

Gadget

COLLABORATORS

	<i>TITLE :</i> Gadget		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 23, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Gadget	1
1.1	Gadget V1.00	1
1.2	initgadget	2
1.3	buttongadget	2
1.4	checkboxgadget	3
1.5	integergadget	4
1.6	listviewgadget	5
1.7	numbergadget	6
1.8	cyclegadget	7
1.9	optiongadget	8
1.10	palettegadget	9
1.11	scrollergadget	10
1.12	slidergadget	11
1.13	stringgadget	12
1.14	textgadget	13
1.15	setgadgetfont	14
1.16	setgadgetflags	14
1.17	usegadgetlist	15
1.18	creategadgetlist	15
1.19	attachgadgetlist	15
1.20	disablegadget	16
1.21	activategadget	16
1.22	refreshgadget	16
1.23	refreshgadgetlist	16
1.24	nogadgetborder	17
1.25	freegadgetlist	17
1.26	setstringtext	17
1.27	getstringtext	17
1.28	setgadgetattrs	18

Chapter 1

Gadget

1.1 Gadget V1.00

Pure Basic Gadget library V1.00

Les 'Gadgets' sont les éléments qui composent une interface graphique et qui permet à l'utilisateur d'inter-agir avec elle. Ils sont composés de divers éléments comme les boutons, les listes d'éléments, les cases à cocher, etc... La bibliothèque PureBasic met à votre disposition un nombre important de gadgets basés sur le jeu officiel de gadgets sur AmigaOS: GadTools. Ces gadgets sont rapides, consomment très peu de ressources et permettent de construire rapidement des interfaces solides et agréables.

Résumées disponibles:

- AttachGadgetList
- ActivateGadget
- ButtonGadget
- CheckBoxGadget
- CreateGadgetList
- CycleGadget
- DisableGadget
- GetStringText
- FreeGadgetList
- InitGadget
- IntegerGadget
- ListViewGadget
- NumberGadget
- NoGadgetBorder
- OptionGadget
- PaletteGadget
- RefreshGadget
- RefreshGadgetList
- ScrollerGadget
- SetGadgetAttrs
- SetGadgetFlags
- SetGadgetFont
- SetStringText
- SliderGadget
- StringGadget

```
TextGadget
UseGadgetList
```

Exemples :

```
Program 1
Wild Manager
```

1.2 initgadget

Syntaxe

```
Résultat = InitGadget(#GadgetLists)
```

Résumé

Initialise l'environnement nécessaire à la gestion des gadgets. Cette fonction essaye d'ouvrir la 'gadtools.library' et retourne la valeur NULLE si une erreur se produit lors de cette initialisation (manque de mémoire, 'gadtools.library' introuvable). Vous devez appeler cette Résuméée avant d'utiliser les autres fonctions de cette bibliothèque.

#GadgetLists: Nombre maximum de listes de gadgets à gérer. Chaque liste de ↵
gadgets
peut contenir un nombre infini de gadgets.

1.3 buttongadget

Syntaxe

```
ButtonGadget(#Gadget, x, y, Width, Height, Text$, TagList)
```

Résumé

Ajoute un gadget de type 'Bouton poussoir' à la liste de gadgets en cours de construction.

Pour les programmeur expérimentés: Cette fonction retourne le pointeur du gadget créé (structure 'Gadget' des RKM).

#Gadget: Identifiant numérique du gadget qui va être créé.

x,y: Coordonnées du coin supérieur gauche du gadget par rapport au coin ↵
supérieur
gauche de la fenêtre.

Width: Largeur du gadget

Height: Hauteur du gadget

Text\$: Label qui sera affiché dans le gadget.

TagList: Liste de tag supplémentaire pour affiner le réglage du gadget (voir ci ↵
dessous

pour les tags disponibles).

Tags disponibles:

#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 checkboxgadget

Syntaxe

CheckBoxGadget(#Gadget, x, y, Width, Height, Text\$, TagList)

Résumé

Ajoute un gadget de type 'Case à cocher' à la liste de gadgets en cours de construction.

