MITSUBISHI ELECTRIC

Changes for the Better

MITSUBISHI CNC

NC Designer

Function Reference Manual

IB-1500109(ENG)-C

MELDAS is a registered trademark of Mitsubishi Electric Corporation.
Other product names and company names are trademarks or registered trademarks of their respective companies.

Introduction

This manual is a guide for users of NC Designer for the Mitsubishi CNC700/70 series.
This manual describes the function groups used in NC Designer. In addition, fully understand "Precautions for Safety" on the next page to use the Mitsubishi CNC700/70 series safely.

Details described in this manual:

⚠ CAUTION

- ⚠ For items described as "Restrictions" or "Usable State" in this manual, the instruction manual issued by the machine tool builder takes precedence over this manual.
- ⚠ Items that are not described in this manual must be interpreted as "not possible".
- ⚠ This manual is written on the assumption that all option functions are added. Confirm the specifications issued by the machine tool builder before use.
- ⚠ Some screens and functions may differ depending on each NC system (or version), and some functions may not be possible. Please confirm the specifications before use.

Precautions for Safety

Always read the specifications issued by the machine tool builder, this manual, related manuals and attached documents before installation, operation, programming, maintenance or inspection to ensure correct use.
Understand this numerical controller, safety items and cautions before using the unit.
This manual ranks the safety precautions into "DANGER", "WARNING" and "CAUTION".

⚠ **DANGER**	When the user may be subject to imminent fatalities or major injuries if handling is mistaken.
⚠ **WARNING**	When the user may be subject to fatalities or major injuries if handling is mistaken.
⚠ **CAUTION**	When the user may be subject to injuries or when physical damage may occur if handling is mistaken.

Note that even items ranked as " ⚠ **CAUTION**", may lead to major results depending on the situation. In any case, important information that must always be observed is described.

⚠ DANGER

Not applicable in this manual.

⚠ WARNING

Not applicable in this manual.

⚠ CAUTION

1. Items related to product and manual

- ⚠ If the descriptions relating to the "restrictions" and "allowable conditions" conflict between this manual and the machine tool builder's instruction manual, the latter has priority over the former.
- ⚠ The operations to which no reference is made in this manual should be considered impossible.
- ⚠ This manual is complied on the assumption that your machine is provided with all optional functions. Confirm the functions available for your machine before proceeding to operation by referring to the specification issued by the machine tool builder.
- ⚠ In some NC system versions, there may be cases that different pictures appear on the screen, the machine operates in a different way on some function is not activated.

Contents

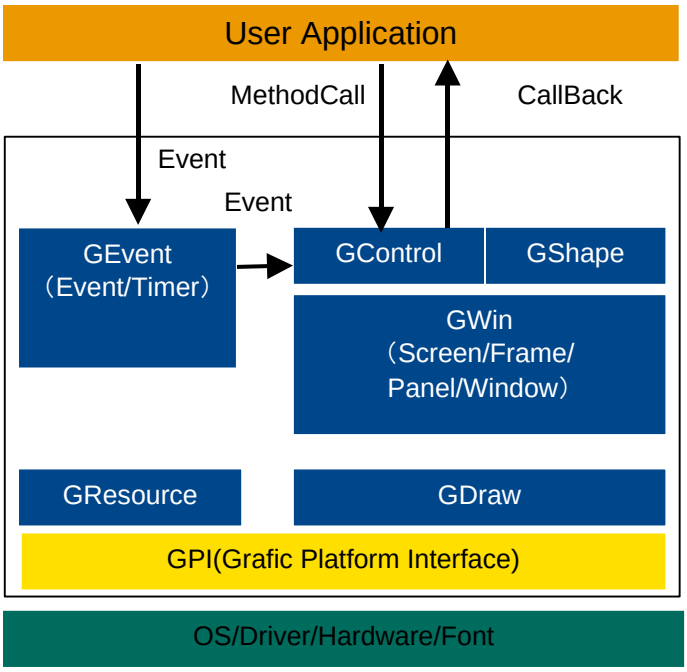
1. Outline 1
1.1 GUI Library 1
2. GDraw 3
2.1 List of GDraw Functions 3
2.2 Outlines of GDraw Functions 8
3. GResource 94
3.1 List of GResource Functions 94
3.2 Outlines of GResource Functions 95
4. GEvent 105
4.1 List of GEvent Functions 105
4.2 Outline of GEvent Functions 106
5. GCS 114
5.1 List of GCS Functions 114
5.2 Description of GCS Functions 131
6. GIO 413
6.1 List of GIP Functions 413
6.2 Outline of GIO Functions 414
7. GMMemory 418
7.1 List of GMMemory Functions 418
7.2 Outline of GMMemory Functions 419
Appendix 1. Data Type Definitions 426
Index ..... X - 1

1. Outline

This section gives an outline of the GUI library.

1.1 GUI Library

The GUI library is a C/C++ language GUI library that gives powerful support for GUI development. The GUI library is equipped with the functions necessary for the construction of not only event management but also GUI elements such as window systems, an it supports simple construction of GUIs. In addition, it is equipped with a Graphic Platform Interface (GPI), which is an arrangement for calling up processing dependent on hardware to make porting to various platforms easy. Porting the GUI library to other platforms is done by creating one with this GPI processing matched to the platform.



GUI Library Configuration

The functions for the various function groups are as follows.

Function Name	Outline
GEvent	Function group for managing various types of events such as the mouse, keyboard and timer and window system events such as GWin/GControl/GShape
GResource	Function group for handling resources such as character strings, 3D borders and images
GWin	Function group for window system management
GControl	Function group for controlling buttons, pictures and the like that operate on GWin
GShape	Function group for managing rectangles, circles, straight lines and other figures that operate on GWin
GDraw	Function group for making settings for the drawing environment such as basic figures, character drawing, coordinates, colors, etc.
GPI	Function group for handling parts that are dependent on hardware or OS

2. GDraw

This section describes the GDraw functions.

2.1 List of GDraw Functions

No.	Name	Type	File name	Outline
1	GDBeginDraw	void	GDBeginDraw.c	Begin drawing
2	GDBitBlt	void	GDBitBlt.c	Block transfer
3	GDCalcBorderRect	void	GDCalcBorderRect.c	Calculate rectangular area based on 3D border settings
4	GDCloseDrawHandle	void	GDCloseDrawHandle.c	Release drawing handle
5	GDDefineFillPattern	void	GDDefineFillPattern.c	Define fill pattern
6	GDDefinePenDash	void	GDDefinePenDash.c	Define line pattern
7	GDDrawArc	void	GDDrawArc.c	Draw arc
8	GDDrawBezier	void	GDDrawBezier.c	Draw Bezier curve
9	GDDrawCaptionCursorString	void	GDDrawCaptionCursorString.c	Draw caption character string (with cursor display)
10	GDDrawCaptionString	void	GDDrawCaptionString.c	Draw caption character string
11	GDDrawCaptionStringResource	void	GDDrawCaptionStringResource.c	Draw caption based on character string resources
12	GDDrawGrf	short	GDDrawGrf.c	Draw BMP
13	GDDrawImage	void	GDDrawImage.c	Draw image
14	GDDrawImageResource	void	GDDrawImageResource.c	Draw resource image
15	GDDrawOval	void	GDDrawOval.c	Draw oval
16	GDDrawOvalBorder	void	GDDrawOvalBorder.c	Draw (oval) circular 3D border
17	GDDrawOvalBorderResource	void	GDDrawOvalBorderResource.c	Draw (oval) circular resource 3D border
18	GDDrawPie	void	GDDrawPie.c	Draw sector
19	GDDrawPoly	void	GDDrawPoly.c	Draw polygon
20	GDDrawRect	void	GDDrawRect.c	Draw rectangle
21	GDDrawRectBorder	void	GDDrawRectBorder.c	Draw rectangular 3D border
22	GDDrawRectBorderResource	void	GDDrawRectBorderResource.c	Draw rectangular resource 3D border
23	GDDrawRRect	void	GDDrawRRect.c	Draw rounded corner rectangle
24	GDDrawString	void	GDDrawString.c	Draw character string
25	GDDrawStringResource	void	GDDrawStringResource.c	Draw resource character string
26	GDEndDraw	void	GDEndDraw.c	End drawing
27	GDFlushScreen	void	GDFlushScreen.c	Refresh screen
28	GDGetCharHeight	short	GDGetCharHeight.c	Get specified character height
29	GDGetCharWidth	short	GDGetCharWidth.c	Get specified character width

No.	Name	Type	File name	Outline
30	GDGetNextChar	char*	GDGetNextChar.c	Get next character
31	GDGetPalette	void	GDGetPalette.c	Get palette color code
32	GDGetPixel	GColor	GDGetPixel.c	Get color code for specified coordinates
33	GDGetPrevChar	char*	GDGetPrevChar.c	Get previous character
34	GDGetStringLength	unsigned long	GDGetStringLength.c	Get character string length
35	GDGetStringWidth	short	GDGetStringWidth.c	Get specified character width
36	GDInceptClipRect	short	GDInceptClipRect.c	Clip the area of overlap between the specified rectangle and the current clip area
37	GDInitPlatform	short	GDInitPlatform.c	Initialization processing for GUI library environment
38	GDLineTo	void	GDLineTo.c	Draw line
39	GDMoveTo	void	GDMoveTo.c	Move pen position
40	GDOffsetLocalOrigin	void	GDOffsetLocalOrigin.c	Offset to local coordinate origin
41	GDOpenDrawHandle	HGDRAW	GDOpenDrawHandle.c	Get drawing handle
42	GDQueryError	unsigned short	GDQueryError.c	Get final error code
43	GDQuitPlatform	short	GDQuitPlatform.c	Quit processing for GUI library environment
44	GDSetBackColor	void	GDSetBackColor.c	Set background color
45	GDSetBrush	void	GDSetBrush.c	Set brush
46	GDSetClipRect	void	GDSetClipRect.c	Define clipping area
47	GDSetDisplay	short	GDSetDisplay.c	Set graphics mode
48	GDSetDrawingMode	void	GDSetDrawingMode.c	Set raster operations for drawing
49	GDSetFillAttribute	void	GDSetFillAttribute.c	Set fill attributes
50	GDSetFillColor	void	GDSetFillColor.c	Set colors for foreground and background
51	GDSetFillPattern	void	GDSetFillPattern.c	Set fill pattern
52	GDSetFont	short	GDSetFont.c	Set font
53	GDSetForeColor	void	GDSetForeColor.c	Set foreground color
54	GDSetLocalMode	void	GDSetLocalMode.c	Set to local (physical) coordinate system
55	GDSetPenColor	void	GDSetPenColor.c	Set line and character color
56	GDSetPenDash	void	GDSetPenDash.c	Set line pattern
57	GDSetPenJoin	void	GDSetPenJoin.c	Specify joint for rectangles and polygons
58	GDSetPenSize	void	GDSetPenSize.c	Set line width (pen size)

No.	Name	Type	File name	Outline
59	GDSetPixel	void	GDSetPixel.c	Draw pixels
60	GDSetSimpleFont	HGFONT	GDSetSimpleFont.c	Set simple font
61	GDSetSimpleFontResource	HGFONT	GDSetSimpleFontResource.c	Set resource simple font
62	GDSetVirtualMode	void	GDSetVirtualMode.c	Set to virtual coordinate system
63	GDSetVirtualRect	void	GDSetVirtualRect.c	Set rectangular area in virtual coordinates
64	GDSetVram	short	GDSetVram.c	Set VRAM for drawing destination
65	GFGetFontAttribute	short	GFGetFontAttribute.c	Get font attributes
66	GFGetFontCount	short	GFGetFontCount.c	Get number of fonts that can be used
67	GFGetFontInformation	short	GFGetFontInformation.c	Get information for font of specified No.
68	GFGetFontUserAttribute	short	GFGetFontUserAttribute.c	Get user font attributes
69	GFGetSize	short	GFGetSize.c	Get font size
70	GFLoadFont	HGFONT	GFLoadFont.c	Load specified font
71	GFReleaseFont	void	GFReleaseFont.c	Release font
72	GFSetFontAttribute	short	GFSetFontAttribute.c	Set font attributes
73	GFSetFontUserAttribute	short	GFSetFontUserAttribute.c	Set user font attributes
74	GFSetSize	short	GFSetSize.c	Set character size
75	GRDuplicateRect	void	GRDuplicateRect.c	Copy rectangle
76	GREqualRect	short	GREqualRect.c	Confirm that rectangle areas are the same
77	GRGetRect	void	GRGetRect.c	Get rectangle area
78	GRInceptRect	short	GRInceptRect.c	Confirm rectangular area overlap
79	GRMapRect	void	GRMapRect.c	Enlarge/reduce rectangular area
80	GRMapToScreenRect	void	GRMapToScreenRect.c	Convert rectangular area defined with pgrRect to screen coordinate system
81	GRMapToVirtualRect	void	GRMapToVirtualRect.c	Convert rectangular area defined with pgrRect to virtual screen coordinate system
82	GROffsetRect	void	GROffsetRect.c	Rectangle offset processing
83	GRPhysicalToVirtualRect	void	GRPhysicalToVirtualRect.c	Convert rectangle physical coordinates to virtual coordinates

No.	Name	Type	File name	Outline
84	GRPointInRect	short	GRPointInRect.c	Check if point coordinates fall within rectangle coordinates
86	GRSetRect	void	GRSetRect.c	Define rectangular area
87	GRShiftRect	short	GRShiftRect.c	Rectangle shift processing
88	GRUnionRect	void	GRUnionRect.c	Join rectangle areas
89	GRVirtualToPhysicalRect	void	GRVirtualToPhysicalRect.c	Convert rectangular area virtual coordinates to physical coordinates
90	GLDuplicateLine	void	GLDuplicateLine.c	Copy line structure
91	GLGetLine	void	GLGetLine.c	Get line
92	GLMapLine	void	GLMapLine.c	Enlarge/reduce line coordinates
93	GLOffsetLine	void	GLOffsetLine.c	Offset processing for line structure
94	GLPhysicalToVirtualLine	void	GLPhysicalToVirtualLine.c	Convert line structure physical coordinates to virtual coordinates
95	GLSetLine	void	GLSetLine.c	Define line structure
96	GLVirtualToPhysicalLine	void	GLVirtualToPhysicalLine.c	Convert line structure virtual coordinates to physical coordinates
97	GPTDuplicatePoint	void	GPTDuplicatePoint.c	Copy point structure
98	GPTGetPoint	void	GPTGetPoint.c	Get point structure content
99	GPTMapPoint	void	GPTMapPoint.c	Map rectangle defined with pgrFrom to rectangle defined with pgrTo
100	GPTMapToScreenPoint	void	GPTMapToScreenPoint.c	Convert point structure to screen coordinates
101	GPTMapToVirtualPoint	void	GPTMapToVirtualPoint.c	Convert point defined in pgptPoint to virtual screen coordinate system
102	GPTOffsetPoint	void	GPTOffsetPoint.c	Point structure offset processing
103	GPTPhysicalToVirtualPoint	void	GPTPhysicalToVirtualPoint.c	Convert point structure physical coordinates to virtual coordinates
104	GPTSetPoint	void	GPTSetPoint.c	Define point structure
105	GPTVirtualToPhysicalPoint	void	GPTVirtualToPhysicalPoint.c	Convert point structure virtual coordinates to physical coordinates
106	GPGetPoly	void	GPGetPoly.c	Get polygon settings
107	GPMapPoly	void	GPMapPoly.c	Enlarge/reduce polygon

No.	Name	Type	File name	Outline
108	GPOffsetPoly	void	GPOffsetPoly.c	Polygon offset processing
109	GPPhysicalToVirtualPoly	void	GPPhysicalToVirtualPoly.c	Convert polygon physical coordinates to virtual coordinates
112	GMSetMemorySpace	short	GMemory.c	Register memory area information
113	GMDeleteMemorySpace	short	GMemory.c	Delete registered memory area
114	GMMemoryAlloc	void*	GMemory.c	Allocate memory area
115	GMMemoryFree	short	GMemory.c	Release memory area
116	GDCreateVram	HGVRAM	GDCreateVram.c	Create virtual VRAM
117	GDDeleteVram	void	GDDeleteVram.c	Release virtual VRAM
118	GDFlushScreenForce	void	GDFlushScreenForce.c	Forcible refresh screen
119	GDGetGradationInfo	void	GDGetGradationInfo.c	Get gradation information
120	GDGetLocalOrigin	void	GDGetLocalOrigin.c	Get local coordinate origin
121	GDGetVram	HGVRAM	GDGetVram.c	Get system VRAM
122	GDRemoveDrawableRegion	short	GDRemoveDrawableRegion.c	Delete rectangular area from drawable area
123	GDSetDrawableRegion	short	GDSetDrawableRegion.c	Set drawable area
124	GDSetGradationInfo	void	GDSetGradationInfo.c	Set gradation information
125	GDSetLocalOrigin	void	GDSetLocalOrigin.c	Set local coordinate origin

2.2 Outlines of GDraw Functions

GDBeginDraw	
	GDBeginDraw.c
Outline	Begin drawing
Syntax	void GDBeginDraw(HGDRAW *hDraw*)
Argument	(i/o) HGDRAW hDraw : draw handle
Return value	None
Details	This must be called just once when drawing on the screen for drawing basic figures, displaying image data and the like.
File name	GDBeginDraw.c

GDBitBlt

GDBitBlt.c

Outline	Block transfer

Syntax

```
void GDBitBlt(
  HGDRAW    hDraw,
  HGVRAM    hSrc,
  short     nSrcX,
  short     nSrcY,
  short     nWidth,
  short     nHeight,
  HGVRAM    hDest,
  short     nDestX,
  short     nDestY,
  short     nDrawingMode
)
```

Argument

(i/o) HGDRAW hDraw	: drawing handle
(i) HGVRAM hSrc	: source VRAM handle
(i) short nSrcX	: source X coordinate
(i) short nSrcY	: source Y coordinate
(i) short nWidth	: transmission area width
(i) short nHeight	: transmission area height
(i) HGVRAM hDest	: destination VRAM handle
(i) nDestX	: destination X coordinate
(i) nDestY	: destination Y coordinate
(i) nDrawingMode	: raster operation

Return value	None

Details

Transfers the nWidth*nHeight sized area based on the specified point (nSrcX, nSrcY) in the VRAM of the transmission source (hSrc) to the nWidth*nHeight sized area based on another specified point (nDestX, nDestY) in the transmission destination (hDest).
The transfer method depends on nDrawingMode.
Corresponds to physical coordinates BitBlt only, and not virtual coordinates.

