled (BOOL) - Set to TRUE to disable gadget, FALSE otherwise (defaults to FALSE). (V36)

#GA_RelVerify (BOOL) - If you want to hear each slider #IDCMP_GADGETUP event (defaults to FALSE). (V36)

#GA_Immediate (BOOL) - If you want to hear each slider #IDCMP_GADGETDOWN event (defaults to FALSE). (V36)

#GTSL_Min (WORD) - Minimum level for slider (defaults to 0). (V36)

#GTSL_Max (WORD) - Maximum level for slider (defaults to 15). (V36)

#GTSL_Level (WORD) - Current level of slider (defaults to 0). (V36)

#GTSL_MaxLevelLen (UWORD) - Maximum length in characters of level string when rendered beside slider (defaults to 2). (V36)

#GTSL_LevelFormat (STRPTR) - C-Style formatting string for slider level. Be sure to use the 'l' (long) modifier. This string is processed using exec.library/RawDoFmt(), so refer to that function for details. (defaults to "%ld"). (V36)

#GTSL_LevelPlace - One of PLACETEXT_LEFT, PLACETEXT_RIGHT, PLACETEXT_ABOVE, or PLACETEXT_BELOW, indicating where the level indicator is to go relative to slider (default to PLACETEXT_LEFT). (V36)

#GTSL_DispFunc (LONG (*function)(struct Gadget *, WORD)) - Function to calculate level to be displayed. A number-of-colors slider might want to set the slider up to think depth, and have a (1 << n) function here. Defaults to none. Your function must take a pointer to gadget as the first parameter, the level (a WORD) as the second, and return the result as a LONG. (V36)

#GTSL_MaxPixelLen (ULONG) - Indicates the maximum pixel size used up by the level display for any value of the slider. This is mostly useful when dealing with proportional fonts. (defaults to FontWidth*MaxLevelLen). (V39)

#GTSL_Justification (UBYTE) - Determines how the level display is to be justified within its allotted space. Choose one of GTJ_LEFT, GTJ_RIGHT, or GTJ_CENTER (defaults to GTJ_LEFT). (V39)

#PGA_Freedom - Set to LORIENT_VERT or LORIENT_HORIZ to have a vertical or horizontal slider (defaults to LORIENT_HORIZ). (V36)

1.13 nstringgadget

SYNTAX

NStringGadget #Gadget, x, y, Width, Height, Text\$, TagList

COMMAND

Create a String gadget in the actual GadgetList. #Gadget will be the

number returned by NEventID command. This command can return the Intuition gadget pointer for advanced programmers.

Available Tags:

#GA_Disabled (BOOL) - Set to TRUE to disable gadget, FALSE otherwise (defaults to FALSE). (V36)

#GA_Immediate (BOOL) - Hear #IDCMP_GADGETDOWN events from string gadget (defaults to FALSE). (V39)

#GA_TabCycle (BOOL) - Set to TRUE so that pressing <TAB> or <Shift-TAB> will activate the next or previous such gadget. (defaults to TRUE, unlike regular Intuition string gadgets which default to FALSE). (V37)

#GTST_String (STRPTR) - The initial contents of the string gadget, or NULL (default) if string is to start empty. (V36)

#GTST_MaxChars (UWORD) - The maximum number of characters that the string gadget is to hold. (V36)

#GTST_EditHook (struct Hook *) - Hook to use as a custom string gadget edit hook (StringExtend->EditHook) for this gadget. GadTools will allocate the StringExtend->WorkBuffer for you. (defaults to NULL). (V37)

#STRINGA_ExitHelp (BOOL) - Set to TRUE to have the help-key cause an exit from the string gadget. You will then receive an #IDCMP_GADGETUP event with Code = 0x5F (rawkey for help). (V37)

#STRINGA_Justification - Controls the justification of the contents of a string gadget. Choose one of #STRINGLEFT, #STRINGRIGHT, or #STRINGCENTER (defaults to #STRINGLEFT). (V37)

#STRINGA_ReplaceMode (BOOL) - If TRUE, this string gadget is in replace-mode (defaults to FALSE (insert-mode)). (V37)

1.14 nsetgadgetfont

SYNTAX

NSetGadgetFont &TextAttr

STATEMENT

Set the font which will be use by the next created gadgets.
It must be an TextAttr structure, so be careful when using it...

1.15 nsetgadgetflags

SYNTAX

NSetGadgetFlags &Text\$

STATEMENT

Set the flags for the next created gadgets.

1.16 nusegadgetlist

SYNTAX

```
NUseGadgetList #GadgetList
```

STATEMENT

Set the specified Gadgetlist as used.

1.17 ncreategadgetlist

SYNTAX

```
result.l = NCreateGadgetList(#GadgetList, ScreenID)
```

FUNCTION

Try to allocate the ressource to futur gadgetlist. The ScreenID is needed, so be sur to pass it !

1.18 nattachgadgetlist

SYNTAX

```
NAttachGadgetList #GadgetList, WindowID
```

STATEMENT

Attach the specified gadgetlist to an opened window specified by the WindowID. Gadgets are automatically refreshed.

1.19 ndisablegadget

SYNTAX

```
NDisableGadget #Gadget, State
```

STATEMENT

Disable or enable a gadget. If State = 1, gadget will be disable, if State = 0 will be enabled.

NOTE: You must use the NRefreshGadget to reflect the changes on the display.

1.20 nactivategadget

SYNTAX

```
NActivateGadget #Gadget
```

STATEMENT

Cause the specified gadget to be activated. Useful for StringGadget.

1.21 nrefreshgadget

SYNTAX

```
NRefreshGadget #Gadget
```

STATEMENT

The specified gadget display will be refreshed.

1.22 nrefreshgadgetlist

SYNTAX

```
NRefreshGadgetList
```

STATEMENT

Refresh the current gadgetlist: all the gadgets will be redraw on the window which contain the gadgetlist.

1.23 nnogadgetborder

SYNTAX

```
NNoGadgetBorder #Gadget
```

STATEMENT

Must be put after a gadget declaration and will remove the border around the specified gadget

```
ie: NButtonGadget 1, 10,10,100,100,"Hello",0  
    NNoGadgetBorder 1
```

1.24 nfreegadgetlist

SYNTAX

```
NFreeGadgetList #GadgetList
```

STATEMENT

Free the memory taken by the specified gadgetlist. Be sure than this gadget list is no more referenced by any windows or you will have a crash !
Call it typically after a window close.

1.25 nsetstringtext

SYNTAX

```
NSetStringText #Gadget, Text$
```

STATEMENT

Change the text content of a string gadget.

1.26 ngetstringtext

SYNTAX

```
Text$ = NGetStringText (#Gadget)
```

STATEMENT

Return the text content of a string gadget.

1.27 ngadgetattrs

SYNTAX

```
NGadgetAttrs #Gadget, #TAG_ITEM, #TAG_DATA
```

STATEMENT

Change the attributs of the given gadget.