Pour les programmeur expérimentés: Cette fonction retourne le pointeur du gadget créé (structure 'Gadget' des RKM).

#Gadget: Identifiant numérique du gadget qui va être créé.

x,y: Coordonnées du coin supérieur gauche du gadget par rapport au coin ←
 supérieur
 gauche de la fenêtre.

Width: Largeur du gadget

Height: Hauteur du gadget

Text\$: Label qui sera affiché dans le gadget.

TagList: Liste de tag supplémentaire pour affiner le réglage du gadget (voir ci ←
 dessous
 pour les tags disponibles).

Tags disponibles:

#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 integergadget

Syntaxe

```
IntegerGadget(#Gadget, x, y, Width, Height, Text$, TagList)
```

Résumé

Ajoute un gadget de type 'Saisie de nombre' à la liste de gadgets en cours de construction.

Pour les programmeur expérimentés: Cette fonction retourne le pointeur du gadget créé (structure 'Gadget' des RKM).

#Gadget: Identifiant numérique du gadget qui va être créé.

x,y: Coordonnées du coin supérieur gauche du gadget par rapport au coin ←
supérieur
gauche de la fenêtre.

Width: Largeur du gadget

Height: Hauteur du gadget

Text\$: Label qui sera affiché dans le gadget.

TagList: Liste de tag supplémentaire pour affiner le réglage du gadget (voir ci ←
dessous
pour les tags disponibles).