File name	GDBitBlt.c

GDCalcBorderRect

GDCalcBorderRect.c

Outline	Calculate rectangular area based on 3D border settings
Syntax	void GDCalcBorderRect(GRect *pBounds, GRect *pInside, GRect *pBorderRect, GBorder *pBorder)
Argument	(i) GRect *pBounds : perimeter rectangle area (o) GRect *pInside : rectangle area inside 3D border (o) GRect *pBorderRect : 3D border area (i) GBorder *pBorder : 3D border
Return value	None
Details	Calculates the position of the rectangle when a 3D border is drawn within pBounds based on the pBorder settings. The rectangular area within the 3D border is stored in plnside. The perimeter rectangle for the 3D border is stored in pBorderRect.
File name	GDCalcBorderRect.c

GDCloseDrawHandle

GDCloseDrawHandle.c

Outline	Release drawing handle
Syntax	void GDCloseDrawHandle(HGDRAW hDraw)
Argument	(i/o) HGDRAW hDraw : drawing handle to be released
Return value	None
Details	The drawing handle is released.
File name	GDCloseDrawHandle.c

GDDefineFillPattern

GDDefineFillPattern.c

Outline	Define fill pattern
Syntax	void GDDefineFillPattern(HGDRAW *hDraw,* short *nFillPattern,* char **pcPattern*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nFillPattern : fill pattern specification (i) char *pcPattern : fill pattern definition
Return value	None
Details	Set pcPattern in nFillPattern No. fill pattern. When a pattern No. outside of the range is set, GERROR_BAD_PATTERN_NUM is set as the error.
File name	GDDefineFillPattern.c

GDDefinePenDash

GDDefinePenDash.c

Outline	Define line pattern
Syntax	void GDDefinePenDash(HGDRAW *hDraw,* short *nPenDash,* char **pcDash*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nPenDash : line pattern No. specification (i) char *pcDash : line pattern
Return value	None
Details	Set pcDash in nPenDash No. line pattern. Since nPenDash number 0 is set to solid line, specify 1 or higher.
File name	GDDefinePenDash.c

GDDrawArc

GDDrawArc.c

Outline	Draw arc
Syntax	void GDDrawArc(HGDRAW *hDraw,* GRect **pgrRect,* short *nStartAngle,* short *nEndAngle*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GRect *pgrRect : rectangle area (i) short nStartAngle : 10 times of the starting angle (i) short nEndAngle : 10 times of the ending angle
Return value	None
Details	Draws an arc in contact with the inside of the rectangle defined with pgrRect. The starting angle is nStartAngle, and the ending angle is nEndAngle. Line color, line type, line thickness and raster operation are enabled as drawing attributes.
File name	GDDrawArc.c

GDDrawBezier

GDDrawBezier.c

Outline	Draw Bezier curve
Syntax	void GDDrawBezier(HGDRAW *hDraw,* GPoly **pgpPoly*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GPoly *pgpPoly : polygon structure
Return value	None
Details	Draws a Bezier curve joining the points defined with pgpPoly. Line color, line type, line thickness and raster operation are enabled as drawing attributes.
File name	GDDrawBezier.c

GDDrawCaptionCursorString

GDDrawCaptionCursorString.c

Outline	Draw caption character string (with cursor display)

Syntax

```
void GDDrawCaptionCursorString(
  HGDRAW   hDraw,
  GRect    *pBounds,
  GCaption  * pCaption,
  GCursor  *pCursor,
  long     lCursorPos,
  char     *pString
)
```

Argument

(i/o) HGDRAW hDraw	: drawing handle
(i) GRect *pBounds	: rectangle area
(i) GCaption *pCaption	: caption information
(i/o) GCursor *pCursor	: data with cursor displayed
(i) long lCursorPos	: character cursor display position
(i) char *pString	: character string

Return value	None
Details	pString character string is drawn with the cursor displayed according to the pCaption settings so that it fills the pBounds rectangular area.
File name	GDDrawCaptionCursorString.c

GDDrawCaptionString

GDDrawCaptionString.c

Outline	Draw caption character string
Syntax	void GDDrawCaptionString(HGDRAW *hDraw,* GRect **pBounds,* GCaption **pCaption,* char **pString*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GRect *pBounds : rectangle area (i) GCaption *pCaption : caption information (i) char *pString : character string
Return value	None
Details	pString character string is drawn according to the pCaption settings so that it fills the pBounds rectangular area.
File name	GDDrawCaptionString.c

GDDrawCaptionStringResource

GDDrawCaptionStringResource.c

Outline	Draw caption based on character string resources

Syntax

```
void GDDrawCaptionStringResource(
  HGDRAW          hDraw,
  GRect           *pBounds,
  GCaption        *pCaption,
  unsigned short  usResourceID
)
```

Argument

(i/o) HGDRAW hDraw	: drawing handle
(i) GRect *pBounds	: rectangle area
(i) GCaption *pCaption	: caption information
(i) unsigned short usResourceID	: character string resource ID

Return value	None
Details	usResourceID resource character string is drawn according to the pCaption settings so that it fills the pBounds rectangular area.
File name	GDDrawCaptionStringResource.c

GDDrawGrf

GDDrawGrf.c

Outline

Draw BMP

Syntax

```
short GDDrawGrf(
  HGDRAW         hDraw,
  GRFHeader      *pGHeader,
  unsigned char  *pData,
  short          nX,
  short          nY,
  short          nWidth,
  short          nHeight
)
```

Argument

(i/o) HGDRAW hDraw	: drawing handle
(i) GRFHeader *pGHeader	: BMP information structure
(i) unsigned char *pData	: BMP data
(i) short nX	: coordinates for starting drawing (upper left of BMP)
(i) short nY	: coordinates for starting drawing (upper left of BMP)
(i) short nWidth	: display all when drawing area size is 0 or lower
(i) short nHeight	: display all when drawing area size is 0 or lower

Return value

FALSE : drawing failed
TRUE : drawing succeeded

Details

The pData image having pGHeader information is drawn in the size of nWidth*nHeight starting at (nX,nY).
When 0 is specified for nWidth, it is displayed in the size of the image.
When 0 is specified for nHeight, it is displayed in the size of the image.

File name

GDDrawGrf.c

GDDrawImage

GDDrawImage.c

Outline	Draw image

Syntax

```
void GDDrawImage(
  HGDRAW   hDraw,
  GImage   *pImage,
  short    nX,
  short    nY,
  short    nWidth,
  shor     nHeight
)
```

Argument		
	(i/o) HGDRAW hDraw	: drawing handle
	(i) GImage *pImage	: image
	(i) short nX	: drawing start position (x coordinate)
	(i) short nY	: drawing start position (y coordinate)
	(i) short nWidth	: drawing area width
	(i) short nHeight	: drawing area height

Return value	None

Details

Draws the GImage image in the size of nWidth*nHeight starting at (nX, nY).
GImage contains the data for any of the images that are supported in the GUI library as standard format data, and the processing function is called according to the type of image.
At present this is only Graphic Raster Format (GRF)

File name	GDDrawImage.c

GDDrawImageResource

GDDrawImageResource.c

Outline	Draw resource image

Syntax

```
void GDDrawImageResource(
  HGDRAW          hDraw,
  unsigned short  usResourceID,
  short           nX,
  short           nY,
  short           nWidth,
  short           nHeight
)
```

Argument	(i/o) HGDRAW hDraw	: drawing handle
	(i) usResourceID	: image resource ID
	(i) short nX	: drawing start position (X coordinate)
	(i) short nY	: drawing start position (Y coordinate)
	(i) short nWidth	: drawing area width
	(i) short nHeight	: drawing area height

Return value	None
Details	Draws the usResourceID resource image in the size of nWidth*nHeight starting at (nX, nY).
File name	GDDrawImageResource.c

GDDrawOval

GDDrawOval.c

Outline	Draw oval
Syntax	void GDDrawOval(HGDRAW *hDraw,* GRect **pgrRect*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GRect *pgrRect : rectangle area
Return value	None
Details	Draws an (oval) circle in contact with the inside of the rectangle defined with pgrRect. For the fill inside the shape, the fill pattern, foreground color and background color, and for the perimeter, line color, line thickness, line type and raster operation are enabled as drawing attributes.
File name	GDDrawOval.c

GDDrawOvalBorder

GDDrawOvalBorder.c

Outline

Draw (oval) circular 3D border

Syntax

```
void GDDrawOvalBorder(
  HGDRAW   hDraw,
  GRect    *pBounds,
  GRect    *pInside,
  GBorder  *pBorder,
  short    nStatus
)
```

Argument

(i/o) HGDRAW hDraw : drawing handle
(i) GRect *pBounds : rectangle area
(i) GRect *pInside : rectangle in contact inside
(i) GBorder *pBorder : 3D border
(i) short nStatus : status
FALSE : convex status
TRUE : concave status

Return value

None

Details

Draws the 3D border in pBorder in an (oval) circle shape in the size of the rectangle in contact with the inside of the rectangular area in pBounds. When, in this instance, nStatus is FALSE, filling of the upper left and lower right borders is done with the colors that have been set, but if it is TRUE, the colors for filling of the upper left and lower right borders is reversed.

File name

GDDrawOvalBorder.c

GDDrawOvalBorderResource

GDDrawOvalBorderResource.c

Outline

Draw (oval) circular resource 3D border

Syntax

```
void GDDrawOvalBorderResource(
  HGDRAW          hDraw,
  GRect           *pBounds,
  GRect           *pInside,
  unsigned short  usResourceID,
  short           nStatus
)
```

Argument

(i/o) HGDRAW hDraw		: drawing handle
(i) GRect *pBounds		: rectangle area
(i) GRect *pInside		: rectangle in contact inside
(i) unsigned short usResourceID		: 3D border resource ID
(i) short nStatus		: status
	FALSE	: convex status
	TRUE	: concave status

Return value

None

Details

Draws the 3D border in usResourceID in an (oval) circle shape in the size of the rectangle in contact with the inside of the rectangular area in pBounds. When, in this instance, nStatus is FALSE, filling of the upper left and lower right borders is done with the colors that have been set, but if it is TRUE, the colors for filling of the upper left and lower right borders is reversed.

File name

GDDrawOvalBorderResource.c

GDDrawPie

GDDrawPie.c

Outline	Draw sector
Syntax	void GDDrawPie(HGDRAW *hDraw,* GRect **pgrRect,* short *nStartAngle,* short *nEndAngle*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GRect *pgrRect : rectangle area (i) short nStartAngle : 10 times of the starting angle (i) short nEndAngle : 10 times of the ending angle
Return value	None
Details	Draws a sector in contact with the inside of the rectangle defined with pgrRect. The starting angle is nStartAngle, and the ending angle is nEndAngle. For the fill inside the shape, the fill pattern, foreground color and background color, and for the perimeter, line color, line thickness, line type and raster operation are enabled as drawing attributes.
File name	GDDrawPie.c

GDDrawPoly

GDDrawPoly.c

Outline	Draw polygon
Syntax	void GDDrawPoly(HGDRAW *hDraw,* GPoly **pgpPoly*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GPoly *pgpPoly : polygon structure
Return value	None
Details	Draws the polygon defined in pgpPoly. For the fill inside the shape, the fill pattern, foreground color and background color, and for the perimeter, line color, line thickness, line type and raster operation are enabled as drawing attributes.
File name	GDDrawPoly.c

GDDrawRect

GDDrawRect.c

Outline	Draw rectangle
Syntax	void GDDrawRect(HGDRAW *hDraw,* GRect **pgrRect*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GRect *pgrRect : rectangle area
Return value	None
Details	Draws the rectangle in the rectangular area defined in pgrRect. For the fill inside the shape, the fill pattern, foreground color and background color, and for the perimeter, line color, line thickness, line type and raster operation are enabled as drawing attributes.
File name	GDDrawRect.c

GDDrawRectBorder

GDDrawRectBorder.c

Outline

Draw rectangular 3D border

Syntax

```
void GDDrawRectBorder(
  HGDRAW   hDraw,
  GRect    *pBounds,
  GRect    *pInside,
  GBorder  *pBorder,
  short    nStatus
)
```

Argument

(i/o) HGDRAW hDraw : drawing handle
(i) GRect *pBounds : rectangle area
(i) GRect *pInside : rectangle in contact inside
(i) GBorder *pBorder : 3D border
(i) short nStatus : status
FALSE : convex status
TRUE : concave status

Return value

None

Details

Draws a 3D border in pBorder in an rectangle in the size of the rectangle in contact with the inside of the rectangular area in pBounds.
When, in this instance, nStatus is FALSE, filling of the upper left and lower right borders is done with the colors that have been set, but if it is TRUE, the colors for filling of the upper left and lower right borders is reversed.

File name

GDDrawRectBorder.c

GDDrawRectBorderResource

GDDrawRectBorderResource.c

Outline	Draw resource rectangular 3D border

Syntax

```
void GDDrawRectBorderResource(
  HGDRAW          hDraw,
  GRect           *pBounds,
  GRect           *pInside,
  unsigned short  usResourceID,
  short           nStatus
)
```

Argument

(i/o) HGDRAW hDraw		: drawing handle
(i) GRect *pBounds		: rectangle area
(i) GRect *pInside		: rectangle in contact inside
(i) unsigned short usResourceID		: 3D border
(i) short nStatus		: status
	FALSE : convex status	
	TRUE : concave status	

Return value	None
Details	Draws a border in usResourceID in a rectangle in the size of the rectangle in contact with the inside of the rectangular area in pBounds. When, in this instance, nStatus is FALSE, filling of the upper left and lower right borders is done with the colors that have been set, but if it is TRUE, the colors for filling of the upper left and lower right borders is reversed.
File name	GDDrawRectBorderResource.c

GDDrawRRect

GDDrawRRect.c

Outline	Draw rounded corner rectangle
Syntax	void GDDrawRRect(HGDRAW *hDraw,* GRect **pgrRect,* short *nX,* short *nY*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GRect *pgrRect : rectangle area (i) short nX : X axis of corner arc (i) short nY : Y axis of corner arc
Return value	None
Details	Draws the rounded corner rectangle in the rectangular area defined in pgrRect. The rounding of the corners is defined with nX and nY. For the fill inside the shape, the fill pattern, foreground color and background color, and for the perimeter, line color, line thickness, line type and raster operation are enabled as drawing attributes.
File name	GDDrawRRect.c

GDDrawString

GDDrawString.c

Outline	Draw character string
Syntax	void GDDrawString(HGDRAW *hDraw,* char **pszString*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) char *pszString : draw character string
Return value	None
Details	Draws the pszString character string starting at the current pen position. Pen color, background color and text mode are enabled as the drawing attributes.
File name	GDDrawString.c

GDDrawStringResource

GDDrawStringResource.c

Outline	Draw resource character string
Syntax	void GDDrawStringResource(HGDRAW *hDraw,* unsigned short *usResourceID*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) unsigned short usResourceID : draw resource character string
Return value	None
Details	Draws the usResourceID resource character string starting at the current pen position. Pen color, background color and text mode are enabled as the drawing attributes.
File name	GDDrawStringResource.c

GDEndDraw

GDEndDraw.c

Outline	End drawing
Syntax	void GDEndDraw(HGDRAW *hDraw*)
Argument	(i/o) HGDRAW hDraw : drawing handle
Return value	None
Details	Quits screen drawing. This is called after drawing on the screen for basic figure drawing, data display and the like.
File name	GDEndDraw.c

GDFlushScreen

GDFlushScreen.c

Outline	Refresh screen
Syntax	void GDFlushScreen(HGDRAW *hDraw,* short *nX,* short *nY,* short *nWidth,* short *nHeight*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nX : X coordinate for drawing starting point (i) short nY : Y coordinate for drawing starting point (i) short nWidth : drawing width (i) short nHeight : drawing height
Return value	None
Details	A screen for the size of nWidth*nHeight is refreshed in the current coordinate system with (nX, nY) as the starting point.
File name	GDFlushScreen.c

GDGetCharHeight

GDGetCharHeight.c

Outline	Get specified character height
Syntax	short GDGetCharHeight(HGDRAW *hDraw,* short *nCharCode*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nCharCode : character code
Return value	Character height
Details	Get specified character height with current font settings.
File name	GDGetCharHeight.c

GDGetCharWidth

GDGetCharWidth.c

Outline	Get specified character width
Syntax	short GDGetCharWidth(HGDRAW *hDraw,* short *nCharCode*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nCharCode : character code
Return value	Character width
Details	Get specified character width with current font settings.
File name	GDGetCharWidth.c

GDGetNextChar

GDGetNextChar.c

Outline	Get next character
Syntax	char* GDGetNextChar(HGDRAW *hDraw,* char **pszString,* char **pszNow*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) char *pszString : character string (i) char *pszNow : character position - starting point for the next character search
Return value	Pointer to the next character of the specified character
Details	Searches for the next character using the specified character in the specified character string as the starting point.
File name	GDGetNextChar.c

GDGetPalette

GDGetPalette.c

Outline	Get palette color code
Syntax	void GDGetPalette(HGDRAW *hDraw,* short *nStart,* short *nCount,* GColor **pgcColorArray*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nStart : starting No. to get (i) short nCount : number of colors to get (o) GColor *pgcColorArray : array storing color code
Return value	None
Details	nCount color codes starting at nStart in the color pallet are acquired in pgcColorArray.
File name	GDGetPalette.c

GDGetPixel

GDGetPixel.c

Outline	Get color code for specified coordinates
Syntax	GColor GDGetPixel(HGDRAW *hDraw,* short *nX,* short *nY*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nX : X coordinate (i) short nY : Y coordinate
Return value	Color code
Details	Gets the color code for (nX, nY). The color code depends on the current color environment.
File name	GDGetPixel.c

GDGetPrevChar

GDGetPrevChar.c

Outline	Get previous character
Syntax	char* GDGetPrevChar(HGDRAW *hDraw,* char * *pszString,* char * *pszNow*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) char *pszString : character string (i) char *pszNow : character position - starting point for the previous character search Character position
Return value	Pointer to the previous character of the specified character
Details	Searches for the previous character using the specified character in the specified character string as the starting point.
File name	GDGetPrevChar.c

GDGetStringLength

GDGetStringLength.c

Outline	Get character length
Syntax	unsigned long GDGetStringLength(HGDRAW *hDraw,* char *pszString*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) char *pszString : character string in which the characters are to be counted
Return value	Number of characters
Details	Gets the number of characters in the specified character string.
File name	GDGetStringLength.c

GDGetStringWidth

GDGetStringWidth.c

Outline	Get specified character width
Syntax	short GDGetStringWidth(HGDRAW *hDraw,* char **pszString*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) char *pszString : character string
Return value	Character string width
Details	Gets the width of a character string specified by pszString.
File name	GDGetStringWidth.c

GDInceptClipRect

GDInceptClipRect.c

Outline	Clip the area of overlap between the specified rectangle and the current clip area
Syntax	short GDInceptClipRect(HGDRAW *hDraw,* GRect **pgrClip,* GRect **pOldClip*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GRect *pgrClip : rectangle area (o) GRect *pOldClip : clip area before change
Return value	FALSE : clipping failed TRUE : clipping succeeded
Details	Defines the portion of overlap between the pgrClip rectangular area displaying characters or graphics on the screen and the current clip area. The clip area up to that point is stored in pOldClip. Either physical or virtual coordinates can be used for the clip area, and the setting is applied to the coordinate system at the time this function is called.
File name	GDInceptClipRect.c

GDInitPlatform

GDInitPlatform.c

Outline	Carrying out initialization processing for GUI library environment
Syntax	short GDInitPlatform(void **pParameter*)
Argument	(i) void* pParameter : additional platform information
Return value	FALSE : initialization failed TRUE : initialization succeeded
Details	Initializes the platform that uses the GUI library.
File name	GDInitPlatform.c

GDLineTo

GDLineTo.c

Outline	Draw line
Syntax	void GDLineTo(HGDRAW *hDraw,* short *nX,* short *nY*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nX : X coordinate (i) short nY : Y coordinate
Return value	None
Details	Draws a line from the current pen position to (nX, nY).
File name	GDLineTo.c

GDMoveTo

GDMoveTo.c

Outline	Move pen position
Syntax	void GDMoveTo(HGDRAW *hDraw,* short *nX,* short *nY*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nX : X coordinate (i) short nY : Y coordinate
Return value	None
Details	Moves pen position to (nX, nY).
File name	GDMoveTo.c

GDOffsetLocalOrigin

GDOffsetLocalOrigin.c

Outline	Offset to local coordinate origin
Syntax	void GDOffsetLocalOrigin(HGDRAW *hDraw,* GPoint **pgptPoint*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GPoint *pgptPoint : point structure
Return value	None
Details	An offset of pgptPoint.nX in the X direction and pgptPoint.nY in the Y direction is applied to the origin for the local (physical) coordinates.
File name	GDOffsetLocalOrigin.c

GDOpenDrawHandle

GDOpenDrawHandle.c

Outline	Get drawing handle
Syntax	HGDRAW GDOpenDrawHandle(short *nType,* void **pParameter*)
Argument	(i) short nType : display mode (i) void *pParameter : additional handle information
Return value	NULL : acquisition failed Other than NULL : drawing handle
Details	Gets the drawing handle.
File name	GDOpenDrawHandle.c

GDQueryError

GDQueryError.c

Outline	Get final error code
Syntax	unsigned short GDQueryError(HGDRAW *hDraw*)
Argument	(i/o) HGDRAW hDraw : drawing handle
Return value	Error No.
Details	The final error code within the GUI library is acquired.
File name	GDQueryError.c

GDQuitPlatform

GDQuitPlatform.c

Outline	Quit processing for GUI library environment
Syntax	short GDQuitPlatform()
Argument	None
Return value	FALSE : failed TRUE : succeeded
Details	Carries out end processing for the platform environment used by the GUI library.
File name	GDQuitPlatform.c

GDSetBackColor

GDSetBackColor.c

Outline Set background color

Syntax
```
void GDSetBackColor(
  HGDRAW    hDraw,
  GColor    gcBackColor
)
```

Argument
(i/o) HGDRAW hDraw : drawing handle
(i) GColor gcBackColor : background color to be set

Return value None

Details Sets the background color.
When NO_DRAW is set for the background color, the background color pixels are not filled in during drawing.

NO_DRAW : no background fill
Color code : fill background using specified color code

File name GDSetBackColor.c

GDSetBrush

GDSetBrush.c

Outline Set brush

Syntax
```
void GDSetBrush(
  HGDRAW    hDraw,
  GBrush    *pBrush
)
```

Argument
(i/o) HGDRAW hDraw : drawing handle
(i) GBrush *pBrush : brush being set

Return value None

Details The brush to be used is set as a fill attribute.
The brush set is registered as a fill pattern, foreground color and background color setting.

File name GDSetBrush.c

GDSetClipRect

GDSetClipRect

Outline	Define clipping area
Syntax	void GDSetClipRect(HGDRAW *hDraw,* GRect **pgrClip*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GRect *pgrClip : rectangle area
Return value	None
Details	A rectangular area for displaying characters or graphics on the screen is set as pgrClip. Either physical or virtual coordinates can be used for the clip area, and the setting is applied to the coordinate system at the time this function is called.
File name	GDSetClipRect.c

GDSetDisplay

GDSetDisplay.c

Outline	Set graphics mode
Syntax	short GDSetDisplay(HGDRAW *hDraw,* short *nType*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nType : graphics mode to be set
Return value	FALSE : setting failed TRUE : setting succeeded
Details	Sets the graphics mode. The mode depends on the platform, so it is not defined as the GUI library.
File name	GDSetDisplay.c

GDSetDrawingMode

GDSetDrawingMode.c

Outline	Set raster operations for drawing
Syntax	void GDSetDrawingMode(HGDRAW *hDraw,* short *nMode*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nMode : raster operation
Return value	None
Details	Sets the raster operation for drawing. GRASTER_OPERATION_REPLACE : overwrite GRASTER_OPERATION_AND : AND with base GRASTER_OPERATION_OR : OR with base GRASTER_OPERATION_XOR : XOR with base
File name	GDSetDrawingMode.c

GDSetFillAttribute

GDSetFillAttribute.c

Outline	Set fill attributes

Syntax

```
void GDSetFillAttribute(
  HGDRAW   hDraw,
  short    nPattern,
  GColor   gcForeColor,
  GColor   gcBackColor
)
```

Argument

(i/o) HGDRAW hDraw : drawing handle
(i) short nPattern : fill pattern
(i) GColor gcForeColor : foreground color
(i) GColor gcBackColor : background color

Return value

None

Details

Sets the fill attributes.

nPattern

- NO_DRAW : no fill
- FILL_FORE_COLOR : fill with single foreground color
- FILL_BACK_COLOR : fill with single background color
- 0 to 37 : fill using pattern for specified No.

gcForeColor

- NO_DRAW : no foreground fill
- Color code : fill foreground using specified color code

gcBackColor

- NO_DRAW : no background fill
- Color code : fill background using specified color code

File name

GDSetFillAttribute.c

GDSetFillColor

GDSetFillColor.c

Outline	Set colors for foreground and background
Syntax	void GDSetFillColor(HGDRAW *hDraw,* GColor *gcForeColor,* GColor *gcBackColor*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GColor gcForeColort : foreground color to be se (i) GColor gcBackColor : background color to be set
Return value	None
Details	Sets the foreground color and background color. gcForeColor NO_DRAW : no foreground fill Color code : fill foreground using specified color code gcBackColor NO_DRAW : no background fill Color code : fill background using specified color code
File name	GDSetFillColor.c

GDSetFillPattern

GDSetFillPattern.c

Outline	Set fill pattern
Syntax	void GDSetFillPattern(HGDRAW *hDraw,* short *nPattern*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nPattern : fill pattern
Return value	None
Details	Sets the fill pattern. nPattern NO_DRAW : no fill FILL_FORE_COLOR : fill with single foreground color FILL_BACK_COLOR : fill with single background color 0 to 37 : fill using pattern for specified No.
File name	GDSetFillPattern.c

GDSetFont

GDSetFont.c

Outline	Set font
Syntax	short GDSetFont(HGDRAW *hDraw,* HGFONT *hFont*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) HGFONT hFont : font handle
Return value	FALSE : setting failed TRUE : setting succeeded
Details	Set the hFont font as the font for character drawing.
File name	GDSetFont.c

GDSetForeColor

GDSetForeColor.c

Outline	Set foreground color
Syntax	void GDSetForeColor(HGDRAW *hDraw,* GColor *gcForeColor*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GColor gcForeColor : foreground color to be set
Return value	None
Details	Sets the foreground color. gcForeColor NO_DRAW : no foreground fill Color code : fill foreground using specified color code
File name	GDSetForeColor.c

GDSetLocalMode

GDSetLocalMode.c

Outline	Set to local (physical) coordinate system
Syntax	void GDSetLocalMode(HGDRAW *hDraw*)
Argument	(i/o) HGDRAW hDraw : drawing handle
Return value	None
Details	Sets the coordinate system to local (physical) coordinates. After the coordinate system is set to local coordinates, all GUI library functions are executed using the local (physical) coordinates system.
File name	GDSetLocalMode.c

GDSetPenColor

GDSetPenColor.c

Outline	Set line and character color
Syntax	void GDSetPenColor(HGDRAW *hDraw,* GColor *gcPenColor*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GColor gcPenColor : color to be set for lines/characters
Return value	None
Details	Sets the color for lines and characters. gcPenColor NO_DRAW : no drawing Color code : draw lines and characters using the specified color code
File name	GDSetPenColor.c

GDSetPenDash

GDSetPenDash.c

Outline	Set line pattern
Syntax	void GDSetPenDash(HGDRAW *hDraw,* short *nDash*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nDash : line pattern
Return value	None
Details	Specifies the line pattern to use when drawing lines. nDash 0 : solid line 1 to 8 : defined line pattern
File name	GDSetPenDash.c

GDSetPenJoin

GDSetPenJoin.c

Outline	Specify joint for rectangles and polygons + D968
Syntax	void GDSetPenJoin(HGDRAW *hDraw,* short *nJoin*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nJoin : type of joint
Return value	None
Details	Specifies the joint for rectangle and polygon perimeters. nJoin NONE : none BEVEL : line ROUND : rounded corners
File name	GDSetPenJoin.c

GDSetPenSize

GDSetPenSize.c

Outline	Set line width (pen size)
Syntax	void GDSetPenSize(HGDRAW *hDraw,* short *nX,* short *nY*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nX : thickness in X direction (i) short nY : thickness in Y direction
Return value	None
Details	Sets the thickness of lines and perimeter lines. nX is the horizontal thickness, and nY is the vertical thickness.
File name	GDSetPenSize.c

GDSetPixel

GDSetPixel.c

Outline	Draw pixels
Syntax	void GDSetPixel(HGDRAW *hDraw,* short *nX,* short *nY*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nX : X coordinate (i) short nY : Y coordinate
Return value	None
Details	Draws a point in the specified (nX, nY) coordinates. After point drawing, the current position changes to the specified position.
File name	GDSetPixel.c

GDSetSimpleFont

GDSetSimpleFont.c

Outline	Set simple font
Syntax	HGFONT GDSetSimpleFont(HGDRAW *hDraw,* GSimpleFont **pFont*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i)GSimpleFont *pFont : set simple font
Return value	Font handle set
Details	Changes the current font to a simple font pFont.
File name	GDSetSimpleFont.c

GDSetSimpleFontResource

GDSetSimpleFontResource.c

Outline	Set resource simple font
Syntax	HGFONT GDSetSimpleFontResource(HGDRAW *hDraw,* unsigned short *usResourceID*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) unsigned short usResourceID : set simple font
Return value	Font handle set
Details	Changes the current font to a resource simple font usResourceID.
File name	GDSetSimpleFontResource.c

GDSetVirtualMode

GDSetVirtualMode.c

Outline	Set to virtual coordinate system
Syntax	void GDSetVirtualMode(HGDRAW *hDraw*)
Argument	(i/o) HGDRAW hDraw : drawing handle
Return value	None
Details	The coordinate system is set to the virtual coordinate system. After the coordinate system is set to the virtual coordinate system, all GUI library functions are executed using the virtual coordinate system.
File name	GDSetVirtualMode.c

GDSetVirtualRect

GDSetVirtualRect.c

Outline	Set rectangular area in virtual coordinates
Syntax	void GDSetVirtualRect(HGDRAW *hDraw,* GRect **pgrVirtual*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GRect *pgrVirtual : virtual coordinate rectangle area
Return value	None
Details	The rectangular area for the virtual coordinate system is set in pgrVirtual.
File name	GDSetVirtualRect.c

GDSetVram

GDSetVram.c

Outline	Set VRAM for drawing destination
Syntax	short GDSetVram(HGDRAW *hDraw,* short *nType,* HGVRAM *hVram*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) short nType DEFAULT_SCREEN : set actual VRAM for drawing destination MEMORY_SCREEN : set virtual VRAM for drawing destination (i) HGVRAM hVram : VRAM handle when specifying virtual VRAM
Return value	FALSE : failed TRUE : succeeded
Details	The drawing destination VRAM is specified in hVram.
File name	GDSetVram.c

GFGetFontAttribute

GFGetFontAttribute.c

Outline	Get font attributes
Syntax	short GFGetFontAttribute(HGFONT *hFont,* GFontAttribute **pgfaAttribute*)
Argument	(i) HGFONT hFont : font (o) GFontattribute *pgfaAttribute : attribute
Return value	FALSE : failed TRUE : succeeded
Details	Gets the attributes for the font handle font.
File name	GFGetFontAttribute.c

GFGetFontCount

GFGetFontCount.c

Outline	Get number of fonts that can be used
Syntax	short GFGetFontCount()
Argument	None
Return value	Number of fonts
Details	This is supported by the current platform. Gets fonts.
File name	GFGetFontCount.c

GFGetFontInformation

GFGetFontInformation.c

Outline	Get font information for the specified number
Syntax	short GFGetFontInformation(short *nIndex,* GFont ***pgfFont*)
Argument	(i) short nIndex : font No. (o) GFont **pgfFont : font information
Return value	FALSE : failed TRUE : succeeded
Details	This is supported by the platform. Gets the font information for the specified font No.
File name	GFGetFontInformation.c

GFGetFontUserAttribute

GFGetFontUserAttribute.c

Outline	Get user font attributes
Syntax	short GFGetFontUserAttribute(HGFONT *hFont,* short **pnAttributeArray*)
Argument	(i) HGFONT hFont : font (o) short *pnAttributeArray : attribute unique to platform
Return value	FALSE : failed TRUE : succeeded
Details	Gets the font attributes peculiar to the font handle.
File name	GFGetFontUserAttribute.c

GFGetSize

GFGetSize.c

Outline	Get font size
Syntax	short GFGetSize(HGFONT *hFont,* short **nWidth,* short **nHeight*)
Argument	(i) HGFONT hFont : font (o) short *nWidth : width (o) short *nHeight : height
Return value	FALSE : failed TRUE : succeeded
Details	Gets the current font size for the specified font handle.
File name	GFGetSize.c

GFLoadFont

GFLoadFont.c

Outline	Load specified font
Syntax	HGFONT GFLoadFont(char *pszFontName)
Argument	(i) char *pszFontName : font name
Return value	Font handle
Details	Sets the specified font to a usable status and returns the handle.
File name	GFLoadFont.c

GFReleaseFont

GFReleaseFont.c

Outline	Release font
Syntax	void GFReleaseFont(HGFONT *hFont*)
Argument	(i) HGFONT hFont : font
Return value	None
Details	Releases the font for the specified handle.
File name	GFReleaseFont.c

GFSetFontAttribute

GFSetFontAttribute.c

Outline	Set font attributes
Syntax	short GFSetFontAttribute(HGFONT *hFont,* GFontAttribute **pgfaAttribute*)
Argument	(i) HGFONT hFont : font (i) GFontattribute *pgfaAttribute : attribute
Return value	FALSE : failed TRUE : succeeded
Details	Sets the attributes for the specified font.
File name	GFSetFontAttribute.c

GFSetFontUserAttribute

GFSetFontUserAttribute.c

Outline	Set font user attributes
Syntax	short GFSetFontUserAttribute(HGFONT *hFont,* short **pnAttributeArray*)
Argument	(i) HGFONT hFont : font (i) short *pnAttributeArray : attribute unique to platform
Return value	FALSE : failed TRUE : succeeded
Details	Sets the font attributes peculiar to the font handle.
File name	GFSetFontUserAttribute.c

GFSetSize

GFSetSize.c

Outline	Set character size
Syntax	short GFSetSize(HGFONT *hFont*, short *nWidth*, short *nHeight*)
Argument	(i) HGFONT hFont : font being specified (i) short nWidth : width (i) short nHeight : height
Return value	FALSE : failed TRUE : succeeded
Details	Sets the character size for the specified font.
File name	GFSetSize.c

GRDuplicateRect

GRDuplicateRect.c

Outline	Copy rectangle
Syntax	void GRDuplicateRect(GRect **pgrSrc*, GRect **pgrDest*)
Argument	(i) GRect *pgrSrc : copy source (o) GRect *pgrDest : copy destination
Return value	None
Details	Copies the rectangular area defined in pgrSrc to the pgrDest rectangular area.
File name	GRDuplicateRect.c

GREqualRect

GREqualRect.c

Outline	Confirm that rectangle areas are the same
Syntax	short GREqualRect(GRect **pgrSrc1,* GRect **pgrSrc2*)
Argument	(i) GRect *pgrSrc1 : rectangle area (i) GRect *pgrSrc2 : rectangle area
Return value	FALSE : not the same TRUE : same
Details	Finds out if the rectangular areas defined in pgrSrc1 and pgrSrc2 are the same.
File name	GREqualRect.c

GRGetRect

GRGetRect.c

Outline	Get rectangle area
Syntax	void GRGetRect(GRect **pgrRect,* short **nX1,* short **nY1,* short **nX2,* short **nY2*)
Argument	(i) GRect *pgrRect : rectangle area (o) short *nX1 : X coordinate for upper left corner (o) short *nY1 : Y coordinate for upper left corner (o) short *nX2 : X coordinate for lower right corner (o) short *nY2 : Y coordinate for lower right corner
Return value	None
Details	Gets the rectangle area. The rectangle's upper left corner coordinates will be set in (nX1, nY1) and the lower right corner coordinates will be set in (nX2, nY2).
File name	GRGetRect.c

GRInceptRect

GRInceptRect.c

Outline	Confirm rectangular area overlap
Syntax	short GRInceptRect(GRect **pgrSrc1,* GRect **pgrSrc2,* GRect **pgrDest*)
Argument	(i) GRect *pgrSrc1 : rectangle area (i) GRect *pgrSrc2 : rectangle area (o) GRect *pgrDest : rectangle area
Return value	FALSE : pgrSrc1 and pgrSrc2 do not overlap. (pgrDest not determined) TRUE : pgrSrc1 and pgrSrc2 overlap.
Details	Finds out if the two rectangular areas in pgrSrc1 and pgrSrc2 overlap. If they overlap, set that area in pgrDest.
File name	GRInceptRect.c

GRMapRect

GRMapRect.c

Outline	Enlarge/reduce rectangular area
Syntax	void GRMapRect(GRect **pgrRect,* GRect **pgrFrom,* GRect **pgrTo*)
Argument	(i/o) GRect *pgrRect : rectangle area (i) GRect *pgrFrom : rectangle area (i) GRect *pgrTo : rectangle area
Return value	None
Details	Converts the rectangular area defined in pgrRect in the same proportion as the conversion of the rectangular area defined in pgrFrom to the rectangular area defined in PgrTo.
File name	GRMapRect.c

GRMapToScreenRect

GRMapToScreenRect.c

Outline	Convert rectangular area defined in pgrRect to actual screen coordinate system
Syntax	void GRMapToScreenRect(HGDRAW *hDraw,* GRect **pgrRect*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i/o) GRect *pgrRect : rectangle
Return value	None
Details	Converts the rectangular area defined in pgrRect to the actual screen coordinate system
File name	GRMapToScreenRect.c

GRMapToVirtualRect

GRMapToVirtualRect.c

Outline	Convert the rectangular area defined in pgrRect to the virtual screen coordinate system
Syntax	void GRMapToVirtualRect(HGDRAW *hDraw,* GRect **pgrRect*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GRect *pgrRect : rectangular area
Return value	None
Details	Converts the rectangular area defined in pgrRect to the virtual screen coordinate system
File name	GRMapToVirtualRect.c

GROffsetRect

GROffsetRect.c

Outline	Rectangle offset processing
Syntax	void GROffsetRect(GRect **pgrRect,* short *nDx,* short *nDy*)
Argument	(i/o) GRect *pgrRect : source rectangular area (i) short nDx : increase in X direction (i) short nDy : increase in Y direction
Return value	None
Details	Moves the rectangle defined in pgrRect nDx and nDy in the X direction and Y direction, respectively.
File name	GROffsetRect.c

GRPhysicalToVirtualRect

GRPhysicalToVirtualRect.c

Outline	Convert rectangle physical coordinates to virtual coordinates
Syntax	void GRPhysicalToVirtualRect(HGDRAW *hDraw,* GRect **pgrRect*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i/o) GRect *pgrRect : rectangle area
Return value	None
Details	Converts the rectangular area defined in pgrRect from physical (local) coordinates to virtual coordinates.
File name	GRPhysicalToVirtualRect.c

GRPointInRect

GRPointInRect.c

Outline	Check if point coordinates fall within rectangle coordinates
Syntax	short GRPointInRect(GPoint **pptPoint,* GRect **pgrRect*)
Argument	(i) GPoint *pptPoint : point coordinates (i) GRect *pgrRect : rectangle area
Return value	FALSE : not inside TRUE : inside (including on the pgrRect)
Details	Finds out if the point coordinates defined in pptPoint are within the rectangular area defined in pgrRect.
File name	GRPointInRect.c

GRResizeRect

GRResizeRect.c

Outline	Resize rectangle
Syntax	void GRResizeRect(GRect **pgrRect,* short *nDx,* short *nDy*)
Argument	(i/o) GRect *pgrRect : rectangle area (i) short nDx : increase in X direction (i) short nDy : increase in Y direction
Return value	None
Details	nDx is subtracted from the initial X coordinate of pgrRect and nDx is added to the final X coordinate. Furthermore, nDy is subtracted from the initial Y coordinate and nDy is added to the final Y coordinate.
File name	GRResizeRect.c

GRSetRect

GRSetRect.c

Outline

Define rectangular area

Syntax

```
void GRSetRect(
  GRect   *pgrRect,
  short   nX1,
  short   nY1,
  short   nX2,
  short   nY2
)
```

Argument

(o) GRect *pgrRect1	: rectangle area
(i) short nX1	: X coordinate for upper left corner
(i) short nY1	: Y coordinate for upper left corner
(i) short nX2	: X coordinate for lower right corner
(i) short nY2	: Y coordinate for lower right corner

Return value

None

Details

The coordinates for the rectangle’s upper left corner are specified as (nX1, nY1) and those for the lower right corner as (nX2, nY2).

File name

GRSetRect.c

GRShiftRect

GRShiftRect.c

Outline	Rectangle shift processing
Syntax	short GRShiftRect(GRect **pgrRect,* short *nDx,* short *nDy,* GRect **pgrRect1,* GRect **pgrRect2*)
Argument	(i) GRect *pgrRect : source rectangular area (i) short nDx : increase in X direction (i) short nDy : increase in Y direction (o) GRect *pgrRect1 : empty rectangular area (o) GRect *pgrRect2 : empty rectangular area
Return value	0 : no empty rectangular area 1 : one empty rectangular area 2 : two empty rectangular areas
Details	Moves the rectangle defined in pgrRect nDx and nDy in the X direction and Y direction, respectively. Furthermore, the empty rectangular area created by the movement is set in pgrRect1 and pgrRect2.
File name	GRShiftRect.c

GRUnionRect

GRUnionRect.c

Outline	Join rectangle areas
Syntax	void GRUnionRect(GRect **pgrSrc1,* GRect **pgrSrc2,* GRect **pgrDest*)
Argument	(i) GRect *pgrSrc1 : rectangle area (i) GRect *pgrSrc2 : rectangle area (o) GRect *pgrDest : combined rectangular area
Return value	None
Details	The minimum sized rectangular area surrounding the specified rectangular areas pgrSrc1 and pgrSrc2 is set in pgrDest.
File name	GRUnionRect.c

GRVirtualToPhysicalRect

GRVirtualToPhysicalRect.c

Outline	Convert rectangular area virtual coordinates to physical coordinates
Syntax	void GRVirtualToPhysicalRect(HGDRAW *hDraw,* GRect **pgrRect*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GRect *pgrRect : rectangle area
Return value	None
Details	Converts the rectangular area defined in pgrRect from virtual coordinates to physical (local) coordinates.
File name	GRVirtualToPhysicalRect.c

GLDuplicateLine

GLDuplicateLine.c

Outline	Copy line structure
Syntax	void GLDuplicateLine(GLine **pglSrc,* GLine **pglDest*)
Argument	(i) GLine *pglSrc : copy source (o) GLine *pglDest : copy destination
Return value	None
Details	The line structure defined in pglSrc is copied to the line structure in pglDest.
File name	GLDuplicateLine.c

GLGetLine

GLGetLine.c

Outline	Get line
Syntax	void GLGetLine(GLine **pglLine,* short **nX1,* short **nY1,* short **nX2,* short **nY2*)
Argument	(i) GLine *pglLine : source line structure being acquired (o) short *nX1 : X coordinate for starting point of line structure (o) short *nY1 : Y coordinate for starting point of line structure (o) short *nX2 : X coordinate for ending point of line structure (o) short *nY2 : Y coordinate for ending point of line structure
Return value	None
Details	Gets the content of the line structure pglLine in (nX1,nY1) and (nX2,nY2)
File name	GLGetLine.c

GLMapLine

GLMapLine.c

Outline	Enlarge/reduce line coordinates
Syntax	void GLMapLine(GLine **pglLine,* GRect **pgrFrom,* GRect **pgrTo*)
Argument	(i/o) GLine *pglLine : line structure (i) GRect *pgrForm : rectangle area (i) GRect *pgrTo : rectangle area
Return value	None
Details	Converts coordinates when the line structure in the pgrFrom rectangular area will be projected onto to the pgrTo rectangular area.
File name	GLMapLine.c

GLOffsetLine

GLOffsetLine.c

Outline	Offset processing for line structure
Syntax	void GLOffsetLine(GLine **pglLine,* short *nDx,* short *nDy*)
Argument	(i/o) GLine *pglLine : source line (i) short nDx : increase in X direction (i) short nDy : increase in Y direction
Return value	None
Details	Moves the line defined in pglLine nDx and nDy in the X direction and Y direction, respectively.
File name	GLOffsetLine.c

GLPhysicalToVirtualLine

GLPhysicalToVirtualLine.c

Outline	Convert line structure physical coordinates to virtual coordinates
Syntax	void GLPhysicalToVirtualLine(HGDRAW *hDraw,* GLine **pglLine*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i/o) GLine *pglLine : line structure
Return value	None
Details	Converts the coordinates defined in pglLine from physical (local) coordinates to virtual coordinates.
File name	GLPhysicalToVirtualLine.c

GLSetLine

GLSetLine.c

Outline

Define line structure

Syntax

```
void GLSetLine(
  GLine   *pglLine,
  short   nX1,
  short   nY1,
  short   nX2,
  short   nY2
)
```

Argument

(o) GLine *pglLine	:	line structure
(i) short nX1	:	X coordinate for starting point
(i) short nY1	:	Y coordinate for starting point
(i) short nX2	:	X coordinate for ending point
(i) short nY2	:	Y coordinate for ending point

Return value

None

Details

Defines a line structure. The coordinates for the starting point of a line are specified as (nX1, nY1) and those for the ending point as (nX2, nY2).