Tags disponibles:

```
#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 listviewgadget

Syntaxe

ListViewGadget(#Gadget, x, y, Width, Height, Text\$, TagList)

Résumé

Ajoute un gadget de type 'Liste d'éléments' à la liste de gadgets en cours de construction.

Pour les programmeur expérimentés: Cette fonction retourne le pointeur du gadget créé (structure 'Gadget' des RKM).

#Gadget: Identifiant numérique du gadget qui va être créé.

x,y: Coordonnées du coin supérieur gauche du gadget par rapport au coin ←
supérieur
gauche de la fenêtre.

Width: Largeur du gadget

Height: Hauteur du gadget

Text\$: Label qui sera affiché dans le gadget.

TagList: Liste de tag supplémentaire pour affiner le réglage du gadget (voir ci ←
dessous
pour les tags disponibles).

Tags disponibles:

#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 numbergadget

Syntaxe

NumberGadget(#Gadget, x, y, Width, Height, Text\$, TagList)

Résumé

Ajoute un gadget de type 'Affichage de nombre' à la liste de gadgets en cours de construction.

Pour les programmeur expérimentés: Cette fonction retourne le pointeur du gadget créé (structure 'Gadget' des RKM).

#Gadget: Identifiant numérique du gadget qui va être créé.

x,y: Coordonnées du coin supérieur gauche du gadget par rapport au coin ←
supérieur
gauche de la fenêtre.

Width: Largeur du gadget

Height: Hauteur du gadget

Text\$: Label qui sera affiché dans le gadget.

TagList: Liste de tag supplémentaire pour affiner le réglage du gadget (voir ci ←
dessous
pour les tags disponibles).

Tags disponibles:

#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_Clippped, 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 Résumé 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_Clippped (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 cyclegadget

Syntaxe

CycleGadget(#Gadget, x, y, Width, Height, Text\$, TagList)

Résumé

Ajoute un gadget de type 'Bouton à choix multiple' à la liste de gadgets en cours de construction.

Pour les programmeur expérimentés: Cette fonction retourne le pointeur du gadget créé (structure 'Gadget' des RKM).

#Gadget: Identifiant numérique du gadget qui va être créé.

x,y: Coordonnées du coin supérieur gauche du gadget par rapport au coin ←
supérieur
gauche de la fenêtre.

Width: Largeur du gadget

Height: Hauteur du gadget

Text\$: Label qui sera affiché dans le gadget.

TagList: Liste de tag supplémentaire pour affiner le réglage du gadget (voir ci ←
dessous
pour les tags disponibles).

Tags disponibles:

#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 optiongadget

Syntaxe

OptionGadget(#Gadget, x, y, Width, Height, Text\$, TagList)

Résumé

Ajoute un gadget de type 'Options exclusives' à la liste de gadgets en cours de construction.

Pour les programmeur expérimentés: Cette fonction retourne le pointeur du gadget créé (structure 'Gadget' des RKM).

#Gadget: Identifiant numérique du gadget qui va être créé.

x,y: Coordonnées du coin supérieur gauche du gadget par rapport au coin ←
 supérieur
 gauche de la fenêtre.

Width: Largeur du gadget

Height: Hauteur du gadget

Text\$: Label qui sera affiché dans le gadget.

TagList: Liste de tag supplémentaire pour affiner le réglage du gadget (voir ci ←
 dessous
 pour les tags disponibles).

Tags disponibles:

#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 palettegadget

Syntaxe

PaletteGadget (#Gadget, x, y, Width, Height, Text\$, TagList)

Résumé

Ajoute un gadget de type 'Palette de couleurs' à la liste de gadgets en cours de construction.

Pour les programmeur expérimentés: Cette fonction retourne le pointeur du gadget créé (structure 'Gadget' des RKM).

#Gadget: Identifiant numérique du gadget qui va être créé.

x,y: Coordonnées du coin supérieur gauche du gadget par rapport au coin ←
supérieur
gauche de la fenêtre.

Width: Largeur du gadget

Height: Hauteur du gadget

Text\$: Label qui sera affiché dans le gadget.

TagList: Liste de tag supplémentaire pour affiner le réglage du gadget (voir ci ←
dessous
pour les tags disponibles).

Tags disponibles:

#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 colour number within the palette gadget itself. (defaults to 1). (V36)

#GTPA_ColorOffset (UBYTE) - First colour to use in palette (defaults to 0). (V36)

#GTPA_IndicatorWidth (UWORD) - The desired width of the current-colour 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 scrollergadget

Syntaxe

ScrollerGadget(#Gadget, x, y, Width, Height, Text\$, TagList)

Résumé

Ajoute un gadget de type 'Ascenseur' à la liste de gadgets en cours de construction.

Pour les programmeur expérimentés: Cette fonction retourne le pointeur du gadget créé (structure 'Gadget' des RKM).

#Gadget: Identifiant numérique du gadget qui va être créé.

x,y: Coordonnées du coin supérieur gauche du gadget par rapport au coin ←