File name

GLSetLine.c

GLVirtualToPhysicalLine

GLVirtualToPhysicalLine.c

Outline	Convert line structure virtual coordinates to physical coordinates
Syntax	void GLVirtualToPhysicalLine(HGDRAW *hDraw,* GLine **pglLine*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i/o) GLine *pglLine : line structure
Return value	None
Details	Converts the coordinates defined in pglLine from virtual coordinates to physical (local) coordinates.
File name	GLVirtualToPhysicalLine.c

GPTDuplicatePoint

GPTDuplicatePoint.c

Outline	Copy point structure
Syntax	void GPTDuplicatePoint(GPoint **pgptSrc,* GPoint **pgptDest*)
Argument	(i) GPoint *pgptSrc : copy source (o) GPoint *pgptDest : copy destination
Return value	None
Details	The point structure defined in pgptSrc is copied to pgptDest.
File name	GPTDuplicatePoint.c

GPTGetPoint

GPTGetPoint.c

Outline	Get point structure content
Syntax	void GPTGetPoint(GPoint **pgptPoint,* short **nX,* short **nY*)
Argument	(i) GPoint *pgptPoint : point structure for acquisition source (o) short *nX : X coordinate for point structure (o) short *nY : Y coordinate for point structure
Return value	None
Details	Gets the content of point structure pgptPoint in (nX, nY).
File name	GPTGetPoint.c

GPTMapPoint

GPTMapPoint.c

Outline	Relative conversion of point coordinates
Syntax	void GPTMapPoint(GPoint **pgptPoint,* GRect **pgrFrom,* GRect **pgrTo*)
Argument	(i/o) GPoint *pgptPoint : point structure (i) GRect *pgrFrom : rectangle area (i) GRect *pgrTo : rectangle area
Return value	None
Details	Converts coordinates when the point structure in the pgrFrom rectangular area will be projected onto to the pgrTo rectangular area.
File name	GPTMapPoint.c

GPTMapToScreenPoint

GPTMapToScreenPoint.c

Outline	Convert point structure to screen coordinates
Syntax	void GPTMapToScreenPoint(HGDRAW *hDraw,* GPoint **pgptPoint*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i/o) GPoint *pgptPoint : point coordinates
Return value	None
Details	Converts the point defined in pgptPoint to the actual screen coordinate system.
File name	GPTMapToScreenPoint.c

GPTMapToVirtualPoint

GPTMapToVirtualPoint.c

Outline	Convert point defined in pgptPoint to virtual screen coordinate system
Syntax	void GPTMapToVirtualPoint(HGDRAW *hDraw,* GPoint ** pgptPoint*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GPoint *pgptPoint : point structure
Return value	None
Details	Converts the point defined in pgptPoint to the virtual screen coordinate system.
File name	GPTMapToVirtualPoint.c

GPTOffsetPoint

GPTOffsetPoint.c

Outline	Point structure offset processing
Syntax	void GPTOffsetPoint(GPoint **pgptPoint,* short *nDx,* short *nDy*)
Argument	(i/o) GPoint *pgptPoint : point structure for movement source (i) short nDx : increase in X direction (i) short nDy : increase in Y direction
Return value	None
Details	Moves the point structure pgptPoint nDx and nDy in the X direction and Y direction, respectively.
File name	GPTOffsetPoint.c

GPTPhysicalToVirtualPoint

GPTPhysicalToVirtualPoint.c

Outline	Convert point structure physical coordinates to virtual coordinates
Syntax	void GPTPhysicalToVirtualPoint(HGDRAW *hDraw,* GPoint **pgptPoint*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i/o) GPoint *pgptPoint : point structure
Return value	None
Details	Converts the point defined in pgptPoint from physical (local) coordinates to virtual coordinates.
File name	GPTPhysicalToVirtualPoint.c

GPTSetPoint

GPTSetPoint.c

Outline	Define point structure
Syntax	void GPTSetPoint(GPoint **pgptPoint,* short *nX,* short *nY*)
Argument	(o) GPoint *pgptPoint : point structure (i) short nX : coordinate in X direction (i) short nY : coordinate in Y direction
Return value	None
Details	Sets the (nX, nY) coordinates in the point structure pgptPoint.
File name	GPTSetPoint.c

GPTVirtualToPhysicalPoint

GPTVirtualToPhysicalPoint.c

Outline	Convert point structure virtual coordinates to physical coordinates
Syntax	void GPTVirtualToPhysicalPoint(HGDRAW *hDraw,* GPoint **pgptPoint*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i/o) GPoint *pgptPoint : point structure
Return value	None
Details	Converts the point specified by pgptPoint from virtual coordinates to physical coordinates.
File name	GPTVirtualToPhysicalPoint.c

GPGetPoly

GPGetPoly.c

Outline	Get polygon settings
Syntax	void GPGetPoly(GPoly **pgpPolygon,* short **nNumPoints,* GPoint ***pgptPoints,* GRect **pgrBounds*)
Argument	(i) GPoly *pgpPolygon : polygon structure for acquisition source (o) short *nNumPoints : number of vertices (o) GPoint **pgptPoints : point structure (o) GRect *pgrBounds : rectangle area
Return value	None
Details	Gets the polygon structure pgpPolygon content in nNumPoints, pgptPoints and pgrBounds.
File name	GPGetPoly.c

GPMapPoly

GPMapPoly.c

Outline	Enlarge/reduce polygon
Syntax	void GPMapPoly(GPoly * *pgpPolygon*, GRect * *pgrFrom*, GRect * *pgrTo*)
Argument	(i/o) GPoly *pgpPolygon : polygon structure (i) GRect *pgrFrom : rectangular area (i) GRect *pgrTo : rectangular area
Return value	None
Details	Converts coordinates when the polygon structure in the pgrFrom rectangular area will be projected onto to the pgrTo rectangular area.
File name	GPMapPoly.c

GPOffsetPoly

GPOffsetPoly.c

Outline	Polygon offset processing
Syntax	void GPOffsetPoly(GPoly **pgpPolygon,* short *nDx,* short *nDy*)
Argument	(i/o) GPoly *pgpPolygon : polygon for movement source (i) short nDx : increase in X direction (i) short nDy : increase in Y direction
Return value	None
Details	Moves the polygon defined in pgpPolygon nDx and nDy in the X direction and Y direction, respectively.
File name	GPOffsetPoly.c

GPPhysicalToVirtualPoly

GPPhysicalToVirtualPoly.c

Outline	Convert polygon physical coordinates to virtual coordinates
Syntax	void GPPhysicalToVirtualPoly(HGDRAW *hDraw,* GPoly **pgpPolygon*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i/o) GPoly *pgpPolygon : polygon structure
Return value	None
Details	Converts the polygon structure defined in pgpPolygon from physical (local) coordinates to virtual coordinates.
File name	GPPhysicalToVirtualPoly.c

GPSetPoly

GPSetPoly.c

Outline	Define polygon
Syntax	void GPSetPoly(GPoly **pgpPolygon,* short *nNumPoints,* GPoint **pgptPoints,* GRect **pgrBounds*)
Argument	(o) GPoly *pgpPolygon : polygon structure (i) short nNumPoints : number of vertices (i) GPoint *pgptPoints : point structure (i) GRect *pgrBounds : rectangle area
Return value	None
Details	Sets nNumPoints, pgptPoints and pgrBounds in the polygon structure pgpPolygon.
File name	GPSetPoly.c

GPVirtualToPhysicalPoly

GPVirtualToPhysicalPoly.c

Outline	Convert polygon virtual coordinates to physical coordinates
Syntax	void GPVirtualToPhysicalPoly(HGDRAW *hDraw,* GPoly **pgpPolygon*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i/o) GPoly *pgpPolygon : polygon structure
Return value	None
Details	Converts the polygon structure defined in pgpPolygon from virtual coordinates to local (physical) coordinates.
File name	GPVirtualToPhysicalPoly.c

GMSetMemorySpace

GMemory.c

Outline	Register memory area information
Syntax	short GMSetMemorySpace(unsigned char *ucMemorySpaceCount,* GMemorySpaceInformation **paMemorySpace*)
Argument	(i) unsigned char ucMemorySpaceCount : number of pieces of data for memory area to be registered (i) GMemorySpaceInformation *paMemorySpace : pointer to memory area information array
Return value	0 : registration failed 1 or more : number of pieces of memory area data successfully registered
Details	GMemorySpaceInformation is set in GUI library 2 for the number specified by ucMemorySpaceCount. GMemorySpaceInformation is passed to paMemorySpace as an array for the number specified by ucMemorySpaceCount. GPISetMemorySpace is called for the internal processing.
File name	GMemory.c

GMDeleteMemorySpace

GMemory.c

Outline	Delete registered memory area
Syntax	short GMDeleteMemorySpace(unsigned char *ucType,* unsigned char *ucPlane*)
Argument	(i) unsigned char ucType : memory area type to be deleted (i) unsigned char ucPlane : memory plane No. for deletion
Return value	FALSE : Deletion failed TRUE : Deletion succeeded
Details	The registered memory area specified by ucType and ucPlane is deleted. GPIDeleteMemorySpace is called for the internal processing.
File name	GMemory.c

GMMemoryAlloc

GMemory.c

Outline	Allocate memory area
Syntax	void* GMMemoryAlloc(unsigned char *ucType,* unsigned char *ucPlane,* unsigned long *ulMemorySpaceSize*)
Argument	(i) unsigned char ucType : type of memory area for memory being allocated GMEM_SPACE_GDRAW(1) : memory area used by Gdraw GMEM_SPACE_GPI(4) : memory area used by GPI GMEM_SPACE_GWIN(5) : memory area used by GWin GMEM_SPACE_GWIN_PANEL(6) : memory area used by GWin base GMEM_SPACE_GWIN_WINDOW(7) : memory area used by GWin window (i) unsigned char ucPlane : memory area plane No. for memory being allocated (i) unsigned long ulMemorySpaceSize : size of memory being allocated
Return value	NULL : memory allocation failed Other than NULL : start of address for memory allocated
Details	Allocates memory of the specified size from the specified memory area. When allocation fails, a NULL is returned as a return value. GPIMemoryAlloc is called for the internal processing.
File name	GMemory.c

GMMemoryFree

GMemory.c

Outline

Release memory area

Syntax

```
short GMMemoryFree (
  unsigned char   ucType,
  unsigned char   ucPlane,
  void* pvMemory  *pvMemory
)
```

Argument

(i) unsigned char ucType	: type of memory area for memory to be released
GMEM_SPACE_GDRAW(1)	: memory area used by Gdraw
GMEM_SPACE_GPI(4)	: memory area used by GPI
GMEM_SPACE_GWIN(5)	: memory area used by GWin
GMEM_SPACE_GWIN_PANEL(6)	: memory area used by GWin base
GMEM_SPACE_GWIN_WINDOW(7)	: memory area used by GWin window
GMEM_SPACE_USER(8)	: memory area used by user
(i) unsigned char ucPlane	: memory area plane No. for memory being released
(i) void *pvMemory	: address of memory being released

Return value

FALSE : memory release failed
TRUE : memory release succeeded

Details

Releases memory allocated from the specified memory area.
GPIMemoryFree is called for the internal processing.

File name

GMemory.c

GDCreateVram

GDCreateVram.c

Outline	Create virtual VRAM
Syntax	HGVRAM GDCreateVram(GVram *pVram*)
Argument	(i) GVram *pVram : VRAM structure
Return value	VRAM handle
Details	Creates virtual VRAM based on the information set in the VRAM structure.
File name	GDCreateVram.c

GDDeleteVram

GDDeleteVram.c

Outline	Release virtual VRAM
Syntax	void GDDeleteVram(HGVRAM *hVram*)
Argument	(i/o) HGVRAM hVram : VRAM handle
Return value	VRAM handle
Details	Releases the virtual VRAM corresponding to the specified VRAM handle.
File name	GDDeleteVram.c

GDFlushScreenForce

GDFlushScreenForce.c

Outline	Forcible refresh screen

Syntax

```
void GDFlushScreenForce(
  HGDRAW   hDraw,
  short    nX,
  short    nY,
  short    nWidth,
  short    nHeight
)
```

Argument	(i/o) HGDRAW hDraw	: drawing handle
	(i) short nX	: X coordinate for drawing starting point
	(i) short nY	: Y coordinate for drawing starting point
	(i) short nWidth	: drawing area width
	(i) short nHeight	: drawing area height

Return value	None
Details	A screen for the size of nWidth*Height is forcibly refreshed in the current coordinate system with the specified point (nX, nY) as the starting point.
File name	GDFlushScreenForce.c

GDGetGradationInfo

GDGetGradationInfo.c

Outline	Get gradation information
Syntax	void GDGetGradationInfo(HGDRAW *hDraw,* GGradationInfo **pggInfo*)
Argument	(i/o) HGDRAW hDraw : drawing handle (o) GGradationInfo *pggInfo : gradation information structure
Return value	None
Details	Gets gradation information.
File name	GDGetGradationInfo.c

GDGetLocalOrigin

GDGetLocalOrigin.c

Outline	Get local coordinate origin
Syntax	void GDGetLocalOrigin(HGDRAW *hDraw,* GPoint **pgptPoint*)
Argument	(i/o) HGDRAW hDraw : drawing handle (o) GPoint *pgptPoint : point structure
Return value	None
Details	Gets local (physical) coordinate origin.
File name	GDGetLocalOrigin.c

GDGetVram

GDGetVram.c

Outline	Get system VRAM
Syntax	HGVRAM GDGetVram(HGDRAW *hDraw*)
Argument	(i/o) HGDRAW hDraw : drawing handle
Return value	HGVRAM : system VRAM handle
Details	Gets system VRAM.
File name	GDGetVram.c

GDRemoveDrawableRegion

GDRemoveDrawableRegion.c

Outline	Delete specified rectangular area from drawable area
Syntax	short GDRemoveDrawableRegion(HGDRAW *hDraw,* GRect * *pRect*)
Argument	(i/o) HGDRAW hDraw : drawing handle (o) GRect *pRect : rectangle area to be deleted
Return value	FALSE : failed TRUE : succeeded
Details	Deletes the rectangular area specified by pRect from the drawable area held by the platform.
File name	GDRemoveDrawableRegion.c

GDSetDrawableRegion

GDSetDrawableRegion.c

Outline	Set drawable area
Syntax	short GDSetDrawableRegion(HGDRAW *hDraw,* GRect * *pRect*)
Argument	(i/o)HGDRAW hDraw : drawing handle (i)GRect *pRect : rectangular area being set
Return value	FALSE : failed TRUE : succeeded
Details	Sets the drawable area held by the platform.
File name	GDSetDrawableRegion.c

GDSetGradationInfo

GDSetGradationInfo.c

Outline	Set gradation information
Syntax	void GDSetGradationInfo(HGDRAW *hDraw,* GGradationInfo **pggInfo*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GGradationInfo *pggInfo : gradation information structure
Return value	None
Details	Sets gradation information.
File name	GDSetGradationInfo.c

GDSetLocalOrigin

GDSetLocalOrigin.c

Outline	Set local coordinate origin
Syntax	void GDSetLocalOrigin(HGDRAW *hDraw,* GPoint **pgptPoint*)
Argument	(i/o) HGDRAW hDraw : drawing handle (i) GPoint *pgptPoint : point structure
Return value	None
Details	Values for pgptPoint.nX in the X direction and pgptPoint.nY in the Y direction are set for the origin for the local (physical) coordinates.
File name	GDSetLocalOrigin.c

3. GResource

This section describes the GResource functions.

3.1 List of GResource Functions

No.	Name	Type	File name	Outline
1	GRCLoadBorder	GBorder *	GRCLoadBorder.c	Get specified ID 3D border resource
2	GRCLoadFont	GSimpleFont *	GRCLoadFont.c	Get specified ID font resource
3	GRCLoadImage	GImage *	GRCLoadImage.c	Get specified ID image resource
4	GRCLoadString	char *	GRCLoadString.c	Get specified ID character string resource
5	GRCLoadLocaleFont	GSimpleFont *	GRCLoadLocaleFont.c	Get font resource for specified locale specified ID
6	GRCLoadLocaleString	GTCHAR *	GRCLoadLocaleString.c	Get character string resource for specified locale specified ID
7	GRCLoadPalette	GPaletteHeader *	GRCLoadPalette.c	Get specified ID palette resource
8	GRCSetLocale	short	GRCLocale.c	Current locale setting
9	GRCGetLocale	unsigned char	GRCLocale.c	Get current locale
10	GRCLoadFillPattern	char *	GRCLoadFillPattern.c	Get fill pattern for specified No.
11	GRCLoadHtmlFileName	GFileName*	GRCLoadHtmlFileName.c	Get HTML file name for specified ID
12	GRCLoadLanguageStr	GTCHAR *	GRCLoadLanguageStr.c	Get specified locale language-identifying character strings resource
13	GRCLoadNowLanguageStr	GTCHAR *	GRCLoadLanguageStr.c	Get current locale language-identifying character strings resource
14	GRCLoadLinePattern	char *	GRCLoadLinePattern.c	Get line drawing pattern for specified No.
15	GRCSetResourceHandle	void	GRCResourceHandle.c	Set current resource handle
16	GRCGetResourceHandle	HGRESOURCE	GRCResourceHandle.c	Get current resource handle
17	GRCUnicodeToMultiByte	unsigned long	GRCUnicode.c	UNICODE → multibyte conversion
18	GRCMultiByteToUnicode	unsigned long	GRCUnicode.c	Multibyte → UNICODE conversion

3.2 Outlines of GResource Functions

GRCLoadBorder

GRCLoadBorder.c

Outline	Get specified ID 3D border resource
Syntax	GBorder *GRCLoadBorder(unsigned short *usResourceID*)
Argument	(i) unsigned short usResourceID : resource ID
Return value	Specified resource
Details	Gets the pointer to the 3D border resource structure with the specified ID.
File name	GRCLoadBorder.c

GRCLoadFont

GRCLoadFont.c

Outline	Get specified ID font resource
Syntax	GSimpleFont *GRCLoadFont(unsigned short *usResourceID*)
Argument	(i) unsigned short usResourceID : resource ID
Return value	Specified resource
Details	Gets the pointer to the simple font resource structure with the specified ID.
File name	GRCLoadFont.c

GRCLoadImage

GRCLoadImage.c

Outline	Get specified ID image resource
Syntax	GImage *GRCLoadImage(unsigned short *usResourceID*)
Argument	(i) unsigned short usResourceID : resource ID
Return value	Specified resource
Details	Gets the pointer to the image resource structure with the specified ID.
File name	GRCLoadImage.c

GRCLoadString

GRCLoadString.c

Outline	Get specified ID character string resource
Syntax	char *GRCLoadString(unsigned short *usResourceID*)
Argument	(i) unsigned short usResourceID : resource ID
Return value	Specified resource
Details	Gets the pointer to the character string resource structure with the specified ID.
File name	GRCLoadString.c

GRCLoadLocaleFont

GRCLoadLocaleFont.c

Outline	Get font resource for specified locale specified ID
Syntax	GSimpleFont *GRCLoadLocaleFont(unsigned char *ucLocale,* unsigned short *usResourceID*)
Argument	(i) unsigned char ucLocale : locale No. (i) unsigned short usResourceID : resource ID
Return value	Pointer for the specified resource for the specified locale
Details	Gets the pointer to the simple font resource structure with the specified locale specified ID.
File name	GRCLoadLocaleFont.c

GRCLoadLocaleString

GRCLoadLocaleString.c

Outline	Get character string resource for specified locale specified ID
Syntax	GTCHAR *GRCLoadLocaleString(unsigned char *ucLocale,* unsigned short *usResourceID*)
Argument	(i) unsigned char ucLocale : locale No. (i) unsigned short usResourceID : resource ID
Return value	Pointer for the specified resource for the specified locale
Details	Gets the pointer to character string resource with the specified ID for the specified locale.
File name	GRCLoadLocaleString.c

GRCLoadPalette

GRCLoadPalette.c

Outline	Get specified ID palette resource
Syntax	GPaletteHeader *GRCLoadPalette(unsigned short *usResourceID*)
Argument	(i) unsigned short usResourceID : resource ID
Return value	Specified resource
Details	Gets the pointer to resource structure with the specified ID.
File name	GRCLoadPalette.c

GRCSetLocale

GRCLocale.c

Outline	Current locale setting
Syntax	short GRCSetLocale(unsigned char *ucNewLocale*)
Argument	(i) unsigned char ucNewLocale : locale No. being set
Return value	TRUE : setting succeeded FALSE : setting failed
Details	Sets the current locale.
File name	GRCLocale.c

GRCGetLocale

GRCLocale.c

Outline	Get current locale
Syntax	unsigned char GRCGetLocale()
Argument	None
Return value	Current locale No.
Details	Gets the current locale.
File name	GRCLocale.c

GRCLoadFillPattern

GRCLoadFillPattern.c

Outline	Get fill pattern for specified No.
Syntax	char *GRCLoadFillPattern(unsigned short *usIndex*)
Argument	(i) unsigned short usIndex : fill No.
Return value	Fill pattern with specified No.
Details	Gets the fill pattern for the specified No.
File name	GRCLoadFillPattern.c

GRCLoadHtmlFileName

GRCLoadHtmlFileName.c

Outline	Get HTML file name for specified ID
Syntax	GFileName* GRCLoadHtmlFileName(unsigned short *usResourceID*)
Argument	(i) unsigned short usResourceID : resource ID
Return value	Specified resource HTML file name
Details	Gets the HTML file name with the specified ID.
File name	GRCLoadHtmlFileName.c

GRCLoadLanguageStr

GRCLoadLanguageStr.c

Outline	Get specified locale language-identifying character strings resource
Syntax	GTCHAR *GRCLoadLanguageStr(unsigned short *usLocaleID*)
Argument	(i) unsigned short usLocaleID : locale ID
Return value	Pointer to specified locale character string
Details	Gets the pointer for the specified locale language-identifying chracter strings resource.
File name	GRCLoadLanguageStr.c

GRCLoadNowLanguageStr

GRCLoadLanguageStr.c

Outline	Get current locale language-identifying character strings resource
Syntax	GTCHAR *GRCLoadNowLanguageStr()
Argument	None
Return value	Pointer to current locale character string
Details	Gets the pointer for the current locale language-identifying chracter strings resource.
File name	GRCLoadLanguageStr.c

GRCLoadLinePattern

GRCLoadLinePattern.c

Outline	Get line drawing pattern for specified No.
Syntax	char *GRCLoadLinePattern(unsigned short *usIndex*)
Argument	(i) unsigned short usIndex : line drawing No.
Return value	Line drawing pattern for specified No.
Details	Gets the pointer to the line drawing pattern for the specified No.
File name	GRCLoadLinePattern.c

GRCSetResourceHandle

GRCResourceHandle.c

Outline	Set current resource handle
Syntax	void GRCSetResourceHandle(HGRESOURCE *hResource*)
Argument	(i) HGRESOURCE hResource : resource handle to be set
Return value	None
Details	Sets the current resource handle.
File name	GRCResourceHandle.c

GRCGetResourceHandle

GRCResourceHandle.c

Outline	Get current resource handle
Syntax	HGRESOURCE GRCGetResourceHandle()
Argument	None
Return value	Current resource handle
Details	Gets the pointer to the current resource handle.
File name	GRCResourceHandle.c

GRCUnicodeToMultiByte

GRCUnicode.c

Outline	UNICODE → multibyte conversion
Syntax	unsigned long GRCUnicodeToMultiByte(GTCHAR *pszLangStr, char *mbstr, wchar_t *wcstr, unsigned long ulCount)
Argument	(i) GTCHAR *pszLangStr : language-identifying character string (o) char *mbstr : buffer for saving multibyte character string (i) wchar_t *wcstr : buffer for saving unicode character string (i) unsigned long ulCount : maximum size that can be stored in output destination buffer
Return value	Write size in output destination multibyte character string.
Details	Converts a unicode character string to a multibyte character string. When NULL is specified in mbstr, the size necessary for the output destination buffer is returned to the return values.
File name	GRCUnicode.c

GRCMultiByteToUnicode

GRCUnicode.c

Outline	multibyte → UNICODE conversion
Syntax	unsigned long GRCMultiByteToUnicode(GTCHAR *pszLangStr, wchar_t *wcstr, char *mbstr, unsigned long ulCount)
Argument	(i) GTCHAR *pszLangStr : language-identifying character string (o) wchar_t *wcstr : buffer storing the unicode character string (i) char *mbstr : buffer storing the multibyte character string (i) unsigned long ulCount : maximum size that can be stored in output destination buffer
Return value	Write size in output destination unicode character string.
Details	Converts a multibyte character string to a unicode character string. When NULL is specified in wcstr, the size necessary for the output destination buffer is returned to the return values.
File name	GRCUnicode.c

4. GEvent

This section describes the GEvent function group.

4.1 List of GEvent Functions

No.	Name	Type	File name	Outline
1	GEInitEvent	short	GEvent.c	Initialize event system
2	GEQuitEvent	void	GEvent.c	Process event system termination
3	GECreateEventMessage	GEventMessage	GEvent.c	Define event structure
4	GESetEvent	short	GEvent.c	Register event
5	GEGetEvent	short	GEvent.c	Acquire event
6	GEClearTimerEvent	void	GEvent.c	Clear timer event registration from event queue
7	GESetTimer	short	GTimer.c	Register timer event
8	GEKillTimer	void	GTimer.c	Clear timer event registration
9	GECheckTimer	void	GTimer.c	Confirm that timer event has been issued
10	GESetSystemTime	short	GTimer.c	Set system time
11	GEGetSystemTime	short	GTimer.c	Get system time
12	GECompareSystemTime	long	GTimer.c	System time comparison
13	GEClearTargetEvent	void	GEvent.c	Clear registration for specified objects event to be sent

4.2 Outline of GEvent Functions

GEInitEvent

GEvent.c

Outline	Initialize event system
Syntax	short GEInitEvent()
Argument	None
Return value	FALSE : initialization failed TRUE : initialization succeeded
Details	Initializes the event system.
File name	GEvent.c

GEQuitEvent

GEvent.c

Outline	Process event system termination
Syntax	void GEQuitEvent()
Argument	None
Return value	None
Details	Quits event system.
File name	GEvent.c

GECreateEventMessage

GEvent.c

Outline	Define event structure

Syntax

```
GEventMessage GECreateEventMessage(
  unsigned short  usMessage,
  void            *pvTarget,
  long            lLParam,
  long            lUParam
)
```

Argument		
	(i)unsigned short usMessage	: event message
	(i)void *pvTarget	: event target object
	(i)long lLParam	: lower-side of additional information
	(i)long lUParam	: upper-side of additional information

Return value	Event structure
Details	Defines event structure.
File name	GEvent.c

GESetEvent

GEvent.c

Outline	Register event
Syntax	short GESetEvent(GEventMessage *gemMsg,* short *fMarge*)
Argument	(i)GEventMessage gemMsg : Event information (i)short fMarge : merge flag
Return value	FALSE : registration failed TRUE : registration succeeded
Details	Registers an event. When there is a same event for the merge flag, this makes the setting either for merging or not for merging. fMarge FALSE : no merge TRUE : merge
File name	GEvent.c

GEGetEvent

GEvent.c

Outline	Acquire event
Syntax	short GEGetEvent(GEventMessage *gemMsg, long lTimeOut)
Argument	(i)GEventMessage *gemMsg : event information (i)long lTimeOut : timeout
Return value	FALSE : no event TRUE : event present
Details	Gets an event. When the event is not generated until the timeout time passes, the status is changed to the waiting.
File name	GEvent.c

GEClearTimerEvent

GEvent.c

Outline	Clear timer event registration from even queue
Syntax	void GEClearTimerEvent(void *pvTarget, unsigned short usTimerID)
Argument	(i) void * pvTarget : timer event target object (i) unsigned short usTimerID : timer ID
Return value	None
Details	Clears the registration of the timer event for the ID specified from the event queue. Specify the object where the timer event is set in pvTarget.
File name	GEvent.c

GESetTimer

GTimer.c

Outline	Register timer event
Syntax	short GESetTimer(void **pvTarget,* unsigned short *usTimerID,* unsigned long *ulTimeOut*)
Argument	(i)void *pvTarget : timer event target object (i)unsigned short usTimerID : timer ID (i)unsigned long ulTimeOut : timer interval
Return value	FALSE : timer event registration failed TRUE : timer event registration succeeded
Details	Registers a timer event.
File name	GTimer.c

GEKillTimer

GTimer.c

Outline	Clear timer event registration
Syntax	void GEKillTimer(void **pvTarget,* unsigned short *usTimerID*)
Argument	(i)void *pvTarget : timer event target object (i)unsigned short usTimerID : timer ID to be cleared
Return value	None
Details	Clears the timer event registration.
File name	GTimer.c

GECheckTimer

GTimer.c

Outline	Confirm that timer event has been issued
Syntax	void GECheckTimer()
Argument	None
Return value	None
Details	Checks the issuance of a timer event. If the timer event being issued is present, the event is registered.
File name	GTimer.c

GESetSystemTime

GTimer.c

Outline	Set system time
Syntax	short GESetSystemTime(GSystemTime *pSystemTime*)
Argument	GSystemTime *pSystemTime : system time
Return value	FALSE : setting failed TRUE : setting succeeded
Details	Sets the current system time.
File name	GTimer.c

GEGetSystemTime

GTimer.c

Outline	Get system time
Syntax	short GEGetSystemTime(GSystemTime *pSystemTime*)
Argument	GSystemTime *pSystemTime : for system time storage
Return value	FALSE : acquisition failed TRUE : acquisition succeeded
Details	Gets the current system time.
File name	GTimer.c

GECompareSystemTime

GTimer.c

Outline	System time comparison
Syntax	long GECompareSystemTime(GSystemTime *pSystemTime1,* GSystemTime *pSystemTime2*)
Argument	GSystemTime *pSystemTime1 : system time GSystemTime *pSystemTime2 : system time
Return value	Difference in system times
Details	Compares the two times.
File name	GTimer.c

GEClearTargetEvent

GEvent.c

Outline	Clear specified target event registration
Syntax	void GEClearTargetEvent(void * pvTarget)
Argument	(i) void * pvTarget : target object
Return value	None
Details	Clears the target event registration set specified in pvTarget.
File name	GEvent.c

5. GCS

This section describes the GCS function group.

5.1 List of GCS Functions

No.	Name	Type	File name	Outline
1	GCSGetType	unsigned short	gwin.h	Get object type
2	GCSGetID	unsigned short	gwin.h	Get object ID
3	GCSGetXPosition	short	gwin.h	Get horizontal position
4	GCSGetYPosition	short	gwin.h	Get vertical position
5	GCSSetPosition	void	gwin.h	Set position
6	GCSGetPosition	void	gwin.h	Get position
7	GCSGetWidth	short	gwin.h	Get width
8	GCSGetHeight	short	gwin.h	Get height
9	GCSGetSize	void	gwin.h	Get size
10	GCSSetBounds	void	gwin.h	Set relative position within parent object
11	GCSGetBounds	void	gwin.h	Get relative position within parent object
12	GCSCalcClientRect	void	gwin.h	Get coordinate position of client area
13	GCSGetClientRect	void	gwin.h	Get object client area
14	GCSClientToScreen	void	gwin.h	Convert client coordinates to screen coordinates
15	GCSGetStyle	unsigned long	gwin.h	Batch style acquisition
16	GCSSetVisibleStatus	void	gwin.h	Set display/non-display status
17	GCSGetVisibleStatus	short	gwin.h	Get display/non-display status
18	GCSSetEnableStatus	void	gwin.h	Set controllable or uncontrollable status
19	GCSGetEnableStatus	short	gwin.h	Get controllable or uncontrollable status
20	GCSSetConsumableStatus	void	gwin.h	Set input event consumable or input event not consumable status
21	GCSGetConsumableStatus	short	gwin.h	Get input event consumable or input event not consumable status
22	GCSSetCaptionStatus	void	gwin.h	Set caption display status
23	GCSGetCaptionStatus	short	gwin.h	Get caption display status
24	GCSSetFocusStatus	void	gwin.h	Set focus status
25	GCSGetFocusStatus	short	gwin.h	Get focus status

No.	Name	Type	File name	Outline
26	GCSSetPanelStatus	void	gwin.h	Set management panel object status
27	GCSGetPanelStatus	short	gwin.h	Get management panel object status
28	GCSSetStyleStatus	void	gwin.h	Set style
29	GCSGetStyleStatus	short	gwin.h	Get style
30	GCSSetParent	void	gwin.h	Set parent object
31	GCSGetParent	GBaseObject*	gwin.h	Get parent object
32	GCSImportProperty	short	gwin.h	Import property settings
33	GCSExportProperty	void	gwin.h	Export property settings
34	GCSCreate	short	gwin.h	Create process
35	GCSCreateVram	short	gwin.h	Create process (create VRAM)
36	GCSDelete	void	gwin.h	Delete process
37	GCSPaint	void	gwin.h	Drawing process
38	GCSOnDraw	void	gwin.h	Client area drawing process
39	GCSOnNcDraw	void	gwin.h	Non-client area drawing process
40	GCSAddRefreshRect	void	gwin.h	Client area redraw area registration
41	GCSLButtonPress	long	gwin.h	Process for pressing left mouse button
42	GCSOnLButtonPress	long	gwin.h	Client area process for pressing left mouse button
43	GCSOnNcLButtonPress	long	gwin.h	Non-client area process for pressing left mouse button
44	GCSLButtonRelease	void	gwin.h	Process for releasing left mouse button
45	GCSOnLButtonRelease	void	gwin.h	Client area process for releasing left mouse button
46	GCSOnNcLButtonRelease	void	gwin.h	Non-client area process for releasing left mouse button
47	GCSMouseMove	void	gwin.h	Pointing device movement process
48	GCSOnMouseMove	void	gwin.h	Pointing device client area movement process
49	GCSOnNcMouseMove	void	gwin.h	Pointing device non-client area movement process
50	GCSKeyPress	long	gwin.h	Process for pressing key
51	GCSKeyRelease	void	gwin.h	Process for releasing key
52	GCSSetFocus	void	gwin.h	Focus setting process
53	GCSKillFocus	void	gwin.h	Focus removing process

No.	Name	Type	File name	Outline
54	GCSTimer	void	gwin.h	Timer process
55	GCSUser	long	gwin.h	User process
56	GCSSetFocusInformation	void	gwin.h	Set focus movement information
57	GCSGetFocusInformation	GFocusInformation *	gwin.h	Get focus movement information
58	GCSChar	void	gwin.h	Character input process
59	GCSClose	void	gwin.h	Issue object close process request
60	GCSDeleteChild	void	gwin.h	Child object deletion process
61	GCSSetMemorySpace	void	gwin.h	Set memory information
62	GCSGetMemorySpace	void	gwin.h	Get memory information
63	GCSSetDrawForceStatus	void	gwin.h	Set forced drawing status
64	GCSGetDrawForceStatus	short	gwin.h	Get forced drawing status
65	GCSSetRedrawStatus	void	gwin.h	Set redraw status
66	GCSGetRedrawStatus	short	gwin.h	Get redraw status
67	GCSClearDrawStatus	void	gwin.h	Clear drawing status
68	GCSAddDirtyRect	void	gwin.h	Client area redraw area registration (internal processing function)
69	GCSSetPrevTimerObject	void	gwin.h	Set previous object on timer bidirectional list
70	GCSGetPrevTimerObject	GBaseObject*	gwin.h	Get previous object on timer bidirectional list
71	GCSSetNextTimerObject	void	gwin.h	Set next object on timer bidirectional list
72	GCSGetNextTimerObject	GBaseObject*	gwin.h	Get next object on timer bidirectional list
73	GCSSetBlinkStatus	void	gwin.h	Set blink status
74	GCSGetBlinkStatus	unsigned char	gwin.h	Get blink status
75	GCSGetParentPanel	void	gwin.h	Get management panel object
76	GCSSetPrevBlinkControl	void	gwin.h	Set previous object on blink bidirectional list
77	GCSGetPrevBlinkControl	GBaseObject*	gwin.h	Get previous object on blink bidirectional list
78	GCSSetNextBlinkControl	void	gwin.h	Set next object on blink bidirectional list
79	GCSGetNextBlinkControl	GBaseObject*	gwin.h	Get next object on blink bidirectional list

No.	Name	Type	File name	Outline
80	GCSSetPrevScrollControl	void	gwin.h	Set previous object on scroll bidirectional list
81	GCSGetPrevScrollControl	GBaseObject*	gwin.h	Get previous object on scroll bidirectional list
82	GCSSetNextScrollControl	void	gwin.h	Set next object on scroll bidirectional list
83	GCSGetNextScrollControl	GBaseObject*	gwin.h	Get next object on scroll bidirectional list
84	GCSLoadData	int	gwin.h	Load interpreter screen file
85	GCSSetHWDCEnvironment	void	gwin.h	Set drawing environment
86	GCSGetScreen	GBaseObject*	gwin.h	Get screen object
87	GCSSetGBaseObject	void	gwin.h	Set object management structure
88	GCSGetGBaseObject	GBaseObject*	gwin.h	Get object management structure
89	GCSFreeLoadData	int	gwin.h	Release memory acquired by LoadData
90	GCSSetModalStatus	void	gwin.h	Set modal status
91	GCSGetModalStatus	short	gwin.h	Get modal window status
92	GCSGetRetValue	short	gwin.h	Get modal window return value
93	GCSBeginModal	void	gwin.h	Start process for modal window
94	GCSEndModal	void	gwin.h	End process for modal window
95	GCSSetTitleBarStatus	void	gwin.h	Set title bar present/none status
96	GCSGetTitleBarStatus	short	gwin.h	Get title bar present/none status
97	GCSSetCloseButtonStatus	void	gwin.h	Set close button present/none status
98	GCSGetCloseButtonStatus	short	gwin.h	Get close button present/none status
99	GCSSetEdgeStatus	void	gwin.h	Set window frame present/none status
100	GCSGetEdgeStatus	short	gwin.h	Get window frame present/none status
101	GCSAddChild	short	gwin.h	Add child object
102	GCSGetChildList	HGBWL	gwin.h	Get child object list
103	GCSSetBackGroundDesign	void	gwin.h	Set background design
104	GCSGetBackGroundDesign	GDesign*	gwin.h	Get background design
105	GCSCreateChildren	short	gwin.h	Process for creating child object
106	GCSDeleteAllChildren	void	gwin.h	Delete all child objects
107	GCSGetChild	GBaseObject *	gwin.h	Search child object

No.	Name	Type	File name	Outline
108	GCSGetChildAll	GBaseObject *	gwin.h	Search child object search in frame
109	GCSSetActiveChild	short	gwin.h	Set active object
110	GCSGetActiveChild	GBaseObject*	gwin.h	Get active object (only direct child object)
111	GCSGetActiveControl	GBaseObject*	gwin.h	Get active control (including terminal child objects)
112	GCSSetBlinkOffInterval	void	gwin.h	Set blinking OFF interval
113	GCSGetBlinkOffInterval	long	gwin.h	Get blinking OFF interval
114	GCSSetBlinkOnInterval	void	gwin.h	Set blinking ON interval
115	GCSGetBlinkOnInterval	long	gwin.h	Get blinking ON interval
116	GCSAddBlinkControl	void	gwin.h	Add caption blink control
117	GCSRemoveBlinkControl	void	gwin.h	Delete caption blink control
118	GCSAddScrollControl	void	gwin.h	Add caption scroll control
119	GCSRemoveScrollControl	void	gwin.h	Delete caption scroll control
120	GCSDrawChildren	void	gwin.h	Child object drawing process
121	GCSAttachPanel	GBaseObject *	gwin.h	Attach panel object
122	GCSGetPanel	GBaseObject *	gwin.h	Get attached GCPanel object
123	GCSSetSystemFontID	void	gwin.h	Set system font resource ID
124	GCSGetSystemFontID	unsigned short	gwin.h	Get system font resource ID
125	GCSSetSystemBorderID	void	gwin.h	Set system 3D border resource ID
126	GCSGetSystemBorderID	unsigned short	gwin.h	Get system 3D border resource ID
127	GCSSetSystemForeColor	void	gwin.h	Set system foreground color
128	GCSGetSystemForeColor	GColor	gwin.h	Get system foreground color
129	GCSSetSystemBackColor	void	gwin.h	Set system background color
130	GCSGetSystemBackColor	GColor	gwin.h	Get system background color
131	GCSSetSystemActiveColor	void	gwin.h	Set system active color
132	GCSGetSystemActiveColor	GColor	gwin.h	Get system active color
133	GCSSetSystemInactiveColor	void	gwin.h	Set system inactive color
134	GCSGetSystemInactiveColor	GColor	gwin.h	Get system inactive color
135	GCSGetFrame	GBaseObject *	gwin.h	Get frame
136	GCSAddWindow	short	gwin.h	Add window
137	GCSChangeFocus	void	gwin.h	Change focus
138	GCSGetFocus	GBaseObject *	gwin.h	Get focus
139	GCSDispatchMessage	long	gwin.h	Event dispatch process
140	GCSGetWDC	HWDC	gwin.h	Get drawing context
141	GCSGetHWDC	HWDC	gwin.h	Get drawing context
142	GCSSetPanelPlane	void	gwin.h	Set memory area panel plane management information