supérieur
gauche de la fenêtre.

Width: Largeur du gadget

Height: Hauteur du gadget

Text\$: Label qui sera affiché dans le gadget.

TagList: Liste de tag supplémentaire pour affiner le réglage du gadget (voir ci ←
dessous
pour les tags disponibles).

Tags disponibles:

#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 slidergadget

Syntaxe

SliderGadget(#Gadget, x, y, Width, Height, Text\$, TagList)

Résumé

Ajoute un gadget de type 'Ascenseur' à la liste de gadgets en cours de construction.

Pour les programmeur expérimentés: Cette fonction retourne le pointeur du gadget créé (structure 'Gadget' des RKM).

#Gadget: Identifiant numérique du gadget qui va être créé.

x,y: Coordonnées du coin supérieur gauche du gadget par rapport au coin ←
 supérieur
 gauche de la fenêtre.

Width: Largeur du gadget

Height: Hauteur du gadget

Text\$: Label qui sera affiché dans le gadget.

TagList: Liste de tag supplémentaire pour affiner le réglage du gadget (voir ci ←
 dessous
 pour les tags disponibles).

Tags disponibles:

#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 Résumé 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_DisFunc (LONG (*Résumé)(struct Gadget *, WORD)) - Résumé 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) Résumé here. Defaults to none. Your Résumé 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_MaxPixellLen (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 stringgadget

Syntaxe

`StringGadget(#Gadget, x, y, Width, Height, Text$, TagList)`

Résumé

Ajoute un gadget de type 'Saisie de texte' à la liste de gadgets en cours de construction.

Pour les programmeur expérimentés: Cette fonction retourne le pointeur du gadget créé (structure 'Gadget' des RKM).

#Gadget: Identifiant numérique du gadget qui va être créé.

x,y: Coordonnées du coin supérieur gauche du gadget par rapport au coin ←
supérieur
gauche de la fenêtre.

Width: Largeur du gadget

Height: Hauteur du gadget

Text\$: Label qui sera affiché dans le gadget.

TagList: Liste de tag supplémentaire pour affiner le réglage du gadget (voir ci ←
dessous
pour les tags disponibles).

Tags disponibles:

#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 textgadget

Syntaxe

TextGadget(#Gadget, x, y, Width, Height, Text\$, TagList)

Résumé

Ajoute un gadget de type 'Affichage de texte' à la liste de gadgets en cours de construction.

Pour les programmeur expérimentés: Cette fonction retourne le pointeur du gadget créé (structure 'Gadget' des RKM).

#Gadget: Identifiant numérique du gadget qui va être créé.

x,y: Coordonnées du coin supérieur gauche du gadget par rapport au coin ←
supérieur
gauche de la fenêtre.

Width: Largeur du gadget

Height: Hauteur du gadget

Text\$: Label qui sera affiché dans le gadget.

TagList: Liste de tag supplémentaire pour affiner le réglage du gadget (voir ci ←
dessous
pour les tags disponibles).

Tags disponibles:

- #GTTX_Text - Pointer to a NULL terminated string to be displayed, as a read-only text-display gadget, or NULL. (defaults to NULL) (V36)
- #GTTX_CopyText (BOOL) - This flag instructs the text-display gadget to copy the supplied text string, instead of using only pointer to the string. This only works for the initial value of #GTTX_Text set at CreateGadget() time. If you subsequently change #GTTX_Text, the new text will be referenced by pointer, not copied. Do not use this tag with a NULL #GTTX_Text. (V37)
- #GTTX_Border (BOOL) - If TRUE, this flag asks for a recessed border to be placed around the gadget. (V36)
- #GTTX_FrontPen (UBYTE) - The pen to use when rendering the text (defaults to DrawInfo->dri_Pens[TEXTPEN]). (V39)
- #GTTX_BackPen (UBYTE) - The pen to use when rendering the background of the text (defaults to leaving the background untouched). (V39)
- #GTTX_Justification (UBYTE) - Determines how the text 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 text within the gadget box. Under V39, using this tag also required using {#GTNM_Clippped, TRUE}, otherwise the text would not show up in the gadget. This has been fixed in V40. (defaults to #GTJ_LEFT). (V39)
- #GTTX_Clippped (BOOL) - Determine whether text should be clipped to the gadget dimensions (defaults to FALSE for gadgets without borders, TRUE for gadgets with borders). (V39)

1.15 setgadgetfont

Syntaxe

SetGadgetFont (@TextAttr)

Résumé

Change la police utilisée par les gadgets par celle spécifiée. Cette fonction est réservée aux programmeurs expérimentés car il faut utiliser une structure 'TextAttr' pour passer en paramètre.

TextAttr: Pointeur vers une structure 'TextAttr' valide.

1.16 setgadgetflags

Syntaxe

SetGadgetFlags (Flags)

Résumé

Change les 'Flags' par défaut que les gadgets utiliseront lors

de leur création.

1.17 usegadgetlist

Syntaxe