No.	Name	Type	File name	Outline
143	GCSGetPanelPlane	unsigned char	gwin.h	Get memory area panel plane management information
144	GCSSetWindowPlane	void	gwin.h	Set memory area window plane management information
145	GCSGetWindowPlane	unsigned char	gwin.h	Get memory area window plane management information
146	GCSSetExitStatus	void	gwin.h	Set exit information
147	GCSGetExitStatus	short	gwin.h	Get exit information
148	GCSSetSystemStatus	void	gwin.h	Set system information
149	GCSGetSystemStatus	short	gwin.h	Get system information
150	GCSDoModal	short	gwin.h	Window modal display
151	GCSSetPaletteStatus	void	gwin.h	Set system palette utilization status
152	GCSGetPaletteStatus	short	gwin.h	Get system palette utilization status
153	GCSSetSystemPaletteID	void	gwin.h	Set system palette resource ID
154	GCSGetSystemPaletteID	unsigned short	gwin.h	Get system palette resource ID
155	GCSSetPaletteColors	void	gwin.h	Set number of colors in system palette
156	GCSGetPaletteColors	unsigned short	gwin.h	Get number of colors in system palette
157	GCSAddTimerObject	void	gwin.h	Add object to system timer list
158	GCSRemoveTimerObject	void	gwin.h	Delete object from system timer list
159	GCSGetWindow	GBaseObject*	gwin.h	Get window
160	GCSGetActiveWindow	GBaseObject*	gwin.h	Get active window
161	GCSShowPanel	short	gwin.h	Display panel switching process
162	GCSCreateInstance	GBaseObject *	gwin.h	Create instance process
163	GCSPageLogOperation	short	gwin.h	Switch screen according to screen history
164	GCSDrawFrame	void	gwin.h	Related panel drawing process
165	GCSDrawPanel	void	gwin.h	Related GCPanel object drawing process
166	GCSOnDrawWindow	void	gwin.h	Window drawing process
167	GCSSetSystemPalette	void	gwin.h	Set system palette
168	GCSDrawBackGround	void	gwin.h	Background drawing process
169	GCSCreateGWindow	short	gwin.h	Create window
170	GCSSetScreenManager	void	gwin.h	Set screen manager
171	GCSSetModuleID	void	gwin.h	Set module ID
172	GCSClearPageLog	void	gwin.h	Clear screen history
173	GCSAddPageLog	short	gwin.h	Add screen history

No.	Name	Type	File name	Outline
174	GCSPrevPage	short	gwin.h	Issue request for displaying previous screen
175	GCSNextPage	short	gwin.h	Issue request for displaying next screen
176	GCSSetPageOffset	void	gwin.h	Set module page offset
177	GCSGetPageOffset	long	gwin.h	Get module page offset
178	GCSGetOperationLog	short	gwin.h	Get operation log
179	GCSClearOperationLog	short	gwin.h	Clear operation log
180	GCSAddOperationLog	short	gwin.h	Add operation log (operation event)
181	GCSAddChangePageLog	short	gwin.h	Add operation log (screen switching)
182	GCSGetPageNumber	long	gwin.h	Get screen No.
183	GCSLoadProjectFile	int	gwin.h	Load interpreter project file
184	GCSLoadGWindow	GBaseObject*	gwin.h	Window loading process
185	GCSModalGWindow	short	gwin.h	Modal window process
186	GCSSetBlinkType	void	gwin.h	Set blink type
187	GCSGetBlinkType	unsigned char	gwin.h	Get blink type
188	GCSSetBlinkEnableStatus	void	gwin.h	Set blink operation
189	GCSGetBlinkEnableStatus	unsigned char	gwin.h	Get blink operation
190	GCSStartCaptionBlink	void	gwin.h	Start blink operation
191	GCSStopCaptionBlink	void	gwin.h	Stop blink operation
192	GCSSetCaptionScrollEnable	void	gwin.h	Set scroll operation
193	GCSGetCaptionScrollEnable	unsigned char	gwin.h	Get scroll operation
194	GCSStartCaptionScroll	void	gwin.h	Start scroll operation
195	GCSStopCaptionScroll	void	gwin.h	Stop scroll operation
196	GCSPauseCaptionScroll	void	gwin.h	Pause scroll operation
197	GCSRestartCaptionScroll	void	gwin.h	Restart scroll operation
198	GCSSetCaptionScrollStatus	void	gwin.h	Set scroll operation status
199	GCSGetCaptionScrollStatus	unsigned char	gwin.h	Get scroll operation status
200	GCSIsCaptionOutOfBounds	short	gwin.h	Confirm overflowing of caption character string
201	GCSSetCaptionScrollDelayTime	void	gwin.h	Set scroll delay time
202	GCSGetCaptionScrollDelayTime	unsigned long	gwin.h	Get scroll delay time
203	GCSSetCaptionScrollRefreshTime	void	gwin.h	Set scroll refresh time
204	GCSGetCaptionScrollRefreshTime	unsigned long	gwin.h	Get scroll refresh time
205	GCSSetCaptionScrollMovementValue	void	gwin.h	Set scroll movement value
206	GCSGetCaptionScrollMovementValue	unsigned short	gwin.h	Get scroll movement value
207	GCSSetCaptionScrollStartPosition	void	gwin.h	Set scroll start position
208	GCSGetCaptionScrollStartPosition	unsigned char	gwin.h	Get scroll start position
209	GCSSetCaptionScrollPosition	void	gwin.h	Set total scroll movement value

No.	Name	Type	File name	Outline
210	GCSGetCaptionScrollPosition	long	gwin.h	Get total scroll movement value
211	GCSUpdateScroll	void	gwin.h	Scroll update process
212	GCSIsCaptionScrollFinish	short	gwin.h	Confirm caption scroll completion
213	GCSCallBackMessage	long	gwin.h	Call back message execution process
214	GCSLoadScreenFile	int	gwin.h	Load interpreter screen file
215	GCSSetPenColor	void	gwin.h	Set line color
216	GCSGetPenColor	GColor	gwin.h	Get line color
217	GCSSetPenDash	void	gwin.h	Set line type
218	GCSGetPenDash	unsigned short	gwin.h	Get line type
219	GCSSetPenSize	void	gwin.h	Set line width
220	GCSGetPenSize	unsigned short	gwin.h	Get line width
221	GCSSetAngle	void	gwin.h	Set start angle and finish angle for sector and arc
222	GCSGetAngle	void	gwin.h	Get start angle and finish angle for sector and arc
223	GCSSetBrush	void	gwin.h	Set fill brush
224	GCSGetBrush	GBrush*	gwin.h	Get fill brush
225	GCSSetStartPoint	void	gwin.h	Set line drawing starting point
226	GCSGetStartPoint	void	gwin.h	Get line drawing starting point
227	GCSSetEndPoint	void	gwin.h	Set line drawing end point
228	GCSGetEndPoint	void	gwin.h	Get line drawing end point
229	GCSSetPoly	short	gwin.h	Set polygon/continuous line
230	GCSGetPoly	GPoly*	gwin.h	Get polygon/continuous line
231	GCSSetDrawAttribute	void	gwin.h	Set drawing attributes for figure
232	GCSButtonSetAction	void	gwin.h	Set button operations
233	GCSButtonGetAction	unsigned char	gwin.h	Get button operations
234	GCSButtonSetDisplay	void	gwin.h	Set button display
235	GCSButtonGetDisplay	unsigned char	gwin.h	Get button display
236	GCSButtonSetBorderID	void	gwin.h	Set 3D border resource ID
237	GCSButtonGetBorderID	unsigned short	gwin.h	Get 3D border resource ID
238	GCSButtonSetOnDesign	void	gwin.h	Set ON status design
239	GCSButtonGetOnDesign	GDesign*	gwin.h	Get ON status design
240	GCSButtonSetOffDesign	void	gwin.h	Set OFF status design
241	GCSButtonGetOffDesign	GDesign*	gwin.h	Get OFF status design
242	GCSButtonSetFocusDesign	void	gwin.h	Set FOCUS status design
243	GCSButtonGetFocusDesign	GDesign*	gwin.h	Get FOCUS status design
244	GCSButtonSetDisableDesign	void	gwin.h	Set Disable status design
245	GCSButtonGetDisableDesign	GDesign*	gwin.h	Get Disable status design
246	GCSButtonSetFontID	void	gwin.h	Set font resource ID

No.	Name	Type	File name	Outline
247	GCSButtonGetFontID	unsigned short	gwin.h	Get font resource ID
248	GCSButtonSetStringID	void	gwin.h	Set caption character string resource ID
249	GCSButtonGetStringID	unsigned short	gwin.h	Get caption character string resource ID
250	GCSButtonSetCaption	void	gwin.h	Set caption information
251	GCSButtonGetCaption	GCaption *	gwin.h	Get caption information
252	GCSButtonSetFocusEffect	void	gwin.h	Set effect during focus
253	GCSButtonGetFocusEffect	unsigned char	gwin.h	Get effect during focus
254	GCSButtonSetStatus	void	gwin.h	Set object status
255	GCSButtonGetStatus	unsigned char	gwin.h	Get object status
256	GCSCheckboxSetBorderID	void	gwin.h	Set 3D border resource ID
257	GCSCheckboxGetBorderID	unsigned short	gwin.h	Get 3D border resource ID
258	GCSCheckboxSetFontID	void	gwin.h	Set font resource ID
259	GCSCheckboxGetFontID	unsigned short	gwin.h	Get font resource ID
260	GCSCheckboxSetStringID	void	gwin.h	Set caption character string resource ID
261	GCSCheckboxGetStringID	unsigned short	gwin.h	Get caption character string resource ID
262	GCSCheckboxSetCaption	void	gwin.h	Set caption information
263	GCSCheckboxGetCaption	GCaption *	gwin.h	Get caption information
264	GCSCheckboxSetFocusEffect	void	gwin.h	Set effect during focus
265	GCSCheckboxGetFocusEffect	unsigned char	gwin.h	Get effect during focus
266	GCSCheckboxSetStatus	void	gwin.h	Set object status
267	GCSCheckboxGetStatus	unsigned char	gwin.h	Get object status
268	GCSCheckboxSetBoxSize	void	gwin.h	Set box size
269	GCSCheckboxGetBoxSize	unsigned short	gwin.h	Get box size
270	GCSCheckboxSetBoxColor	void	gwin.h	Set box color
271	GCSCheckboxGetBoxColor	GColor	gwin.h	Get box color
272	GCSCheckboxSetFocusColor	void	gwin.h	Set background color during focus
273	GCSCheckboxGetFocusColor	GColor	gwin.h	Get background color during focus
274	GCSCheckboxSetDisableCaptionColor	void	gwin.h	Set character color when disabled
275	GCSCheckboxGetDisableCaptionColor	GColor	gwin.h	Get character color when disabled
276	GCSCheckboxSetDisableBoxColor	void	gwin.h	Set box color when disabled
277	GCSCheckboxGetDisableBoxColor	GColor	gwin.h	Get box color when disabled
278	GCSEditSetBorderID	void	gwin.h	Set 3D border resource ID
279	GCSEditGetBorderID	unsigned short	gwin.h	Get 3D border resource ID

No.	Name	Type	File name	Outline
280	GCSEditSetFontID	void	gwin.h	Set font resource ID
281	GCSEditGetFontID	unsigned short	gwin.h	Get font resource ID
282	GCSEditAddString	short	gwin.h	Add character string
283	GCSEditSetString	short	gwin.h	Set display character string
284	GCSEditGetString	short	gwin.h	Get display character string
285	GCSEditSetTextColor	void	gwin.h	Set character display color
286	GCSEditGetTextColor	GColor	gwin.h	Get character display color
287	GCSEditSetBackColor	void	gwin.h	Set background color
288	GCSEditGetBackColor	GColor	gwin.h	Get background color
289	GCSEditSetBrush	void	gwin.h	Set brush
290	GCSEditDeleteString	void	gwin.h	Delete character string
291	GCSEditAddLine	short	gwin.h	Add line
292	GCSEditInsertLine	short	gwin.h	Insert line
293	GCSEditDeleteLine	short	gwin.h	Delete line
294	GCSEditGetLineString	short	gwin.h	Get line character string
295	GCSEditSetLineFeedCode	void	gwin.h	Set line feed character string
296	GCSEditGetLineFeedCode	unsigned char	gwin.h	Get line feed character string
297	GCSEditSetCursor	short	gwin.h	Set cursor position
298	GCSEditGetCursor	short	gwin.h	Get cursor position
299	GCSEditSetForeColor	void	gwin.h	Set foreground color
300	GCSEditGetForeColor	GColor	gwin.h	Get foreground color
301	GCSEditSetFillPattern	void	gwin.h	Set fill pattern
302	GCSEditGetFillPattern	short	gwin.h	Get fill pattern
303	GCSEditSetInsertMode	void	gwin.h	Set insert/overwrite mode
304	GCSEditGetInsertMode	short	gwin.h	Get insert/overwrite mode
305	GCSEditSetLineBrush	short	gwin.h	Set line brush
306	GCSEditGetLineBrush	short	gwin.h	Get line brush
307	GCSEditSetLineTextColor	short	gwin.h	Set line text display color
308	GCSEditGetLineTextColor	GColor	gwin.h	Get line text display color
309	GCSEditGetLength	unsigned long	gwin.h	Get character length
310	GCSEditFind	short	gwin.h	Search forward
311	GCSEditReverseFind	short	gwin.h	Search backward
312	GCSEditReplace	short	gwin.h	Replace
313	GCSEditReplaceAll	unsigned long	gwin.h	Replace all
314	GCSEditInsertString	short	gwin.h	Insert character string
315	GCSEditGetLineCount	unsigned long	gwin.h	Get number of lines
316	GCSEditGetBrush	short	gwin.h	Get brush
317	GCSEditRemoveString	short	gwin.h	Delete character string
318	GCSHtmlbrowserSetBorderID	void	gwin.h	Set 3D border resource ID
319	GCSHtmlbrowserGetBorderID	unsigned short	gwin.h	Get 3D border resource ID

No.	Name	Type	File name	Outline
320	GCSHtmlbrowserSetFontID	void	gwin.h	Set font resource ID
321	GCSHtmlbrowserGetFontID	unsigned short	gwin.h	Get font resource ID
322	GCSHtmlbrowserSetScrollBarWidth	void	gwin.h	Set scroll bar width
323	GCSHtmlbrowserGetScrollBarWidth	unsigned char	gwin.h	Get scroll bar width
324	GCSHtmlbrowserSetTextColor	void	gwin.h	Set text display color
325	GCSHtmlbrowserGetTextColor	GColor	gwin.h	Get text display color
326	GCSHtmlbrowserSetBackColor	void	gwin.h	Set background color
327	GCSHtmlbrowserGetBackColor	GColor	gwin.h	Get background color
328	GCSHtmlbrowserSetLinkColor	void	gwin.h	Set link color
329	GCSHtmlbrowserGetLinkColor	GColor	gwin.h	Get link color
330	GCSHtmlbrowserSetHtmlFileName	void	gwin.h	Set HTML file name
331	GCSHtmlbrowserGetHtmlFileName	GTCHAR*	gwin.h	Get HTML file name
332	GCSLabelSetFontID	void	gwin.h	Set font resource ID
333	GCSLabelGetFontID	unsigned short	gwin.h	Get font resource ID
334	GCSLabelSetStringID	void	gwin.h	Set caption character string resource ID
335	GCSLabelGetStringID	unsigned short	gwin.h	Get caption character string resource ID
336	GCSLabelSetCaption	void	gwin.h	Set caption information
337	GCSLabelGetCaption	GCaption *	gwin.h	Get caption information
338	GCSListSetBorderID	void	gwin.h	Set 3D border resource ID
339	GCSListGetBorderID	unsigned short	gwin.h	Get 3D border resource ID
340	GCSListSetFontID	void	gwin.h	Set font resource ID
341	GCSListGetFontID	unsigned short	gwin.h	Get font resource ID
342	GCSListSetCaption	void	gwin.h	Set caption information
343	GCSListGetCaption	GCaption *	gwin.h	Get caption information
344	GCSListSetFocusColor	void	gwin.h	Set background color during focus
345	GCSListGetFocusColor	GColor	gwin.h	Get background color during focus
346	GCSListSetScrollButtonColor	void	gwin.h	Set scroll button color
347	GCSListGetScrollButtonColor	GColor	gwin.h	Get scroll button color
348	GCSListSetScrollBarColor	void	gwin.h	Set scroll bar color
349	GCSListGetScrollBarColor	GColor	gwin.h	Get scroll bar color
350	GCSListSetNormalColor	void	gwin.h	Set normal background color
351	GCSListGetNormalColor	GColor	gwin.h	Get normal background color
352	GCSListSetDisableColor	void	gwin.h	Set background color when disabled
353	GCSListGetDisableColor	GColor	gwin.h	Get background color when disabled

No.	Name	Type	File name	Outline
354	GCSListSetScrollBarWidth	void	gwin.h	Set scroll bar width
355	GCSListGetScrollBarWidth	unsigned char	gwin.h	Get scroll bar width
356	GCSListSetSelectBarColor	void	gwin.h	Set selection bar color
357	GCSListGetSelectBarColor	GColor	gwin.h	Get selection bar color
358	GCSListSetMaxListLines	void	gwin.h	Set maximum number of lines in list
359	GCSListGetMaxListLines	unsigned short	gwin.h	Get maximum number of lines in list
360	GCSListAddString	short	gwin.h	Add list character string
361	GCSListGetListString	short	gwin.h	Get list character string
362	GCSListRemoveAllStrings	void	gwin.h	Delete all list character strings
363	GCSListSetCurrentSelect	short	gwin.h	Set selected line
364	GCSListGetCurrentSelect	long	gwin.h	Get selected line
365	GCSListSetVisibleScrollBarStatus	void	gwin.h	Set scroll bar display/non-display status
366	GCSListGetVisibleScrollBarStatus	short	gwin.h	Get scroll bar display/non-display status
367	GCSListSetEnableScrollBarStatus	void	gwin.h	Set scroll bar controllable or uncontrollable status
368	GCSListGetEnableScrollBarStatus	short	gwin.h	Get scroll bar controllable or uncontrollable status
369	GCSListSetTopLine	void	gwin.h	Set list top line
370	GCSListGetTopLine	long	gwin.h	Get list top line
371	GCSListInsertString	short	gwin.h	Insert list character string
372	GCSListGetLineCount	short	gwin.h	Get number of lines in list
373	GCSListRemoveString	short	gwin.h	Delete list character string
374	GCSProgressbarSetBackGroundDesign	void	gwin.h	Set background design
375	GCSProgressbarGetBackGroundDesign	GDesign*	gwin.h	Get background design
376	GCSProgressbarSetBarDesign	void	gwin.h	Set bar design
377	GCSProgressbarGetBarDesign	GDesign*	gwin.h	Get bar design
378	GCSProgressbarSetRange	void	gwin.h	Set progress bar range
379	GCSProgressbarGetRange	void	gwin.h	Get progress bar range
380	GCSProgressbarSetDirection	void	gwin.h	Set fill direction
381	GCSProgressbarGetDirection	unsigned char	gwin.h	Get fill direction
382	GCSProgressbarGetValue	short	gwin.h	Get current progress bar value
383	GCSProgressbarSetValue	void	gwin.h	Set current progress bar value
384	GCSRadiobuttonSetBorderID	void	gwin.h	Set 3D border resource ID
385	GCSRadiobuttonGetBorderID	unsigned short	gwin.h	Get 3D border resource ID
386	GCSRadiobuttonSetFontID	void	gwin.h	Set font resource ID
387	GCSRadiobuttonGetFontID	unsigned short	gwin.h	Get font resource ID

No.	Name	Type	File name	Outline
388	GCSRadiobuttonSetStringID	void	gwin.h	Set caption character string resource ID
389	GCSRadiobuttonGetStringID	unsigned short	gwin.h	Get caption character string resource ID
390	GCSRadiobuttonSetCaption	void	gwin.h	Set caption information
391	GCSRadiobuttonGetCaption	GCaption *	gwin.h	Get caption information
392	GCSRadiobuttonSetFocusEffect	void	gwin.h	Set effect during focus
393	GCSRadiobuttonGetFocusEffect	unsigned char	gwin.h	Get effect during focus
394	GCSRadiobuttonSetStatus	void	gwin.h	Set object status
395	GCSRadiobuttonGetStatus	unsigned char	gwin.h	Get object status
396	GCSRadiobuttonSetBoxSize	void	gwin.h	Set box size
397	GCSRadiobuttonGetBoxSize	unsigned short	gwin.h	Get box size
398	GCSRadiobuttonSetBoxColor	void	gwin.h	Set box color
399	GCSRadiobuttonGetBoxColor	GColor	gwin.h	Get box color
400	GCSRadiobuttonSetFocusColor	void	gwin.h	Set background color during focus
401	GCSRadiobuttonGetFocusColor	GColor	gwin.h	Get background color during focus
402	GCSRadiobuttonSetDisableCaptionColor	void	gwin.h	Set character color when disable
403	GCSRadiobuttonGetDisableCaptionColor	GColor	gwin.h	Get character color when disable
404	GCSRadiobuttonSetDisableBoxColor	void	gwin.h	Set box color when disable
405	GCSRadiobuttonGetDisableBoxColor	GColor	gwin.h	Get box color when disable
406	GCSRadiobuttonSetRadioGroup	void	gwin.h	Set radio group No.
407	GCSRadiobuttonGetRadioGroup	unsigned short	gwin.h	Get radio group No.
408	GCSScrollbarexSetDisplay	void	gwin.h	Set display type
409	GCSScrollbarexGetDisplay	unsigned char	gwin.h	Get display type
410	GCSScrollbarexSetDirection	void	gwin.h	Set scroll bar direction
411	GCSScrollbarexGetDirection	unsigned char	gwin.h	Get scroll bar direction
412	GCSScrollbarexSetPageSize	void	gwin.h	Set page size
413	GCSScrollbarexGetPageSize	unsigned long	gwin.h	Get page size
414	GCSScrollbarexSetRange	void	gwin.h	Set scroll bar range
415	GCSScrollbarexGetRange	void	gwin.h	Get scroll bar range
416	GCSScrollbarexSetScrollArrowColor	void	gwin.h	Set scroll arrow color
417	GCSScrollbarexGetScrollArrowColor	GColor	gwin.h	Get scroll arrow color
418	GCSScrollbarexSetScrollPosition	void	gwin.h	Set scroll bar current position
419	GCSScrollbarexGetScrollPosition	long	gwin.h	Get scroll bar current position
420	GCSScrollbarexSetPrevBtnOnImgID	void	gwin.h	Set design (upper) resource ID when button is ON
421	GCSScrollbarexSetPrevBtnOffImgID	void	gwin.h	Set design (upper) resource ID when button is OFF

No.	Name	Type	File name	Outline
422	GCSScrollbarexGetPrevBtnOnImgID	unsigned short	gwin.h	Get design (upper) resource ID when button is ON
423	GCSScrollbarexGetPrevBtnOffImgID	unsigned short	gwin.h	Get design (upper) resource ID when button is OFF
424	GCSScrollbarexSetNextBtnOnImgID	void	gwin.h	Set design (lower) resource ID when button is ON
425	GCSScrollbarexSetNextBtnOffImgID	void	gwin.h	Set design (lower) resource ID when button is OFF
426	GCSScrollbarexGetNextBtnOnImgID	unsigned short	gwin.h	Get design (lower) resource ID when button is ON
427	GCSScrollbarexGetNextBtnOffImgID	unsigned short	gwin.h	Get design (lower) resource ID when button is OFF
428	GCSScrollbarexSetBackGroundColor	void	gwin.h	Set background color
429	GCSScrollbarexGetBackGroundColor	GColor	gwin.h	Get background color
430	GCSScrollbarexSetOutLineBorderID	void	gwin.h	Set outline 3D border
431	GCSScrollbarexGetOutLineBorderID	unsigned short	gwin.h	Get outline 3D border
432	GCSScrollbarexSetPinchWidth	void	gwin.h	Set pinch width
433	GCSScrollbarexGetPinchWidth	unsigned short	gwin.h	Get pinch width
434	GCSScrollbarexSetPinchColor	void	gwin.h	Set pinch color
435	GCSScrollbarexGetPinchColor	GColor	gwin.h	Get pinch color
436	GCSScrollbarexSetBarEnableStatus	void	gwin.h	Set bar display/non-display
437	GCSScrollbarexGetBarEnableStatus	short	gwin.h	Get bar display/non-display
438	GCSTextboxSetBorderID	void	gwin.h	Set 3D border resource ID
439	GCSTextboxGetBorderID	unsigned short	gwin.h	Get 3D border resource ID
440	GCSTextboxSetFontID	void	gwin.h	Set font resource ID
441	GCSTextboxGetFontID	unsigned short	gwin.h	Get font resource ID
442	GCSTextboxSetCaption	void	gwin.h	Set caption information
443	GCSTextboxGetCaption	GCaption *	gwin.h	Get caption information
444	GCSTextboxSetFocusEffect	void	gwin.h	Set effect during focus
445	GCSTextboxGetFocusEffect	unsigned char	gwin.h	Get effect during focus
446	GCSTextboxSetFocusColor	void	gwin.h	Set background color during focus
447	GCSTextboxGetFocusColor	GColor	gwin.h	Get background color during focus
448	GCSTextboxSetNormalColor	void	gwin.h	Set normal background color
449	GCSTextboxGetNormalColor	GColor	gwin.h	Get normal background color
450	GCSTextboxSetDisableColor	void	gwin.h	Set background color when disabled
451	GCSTextboxGetDisableColor	GColor	gwin.h	Get background color when disabled
452	GCSTextboxSetTextType	void	gwin.h	Set character string type
453	GCSTextboxGetTextType	unsigned char	gwin.h	Get character string type

No.	Name	Type	File name	Outline
454	GCSTextboxSetFormatID	void	gwin.h	Set character string display format resource ID
455	GCSTextboxGetFormatID	unsigned short	gwin.h	Get character string display format resource ID
456	GCSTextboxSetPasswordStatus	void	gwin.h	Set password display status
457	GCSTextboxGetPasswordStatus	short	gwin.h	Get password display status
458	GCSTextboxSetCommaStatus	void	gwin.h	Set comma display status
459	GCSTextboxGetCommaStatus	short	gwin.h	Get comma display status
460	GCSTextboxSetRefuseInputNumberStatus	void	gwin.h	Set no number input status
461	GCSTextboxGetRefuseInputNumberStatus	short	gwin.h	Get no number input status
462	GCSTextboxSetRefuseInputSmallLetterStatus	void	gwin.h	Set one-byte lower case character input disabled status
463	GCSTextboxGetRefuseInputSmallLetterStatus	short	gwin.h	Get one-byte lower case character input disabled status
464	GCSTextboxSetRefuseInputCapitalLetterStatus	void	gwin.h	Set one-byte upper case character input disabled status
465	GCSTextboxGetRefuseInputCapitalLetterStatus	short	gwin.h	Ge one-byte upper case character input disabled status
466	GCSTextboxSetRefuseInputSymbolLetterStatus	void	gwin.h	Set one-byte symbol input disabled status
467	GCSTextboxGetRefuseInputSymbolLetterStatus	short	gwin.h	Get one-byte symbol input disabled status
468	GCSTextboxSetRefuseInputFullLetterStatus	void	gwin.h	Set two-byte character input disabled status
469	GCSTextboxGetRefuseInputFullLetterStatus	short	gwin.h	Get two-byte character input disabled status
470	GCSTextboxSetCheckMaxStatus	void	gwin.h	Set maximum value check status
471	GCSTextboxGetCheckMaxStatus	short	gwin.h	Get maximum value check status
472	GCSTextboxSetCheckMinStatus	void	gwin.h	Set minimum value check status
473	GCSTextboxGetCheckMinStatus	short	gwin.h	Get minimum value check status
474	GCSTextboxSetMaxGValue	void	gwin.h	Set maximum value
475	GCSTextboxGetMaxGValue	GValue	gwin.h	Get maximum value
476	GCSTextboxSetMinGValue	void	gwin.h	Set minimum value
477	GCSTextboxGetMinGValue	GValue	gwin.h	Get minimum value
478	GCSTextboxSetString	short	gwin.h	Set display character string
479	GCSTextboxGetString	short	gwin.h	Get display character string

No.	Name	Type	File name	Outline
480	GCSTextboxSetGValue	short	gwin.h	Set display value
481	GCSTextboxGetGValue	short	gwin.h	Get display value
482	GCSTextboxSetStringBuffer	short	gwin.h	Set display buffer
483	GCSTextboxGetStringBuffer	GTCHAR*	gwin.h	Get display buffer
484	GCSTextboxSetBackGroundPattern	void	gwin.h	Set background fill status
485	GCSTextboxGetBackGroundPattern	short	gwin.h	Get background fill status
486	GCSBWLInit	void	gwin.h	Initialization
487	GCSBWLAdd	short	gwin.h	Addition
488	GCSBWLRemove	short	gwin.h	Deletion
489	GCSBWLClear	void	gwin.h	All clear
490	GCSBWLMoveFirst	short	gwin.h	Move to first
491	GCSBWLMoveLast	short	gwin.h	Move to last
492	GCSBWLGetFirst	GBaseObject*	gwin.h	Get first object
493	GCSBWLGetLast	GBaseObject*	gwin.h	Get last object
494	GCSBWLFindListItem	short	gwin.h	Find object
495	GCSBWLFindFirstItem	short	gwin.h	Move search position to first item
496	GCSBWLFindNextItem	short	gwin.h	Acquire object at search position and move search position to next
497	GCSBWLFindLastItem	short	gwin.h	Move search position to last item
498	GCSBWLFindPrevItem	short	gwin.h	Acquire object at search position and move search position to previous
499	GCSBWLCount	long	gwin.h	Get number of list items
500	GCSBWLSetItemPosition	void	gwin.h	Set current list search position
501	GCSBWLGetItemPosition	void*	gwin.h	Get current list search position
502	GCSBWLNew	HGBWL	gwin.h	Create list object
503	GCSBWLDelete	void	gwin.h	Delete list object
504	GCSWDCGetHGDRAW	HGDRAW	gwin.h	Get drawing handle
505	GCSWDCOffsetOrigin	void	gwin.h	Move position of origin
506	GCSWDCSetClipRect	void	gwin.h	Define clipping area
507	GCSWDCGetClipRect	void	gwin.h	Get clipping area
508	GCSWDCInceptClipRect	short	gwin.h	Define clipping for the part overlapping the current clipping area
509	GCSWDCBeginDraw	void	gwin.h	Start drawing process
510	GCSWDCEndDraw	void	gwin.h	End drawing process
511	GCSWDCNew	HWDC	gwin.h	Create device context
512	GCSWDCDelete	void	gwin.h	Create device context
513	GCSWDCSetOrigin	void	gwin.h	Set position of origin

No.	Name	Type	File name	Outline
514	GCSPictureSetDisplay	void	gwin.h	Set picture display
515	GCSPictureGetDisplay	unsigned char	gwin.h	Get picture display
516	GCSPictureSetBorderID	void	gwin.h	Set 3D border resource ID
517	GCSPictureGetBorderID	unsigned short	gwin.h	Get 3D border resource ID
518	GCSPictureSetFontID	void	gwin.h	Set font resource ID
519	GCSPictureGetFontID	unsigned short	gwin.h	Get font resource ID
520	GCSPictureSetStringID	void	gwin.h	Set caption character string resource ID
521	GCSPictureGetStringID	unsigned short	gwin.h	Get caption character string resource ID
522	GCSPictureSetCaption	void	gwin.h	Set caption information
523	GCSPictureGetCaption	GCaption*	gwin.h	Get caption information
524	GCSPictureSetStatus	void	gwin.h	Set object status
525	GCSPictureGetStatus	unsigned char	gwin.h	Get object status
526	GCSPictureSetMaxStatus	void	gwin.h	Set maximus picture status
527	GCSPictureGetMaxStatus	unsigned char	gwin.h	Get maximus picture status
528	GCSPictureSetStatusDesign	void	gwin.h	Set picture status design
529	GCSPictureGetStatusDesign	GDesign*	gwin.h	Get picture status design
530	GCSChangeActiveFocus	void	gwin.h	Change active focus
531	GCSNCPLCButtonSetAction	void	gncwin.h	Set PLC button operations
532	GCSNCPLCButtonGetAction	unsigned char	gncwin.h	Get PLC button operations
533	GCSNCPLCButtonSetDisplay	void	gncwin.h	Set PLC button display
534	GCSNCPLCButtonGetDisplay	unsigned char	gncwin.h	Get PLC button display
535	GCSNCPLCButtonSetBorderID	void	gncwin.h	Set 3D border resource ID
536	GCSNCPLCButtonGetBorderID	unsigned short	gncwin.h	Get 3D border resource ID
537	GCSNCPLCButtonSetOnDesign	void	gncwin.h	Set ON status design
538	GCSNCPLCButtonGetOnDesign	GDesign*	gncwin.h	Get ON status design
539	GCSNCPLCButtonSetOffDesign	void	gncwin.h	Set OFF status design
540	GCSNCPLCButtonGetOffDesign	GDesign*	gncwin.h	Get OFF status design
541	GCSNCPLCButtonSetOnOnDesign	void	gncwin.h	Set ONON status design
542	GCSNCPLCButtonGetOnOnDesign	GDesign*	gncwin.h	Get ONON status design
543	GCSNCPLCButtonSetOnOffDesign	void	gncwin.h	Set ONOFF status design
544	GCSNCPLCButtonGetOnOffDesign	GDesign*	gncwin.h	Get ONOFF status design
545	GCSNCPLCButtonSetFocusDesign	void	gncwin.h	Set FOCUS status design
546	GCSNCPLCButtonGetFocusDesign	GDesign*	gncwin.h	Get FOCUS status design
547	GCSNCPLCButtonSetDisableDesign	void	gncwin.h	Set Disable status design
548	GCSNCPLCButtonGetDisableDesign	GDesign*	gncwin.h	Get Disable status design
549	GCSNCPLCButtonSetFontID	void	gncwin.h	Set font resource ID
550	GCSNCPLCButtonGetFontID	unsigned short	gncwin.h	Get font resource ID

No.	Name	Type	File name	Outline
551	GCSNCPLCButtonSetOnStringID	void	gncwin.h	Set ON status caption character string resource ID
552	GCSNCPLCButtonGetOnStringID	unsigned short	gncwin.h	Get ON status caption character string resource ID
553	GCSNCPLCButtonSetOffStringID	void	gncwin.h	Set OFF status caption character string resource ID
554	GCSNCPLCButtonGetOffStringID	unsigned short	gncwin.h	Get OFF status caption character string resource ID
555	GCSNCPLCButtonSetOnOnStringID	void	gncwin.h	Set ONON status caption character string resource ID
556	GCSNCPLCButtonGetOnOnStringID	unsigned short	gncwin.h	Get ONON status caption character string resource ID
557	GCSNCPLCButtonSetOnOffStringID	void	gncwin.h	Set ONOFF status caption character string resource ID
558	GCSNCPLCButtonGetOnOffStringID	unsigned short	gncwin.h	Get ONOFF status caption character string resource ID
559	GCSNCPLCButtonSetCaption	void	gncwin.h	Set caption information
560	GCSNCPLCButtonGetCaption	Gcaption *	gncwin.h	Get caption information
561	GCSNCPLCButtonSetFocusEffect	void	gncwin.h	Set effect during focus
562	GCSNCPLCButtonGetFocusEffect	unsigned char	gncwin.h	Get effect during focus
563	GCSNCPLCButtonSetStatus	void	gncwin.h	Set object status
564	GCSNCPLCButtonGetStatus	unsigned char	gncwin.h	Get object status
565	GCSNCPLCButtonSetDeviceType	short	gncwin.h	Set PLC device operations
566	GCSNCPLCButtonGetDeviceType	short	gncwin.h	Get PLC device operations
567	GCSNCPLCButtonSetDevice	short	gncwin.h	Set PLC device
568	GCSNCPLCButtonGetDevice	short	gncwin.h	Get PLC device
569	GCSNCPLCButtonSetDeviceVal	short	gncwin.h	Set PLC device value
570	GCSNCPLCButtonGetDeviceVal	short	gncwin.h	Get PLC device value
571	GCSNCPLCTextboxSetBorderID	void	gncwin.h	Set 3D border resource ID
572	GCSNCPLCTextboxGetBorderID	unsigned short	gncwin.h	Get 3D border resource ID
573	GCSNCPLCTextboxSetFontID	void	gncwin.h	Set font resource ID
574	GCSNCPLCTextboxGetFontID	unsigned short	gncwin.h	Get font resource ID
575	GCSNCPLCTextboxSetCaption	void	gncwin.h	Set caption information
576	GCSNCPLCTextboxGetCaption	Gcaption *	gncwin.h	Get caption information
577	GCSNCPLCTextboxSetFocusEffect	void	gncwin.h	Set effect during focus
578	GCSNCPLCTextboxGetFocusEffect	unsigned char	gncwin.h	Get effect during focus
579	GCSNCPLCTextboxSetFocusColor	void	gncwin.h	Set background color during focus
580	GCSNCPLCTextboxGetFocusColor	GColor	gncwin.h	Get background color during focus
581	GCSNCPLCTextboxSetNormalColor	void	gncwin.h	Set normal background color

No.	Name	Type	File name	Outline
582	GCSNCPLCTextboxGetNormalColor	GColor	gncwin.h	Get normal background color
583	GCSNCPLCTextboxSetDisableColor	void	gncwin.h	Set background color when disabled
584	GCSNCPLCTextboxGetDisableColor	GColor	gncwin.h	Get background color when disabled
585	GCSNCPLCTextboxSetTextType	void	gncwin.h	Set character string type
586	GCSNCPLCTextboxGetTextType	unsigned char	gncwin.h	Get character string type
587	GCSNCPLCTextboxSetPasswordStatus	void	gncwin.h	Set password display status
588	GCSNCPLCTextboxGetPasswordStatus	short	gncwin.h	Get password display status
589	GCSNCPLCTextboxSetCommaStatus	void	gncwin.h	Set comma display status
590	GCSNCPLCTextboxGetCommaStatus	short	gncwin.h	Get comma display status
591	GCSNCPLCTextboxSetCheckMaxStatus	void	gncwin.h	Set maximum value check status
592	GCSNCPLCTextboxSetCheckMaxStatus	short	gncwin.h	Get maximum value check status
593	GCSNCPLCTextboxSetCheckMinStatus	void	gncwin.h	Set minimum value check status
594	GCSNCPLCTextboxGetCheckMinStatus	short	gncwin.h	Get minimum value check status
595	GCSNCPLCTextboxSetMaxGValue	void	gncwin.h	Set maximum value
596	GCSNCPLCTextboxGetMaxGValue	GValue	gncwin.h	Get maximum value
597	GCSNCPLCTextboxSetMinGValue	void	gncwin.h	Set minimum value
598	GCSNCPLCTextboxGetMinGValue	GValue	gncwin.h	Get minimum value
599	GCSNCPLCTextboxSetString	short	gncwin.h	Set display character string
600	GCSNCPLCTextboxGetString	short	gncwin.h	Get display character string
601	GCSNCPLCTextboxSetGValue	short	gncwin.h	Set display value
602	GCSNCPLCTextboxGetGValue	short	gncwin.h	Get display value
603	GCSNCPLCTextboxSetStringBuffer	short	gncwin.h	Set display buffer
604	GCSNCPLCTextboxGetStringBuffer	GTCHAR*	gncwin.h	Get display buffer
605	GCSNCPLCTextboxSetBackGroundPattern	void	gncwin.h	Set background fill status
606	GCSNCPLCTextboxGetBackGroundPattern	short	gncwin.h	Get background fill status
607	GCSNCPLCTextboxSetSelectStringStartPos	void	gncwin.h	Set selected character string start position
608	GCSNCPLCTextboxGetSelectStringStartPos	unsigned short	gncwin.h	Get selected character string start position
609	GCSNCPLCTextboxSetSelectStringEndPos	void	gncwin.h	Set selected character string end position
610	GCSNCPLCTextboxGetSelectStringEndPos	unsigned short	gncwin.h	Get selected character string end position