```
UseGadgetList (#GadgetList)
```

Résumé

Change la liste des gadget courante par celle spécifiée.

#GadgetList: Identifiant numérique de la liste de gadgets.

1.18 creategadgetlist

Syntaxe

```
Résultat = CreateGadgetList (#GadgetList, ScreenID)
```

Résumé

Crée une structure vide pour gérer la future liste des gadgets. Vous devez appeler cette fonction à chaque fois que vous voulez créer une nouvelle liste de gadgets.

#GadgetList: Identifiant numérique de la liste des gadgets.

ScreenID: Identifiant de l'écran sur lequel la liste des gadgets sera affichée. Pour obtenir rapidement l'identifiant de l'écran courant, vous pouvez utiliser la fonction `ScreenID()` .

1.19 attachgadgetlist

Syntaxe

```
AttachGadgetList (#GadgetList, WindowID)
```

Résumé

Attache la liste des gadgets spécifiée à une fenêtre donnée. Les gadgets sont automatiquement retracés et l'affichage mis à jour. Chaque fenêtre ne peut contenir qu'une seule liste de gadget.

#GadgetList: Identifiant numérique de la liste des gadgets à attacher.

WindowID: Identifiant de la fenêtre sur laquelle la liste des gadgets doit être attachée. Pour obtenir rapidement l'identifiant de la fenêtre courante, vous pouvez utiliser la fonction `WindowID()` .

1.20 disablegadget

Syntaxe

```
DisableGadget (#Gadget, State)
```

Résumé

Désactive ou ré-active le gadget spécifié. Vous devez utiliser la commande `RefreshGadget()` pour afficher le changement sur l'écran.

#Gadget: Identifiant numérique du gadget

State: Si State=1, alors le gadget sera désactivé, si State=0, alors le gadget sera ré-activé.

1.21 activategadget

Syntaxe

```
ActivateGadget (#Gadget)
```

Résumé

Active le gadget spécifié. Très utile pour activer un gadget de type 'String' par exemple.

#Gadget: Identifiant numérique du gadget à activer.

1.22 refreshgadget

Syntaxe

```
RefreshGadget (#Gadget)
```

Résumé

Retrace l'affichage du gadget spécifié.

#Gadget: Identifiant numérique du gadget à retracer.

1.23 refreshgadgetlist

Syntaxe

```
RefreshGadgetList ()
```

Résumé

Retrace l'affichage de tous les gadgets contenus dans la liste de gadgets courante.

1.24 nogadgetborder

Syntaxe

```
NoGadgetBorder(#Gadget)
```

Résumé

Enlève les bordures d'un gadget spécifié. Cette fonction doit être appelée immédiatement après la déclaration de ce gadget

#Gadget: Identifiant numérique du gadget

Exemple:

```
ButtonGadget(1, 10,10,100,100,"Hello",0)  
NoGadgetBorder(1)
```

1.25 freegadgetlist

Syntaxe

```
FreeGadgetList(#GadgetList)
```

Résumé

Libère la mémoire alloué par la liste de gadget spécifiée. Cette liste de gadget ne doit plus être utilisée au moment de la destruction, et elle ne pourra plus être utilisée ultérieurement.

#GadgetList: Identifiant numérique de la liste des gadgets à supprimer.

1.26 setstringtext

Syntaxe

```
SetStringText(#Gadget, Text$)
```

Résumé

Modifie le texte contenu d'un gadget de type 'String'.

#Gadget: Identifiant numérique du gadget de type 'String'.

Text\$: Nouveau texte.

1.27 getstringtext

Syntaxe

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

Résumé

Retourne le contenu d'un gadget de type 'String'

#Gadget: Identifiant numérique du gadget de type 'String'.

1.28 setgadgetattrs

Syntaxe

SetGadgetAttrs(#Gadget, #TAG_ITEM, #TAG_DATA)

Résumé

Modifie les attributs du gadget concerné.

#Gadget: Identifiant numérique du gadget à modifier.

#TAG_ITEM, #TAG_DATA: Couple de tags permettant de modifier une valeur précise du gadget. Chaque type de gadget a ses propres tags. Pour la liste complète des tags disponibles pour chaque gadget, référez-vous à la documentation de ce gadget.