No.	Name	Type	File name	Outline
611	GCSNCPLCTextboxSetIntegerWidth	void	gncwin.h	Set integer part width
612	GCSNCPLCTextboxGetIntegerWidth	unsigned char	gncwin.h	Get integer part width
613	GCSNCPLCTextboxSetDecimalWidth	void	gncwin.h	Set decimal part width
614	GCSNCPLCTextboxGetDecimalWidth	unsigned char	gncwin.h	Get decimal part width
615	GCSNCPLCTextboxSetScale	void	gncwin.h	Set scale
616	GCSNCPLCTextboxGetScale	void	gncwin.h	Get scale
617	GCSNCPLCTextboxSetOffset	void	gncwin.h	Set offset
618	GCSNCPLCTextboxGetOffset	long	gncwin.h	Get offset
619	GCSNCPLCTextboxSetDevice	short	gncwin.h	Set PLC device
620	GCSNCPLCTextboxGetDevice	short	gncwin.h	Get PLC device
621	GCSNCPLCTextboxSetTextSize	void	gncwin.h	Set size
622	GCSNCPLCTextboxGetTextSize	unsigned long	gncwin.h	Get size
623	GCSNCPLCTextboxSetZeroSuppressStatus	void	gncwin.h	Set zero suppress display status
624	GCSNCPLCTextboxGetZeroSuppressStatus	short	gncwin.h	Get zero suppress display status
625	GCSNCTableSetCellForeColor	void	gncwin.h	Set foreground color of each cell
626	GCSNCTableSetCellBackColor	void	gncwin.h	Set background color of each cell
627	GCSNCTableSetCellString	void	gncwin.h	Set character string of each cell
628	GCSNCTableGetCellString	void	gncwin.h	Get character string of each cell
629	GCSNCTableSetSubCursorNum	void	gncwin.h	Set sub-cursor position
630	GCSNCTableGetSubCursorNum	void	gncwin.h	Get sub-cursor position
631	GCSNCTableSetCellWidth	void	gncwin.h	Set cell width
632	GCSNCTableSetCellHight	void	gncwin.h	Set cell height
633	GCSNCTableGetCellNumFromPoint	void	gncwin.h	Get cell No. including pertinent coordinates
634	GCSNCTableSetLineTitleString	void	gncwin.h	Set line title
635	GCSNCTableGetLineTitleString	void	gncwin.h	Get line title
636	GCSNCTableSetRowTitleString	void	gncwin.h	Set row title
637	GCSNCTableGetRowTitleString	void	gncwin.h	Get row title
638	GCSNCTableSetLineTitleStringPosition	void	gncwin.h	Set line title character string position
639	GCSNCTableSetRowTitleStringPosition	void	gncwin.h	Set row title character string position

No.	Name	Type	File name	Outline
640	GCSNCTableSetCellStringPosition	void	gncwin.h	Set area part character string position
641	GCSNCTableSetBeforeSubCursorNum	void	gncwin.h	Set last sub-cursor No.
642	GCSNCTableGetBeforeSubCursorNum	void	gncwin.h	Get last sub-cursor No.
643	GCSNCTableSetLineTitleForeColor	void	gncwin.h	Set foreground color of line title
644	GCSNCTableSetLineTitleBackColor	void	gncwin.h	Set background color of line title
645	GCSNCTableSetRowTitleForeColor	void	gncwin.h	Set foreground color of row title
646	GCSNCTableSetRowTitleBackColor	void	gncwin.h	Set background color of row title
647	GCSNCTableSetSubCursorForeColor	void	gncwin.h	Set foreground color of sub-cursor
648	GCSNCTableSetSubCursorBackColor	void	gncwin.h	Set background color of sub-cursor

5.2 Description of GCS Functions

GCSGetType

gwin.h

Outline	Get object type
Syntax	unsigned short GCSGetType(GBaseObject *pSelf*)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Object type
Details	Gets the object type.
File name	gwin.h

GCSGetID

gwin.h

Outline	Get object ID
Syntax	Unsigned short GCSGetID(GBaseObject *pSelf*)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Object ID
Details	Gets the object ID.
File name	gwin.h

GCSGetXPosition

gwin.h

Outline	Get horizontal position
Syntax	short GCSGetXPosition(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Horizontal position
Details	Gets the horizontal position of the object.
File name	gwin.h

GCSGetYPosition

gwin.h

Outline	Get vertical position
Syntax	short GCSGetYPosition(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Vertical position
Details	Gets the vertical position of the object.
File name	gwin.h

GCSSetPosition

gwin.h

Outline	Set position
Syntax	void GCSSetPosition(GBaseObject *pSelf, GPoint *pPosition)
Argument	(i) GBaseObject *pSelf : object for operation (i) GPoint *pPosition : point structure being set
Return value	None
Details	Sets the position of the object using a point structure.
File name	gwin.h

GCSGetPosition

gwin.h

Outline	Get position
Syntax	void GCSGetPosition(GBaseObject *pSelf, GPoint *pPosition)
Argument	(i) GBaseObject *pSelf : object for operation (o) GPoint *pPosition : point structure being stored
Return value	None
Details	Gets the position of the object using a point structure.
File name	gwin.h

GCSGetWidth

gwin.h

Outline	Get width
Syntax	short GCSGetWidth(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Width
Details	Gets the object width.
File name	gwin.h

GCSGetHeight

gwin.h

Outline	Get height
Syntax	short GCSGetHeight(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Height
Details	Gets the object height.
File name	gwin.h

GCSGetSize

gwin.h

Outline	Get size
Syntax	void GCSGetSize(GBaseObject *pSelf, GPoint *pSize)
Argument	(i) GBaseObject *pSelf : object for operation (o) GPoint *pSize : point structure for which the size is being stored
Return value	None
Details	Gets the object size.
File name	gwin.h

GCSSetBounds

gwin.h

Outline	Set relative position within parent object
Syntax	void GCSSetBounds(GBaseObject *pSelf, GRect *pBounds)
Argument	(i) GBaseObject *pSelf : object for operation (i) GRect *pBounds : rectangle variable
Return value	None
Details	Sets the relative position of the object within parent object.
File name	gwin.h

GCSGetBounds

gwin.h

Outline	Get relative position within parent object
Syntax	void GCSGetBounds(GBaseObject *pSelf, GRect *pBounds)
Argument	(i) GBaseObject *pSelf : object for operation (o) GRect *pBounds : rectangle variable
Return value	None
Details	Gets the relative position of the object within parent object.
File name	gwin.h

GCSCalcClientRect

gwin.h

Outline	Get coordinate position of client area
Syntax	void GCSCalcClientRect(GBaseObject *pSelf, GRect *pClient)
Argument	(i) GBaseObject *pSelf : object for operation (o) GRect *pClinet : client area coordinates
Return value	None
Details	Gets the client area for the object with the upper left of the object as the origin.
File name	gwin.h

GCSGetClientRect

gwin.h

Outline	Get object client area
Syntax	void GCSGetClientRect(GBaseObject *pSelf, GRect *pClient)
Argument	(i) GBaseObject *pSelf : object for operation (o) GRect *pClinet : client area
Return value	None
Details	Gets the client area for the object. Gets the rectangle with the upper left of the object as the origin.
File name	gwin.h

GCSClientToScreen

gwin.h

Outline	Convert client coordinates to screen coordinates
Syntax	void GCSClientToScreen(GBaseObject *pSelf, GRect *pRect)
Argument	(i) GBaseObject *pSelf : object for operation (i/o) GRect *pRect : client coordinate rectangle being converted
Return value	None
Details	Converts the object client coordinates to screen coordinates.
File name	gwin.h

GCSGetStyle

gwin.h

Outline	Batch style acquisition
Syntax	unsigned long GCSGetStyle(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Style value
Details	Gets the styles of the object all at once.
File name	gwin.h

GCSSetVisibleStatus

gwin.h

Outline	Set display/non-display status
Syntax	void GCSSetVisibleStatus(GBaseObject *pSelf, short fStatus)
Argument	(i) GBaseObject *pSelf : object for operation (i) short fStatus : display/non-display status FALSE : non-display TRUE : display
Return value	None
Details	Sets the display/non-display status for the object. When switching the display/non-display, the object area is registered as a redraw area.
File name	gwin.h

GCSGetVisibleStatus

gwin.h

Outline	Get display/non-display status
Syntax	short GCSGetVisibleStatus(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Display/non-display status FALSE : non-display TRUE : display
Details	Gets the display/non-display status for the object.
File name	gwin.h

GCSSetEnableStatus

gwin.h

Outline	Set controllable or uncontrollable status
Syntax	void GCSSetEnableStatus(GBaseObject *pSelf, short fStatus)
Argument	(i) GBaseObject *pSelf : object for operation (i) short fStatus : controllable or uncontrollable status FALSE : uncontrollable TRUE : controllable
Return value	None
Details	Sets the controllable or uncontrollable status for the object. Controllability with mouse input or key input is determined according to this flag.
File name	gwin.h

GCSGetEnableStatus

gwin.h

Outline	Get controllable or uncontrollable status
Syntax	short GCSGetEnableStatus(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	FALSE : uncontrollable TRUE : controllable
Details	Gets the controllable or uncontrollable status for the object.
File name	gwin.h

GCSSetConsumableStatus

gwin.h

Outline	Set input event consumable or input event not consumable status
Syntax	void GCSSetConsumableStatus(GBaseObject *pSelf, short fStatus)
Argument	(i) GBaseObject *pSelf : object for operation (i) short fStatus : consumable/not consumable status FALSE : not consumable TRUE : consumable
Return value	None
Details	Sets the event consumption or non-consumption when the object has uncontrollable status. Whether the event is consumed or not consumed is controlled by this flag for this object.
File name	gwin.h

GCSGetConsumableStatus

gwin.h

Outline	Get input event consumable or input event not consumable status
Syntax	short GCSGetConsumableStatus(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Consumable/not consumable status FALSE : not consumable TRUE : consumable
Details	Gets the input event consumable or input event not consumable status.
File name	gwin.h

GCSSetCaptionStatus

gwin.h

Outline	Set caption display status
Syntax	void GCSSetCaptionStatus(GBaseObject *pSelf, short fStatus)
Argument	(i) GBaseObject *pSelf : object for operation (i) short fStatus : display/non-display status FALSE : display TRUE : non-display
Return value	None
Details	Sets the object caption display status. Displaying or not displaying captions for the objects that have captions is switched according to this setting.
File name	gwin.h

GCSGetCaptionStatus

gwin.h

Outline	Get caption display status
Syntax	short GCSGetCaptionStatus(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Caption display status FALSE : display TRUE : non-display
Details	Gets the object caption display status.
File name	gwin.h

GCSSetFocusStatus

gwin.h

Outline	Set focus status
Syntax	void GCSSetFocusStatus(GBaseObject *pSelf, short fStatus)
Argument	(i) GBaseObject *pSelf : object for operation (i) short fStatus : focused/not focused status FALSE : not focused TRUE : focused
Return value	None
Details	Sets the object focus status for the object.
File name	gwin.h

GCSGetFocusStatus

gwin.h

Outline	Get focus status
Syntax	short GCSGetFocusStatus(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Focus status FALSE : not focused TRUE : focused
Details	Gets the object focus status for the object.
File name	gwin.h

GCSSetPanelStatus

gwin.h

Outline	Set management panel object status
Syntax	void GCSSetPanelStatus(GBaseObject *pSelf, short fStatus)
Argument	(i) GBaseObject *pSelf : object for operation (i) short fStatus : management panel status FALSE : other than management panel object TRUE : management panel object
Return value	None
Details	Sets the object as a management panel object or not.
File name	gwin.h

GCSGetPanelStatus

gwin.h

Outline	Get management panel object status
Syntax	short GCSGetPanelStatus(GBaseObject *pSelf*)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Management panel status FALSE : other than management panel object TRUE : management panel object
Details	Gets the object focus status for the object.
File name	gwin.h

GCSSetStyleStatus

gwin.h

Outline	Set style
Syntax	void GCSSetStyleStatus(GBaseObject *pSelf,* unsigned long *ulStyle,* short *fStatus*)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned long ulStyle : style to be set (i) short fStatus : specified style status
Return value	None
Details	Sets the object style. With this method, it is necessary to redraw when redrawing is necessary to make this setting.
File name	gwin.h

GCSGetStyleStatus

gwin.h

Outline	Get style
Syntax	short GCSGetStyleStatus(GBaseObject *pSelf, unsigned long ulStyle)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned long ulStyle : style being acquired
Return value	Specified style status (depends on style being acquired)
Details	Gets the specified style status for the object.
File name	gwin.h

GCSSetParent

gwin.h

Outline	Set parent object
Syntax	void GCSSetParent(GBaseObject *pSelf, GBaseObject *pParent)
Argument	(i) GBaseObject *pSelf : object for operation (i) GBaseObject *pParent : parent object being set
Return value	None
Details	Sets the parent object for managing the object.
File name	gwin.h

GCSGetParent

gwin.h

Outline	Get parent object
Syntax	GBaseObject* GCSGetParent(GBaseObject *pSelf*)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	NULL : no parent object Other than NULL : parent object
Details	Gets the parent object for managing the object.
File name	gwin.h

GCSImportProperty

gwin.h

Outline	Import property settings
Syntax	short GCSImportProperty(GBaseObject *pSelf,* const GBaseWindowProperty *pImport*)
Argument	(i) GBaseObject *pSelf : object for operation (i) const GBaseWindowProperty *pImport : property structure
Return value	FALSE : acquisition failed TRUE : acquisition succeeded
Details	Imports the property setting as a setting object.
File name	gwin.h

GCSExportProperty

gwin.h

Outline	Export property settings
Syntax	void GCSExportProperty(GBaseObject *pSelf, GBaseWindowProperty * pExport)
Argument	(i) GBaseObject *pSelf : object for operation (o) GBaseWindowProperty *pExport : property structure
Return value	None
Details	Exports the object setting to a property structure.
File name	gwin.h

GCSCreate

gwin.h

Outline	Create process
Syntax	short GCSCreate(GBaseObject *pSelf, GBaseObject *pParent)
Argument	(i) GBaseObject *pSelf : object for operation (i) GBaseObject *pParent : management object
Return value	FALSE : creation failed TRUE : creation succeeded
Details	Creates an object.
File name	gwin.h

GCSCreateVram

gwin.h

Outline	Create process (create VRAM)
Syntax	short GCSCreateVram(GBaseObject *pScreen, GBaseObject *pParent, void *pData)
Argument	(i) GBaseObject *pSelf : object for operation (i) GBaseObject *pParent : management object (i) void *pData : additional information when drawing handle is created
Return value	FALSE : creation failed TRUE : creation succeeded
Details	Creates an object. In the process for creating a GCScreen object, a drawing handle and drawing context are created.
File name	gwin.h

GCSDelete

gwin.h

Outline	Delete process
Syntax	void GCSDelete(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	None
Details	Deletes an object.
File name	gwin.h

GCSPaint

gwin.h

Outline	Drawing process
Syntax	void GCSPaint(GBaseObject *pSelf, HWDC hWDC)
Argument	(i) GBaseObject *pSelf : object for operation (i) HWDC hWDC : drawing context
Return value	None
Details	Carries out drawing process for object GM_PAINT message. Since there is no special distinction between the client area and non-client area with the GBaseObject class, only client area drawing is called.
File name	gwin.h

GCSOnDraw

gwin.h

Outline	Client area drawing process
Syntax	void GCSOnDraw(GBaseObject *pSelf, HWDC hWDC)
Argument	(i) GBaseObject *pSelf : object for operation (i) HWDC hWDC : drawing context
Return value	None
Details	Carries out drawing process for object client area. Since there is no special distinction between the client area and non-client area with the GBaseObject class, calling is only for the client area.
File name	gwin.h

GCSOnNcDraw

gwin.h

Outline	Non-client area drawing process
Syntax	void GCSOnNcDraw(GBaseObject *pSelf, HWDC hWDC)
Argument	(i) GBaseObject *pSelf : object for operation (i) HWDC hWDC : drawing context
Return value	None
Details	Carries out drawing process for object non-client area. Since there is no special distinction between the client area and non-client area with the GBaseObject class, non-client area is not called.
File name	gwin.h

GCSAddRefreshRect

gwin.h

Outline	Client area redraw area registration
Syntax	void GCSAddRefreshRect(GBaseObject *pSelf, GRect *pRefreshRect, short fErase)
Argument	(i) GBaseObject *pSelf : object for operation (i) GRect *pRefreshRect : area for redrawing (client coordinates) (i) short fErase : background drawing present/none GW_REFRESH_BGERASE : background drawing GW_REFRESH_BGNOERASE : no background drawing
Return value	None
Details	Registers the redrawing area in object client coordinates. Background redrawing present/none is selected using fErase.
File name	gwin.h

GCSLButtonPress

gwin.h

Outline	Process for pressing left mouse button
Syntax	long GCSLButtonPress(GBaseObject **pSelf,* unsigned long *ulStatus,* GPoint *gptPoint*)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned long ulStatus : pressed button status (i) GPoint gptPoint : position of point device coordinates
Return value	FALSE : do not consume mouse event TRUE : consume mouse event
Details	Carries out the processing for pressing the left mouse button for the object. Calls the left mouse button press processing for the client area. When the mouse event is not consumed, FALSE is returned, and when the mouse event is consumed, TRUE is returned.
File name	gwin.h

GCSOnLButtonPress

gwin.h

Outline	Client area process for pressing left mouse button
Syntax	long GCSOnLButtonPress(GBaseObject **pSelf,* unsigned long *ulStatus,* GPoint *gptPoint*)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned long ulStatus : pressed button status (i) GPoint gptPoint : position of point device coordinates
Return value	FALSE : do not consume mouse event TRUE : consume mouse event
Details	Carries out the processing for pressing the left mouse button for the client area on the object. When the mouse event is not consumed, FALSE is returned, and when the mouse event is consumed, TRUE is returned.
File name	gwin.h

GCSOnNcLButtonPress

gwin.h

Outline	Non-client area process for pressing left mouse button
Syntax	long GCSOnNcLButtonPress(GBaseObject *pSelf, unsigned long ulStatus, GPoint gptPoint)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned long ulStatus : pressed button status (i) GPoint gptPoint : position of point device coordinates
Return value	FALSE : do not consume mouse event TRUE : consume mouse event
Details	Carries out the processing for pressing the left mouse button for the non-client area on the object. When the mouse event is not consumed, FALSE is returned, and when the mouse event is not consumed, TRUE is returned.
File name	gwin.h

GCSLButtonRelease

gwin.h

Outline	Process for releasing left mouse button
Syntax	void GCSLButtonRelease(GBaseObject **pSelf,* unsigned long *ulStatus,* GPoint *gptPoint*)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned long ulStatus : pressed button status (i) GPoint gptPoint : position of point device coordinates
Return value	None
Details	Carries out the processing for releasing the left mouse button for the object. Calls the left mouse button release processing for the client area.
File name	gwin.h

GCSOnLButtonRelease

gwin.h

Outline	Client area process for releasing left mouse button
Syntax	void GCSOnLButtonRelease(GBaseObject **pSelf,* unsigned long *ulStatus,* GPoint *gptPoint*)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned long ulStatus : pressed button status (i) GPoint gptPoint : position of point device coordinates
Return value	None
Details	Carries out the processing for releasing the left mouse button for the client area on the object.
File name	gwin.h

GCSOnNcLButtonRelease

gwin.h

Outline	Non-client area process for releasing left mouse button
Syntax	void GCSOnNcLButtonRelease(GBaseObject **pSelf,* unsigned long *ulStatus,* GPoint *gptPoint*)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned long ulStatus : pressed button status (i) GPoint gptPoint : position of point device coordinates
Return value	None
Details	Carries out the processing for releasing the left mouse button for the non-client area on the object.
File name	gwin.h

GCSMouseMove

gwin.h

Outline	Pointing device movement process
Syntax	void GCSMouseMove(GBaseObject **pSelf,* long *ulStatus,* GPoint *gptPoint*)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned long ulStatus : pressed button status (i) GPoint gptPoint : position of point device coordinates
Return value	None
Details	Carries out the processing for pointing device movement for the object. Calls the pointing device movement processing for the client area.
File name	gwin.h

GCSOnMouseMove

gwin.h

Outline	Pointing device client area movement process
Syntax	void GCSOnMouseMove(GBaseObject *pSelf, long ulStatus, GPoint gptPoint)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned long ulStatus : pressed button status (i) GPoint gptPoint : position of point device coordinates
Return value	None
Details	Carries out the processing for pointing device movement for the object for the client area.
File name	gwin.h

GCSOnNcMouseMove

gwin.h

Outline	Pointing device non-client area movement process
Syntax	void GCSOnNcMouseMove(GBaseObject *pSelf, long ulStatus, GPoint gptPoint)
Argument	(i) GBaseObject *pSelf : object for operation (i) long ulStatus : pressed button status (i) GPoint gptPoint : position of point device coordinates
Return value	None
Details	Carries out the processing for pointing device movement for the object for the non-client area.
File name	gwin.h

GCSKeyPress

gwin.h

Outline	Process for pressing key
Syntax	long GCSKeyPress(GBaseObject **pSelf,* unsigned short *usKeyCode,* unsigned long *ulStatus*)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned short usKeyCode : virtual key code (i) unsigned long ulStatus : key status
Return value	FALSE : do not consume key event TRUE : consume key event
Details	Carries out the processing for pressing a key for the object. When the key event is not consumed, FALSE is returned, and when the key event is consumed, TRUE is returned.
File name	gwin.h

GCSKeyRelease

gwin.h

Outline	Process for releasing key
Syntax	void GCSKeyRelease(GBaseObject **pSelf,* unsigned short *usKeyCode,* unsigned long *ulStatus*)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned short usKeyCode : virtual key code (i) unsigned long ulStatus : key status
Return value	None
Details	Carries out the processing for releasing a key for the object.
File name	gwin.h

GCSSetFocus

gwin.h

Outline	Focus setting process
Syntax	void GCSSetFocus(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	None
Details	Carries out the processing for setting the focus for the object.
File name	gwin.h

GCSKillFocus

gwin.h

Outline	Focus removing process
Syntax	void GCSKillFocus(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	None
Details	Carries out the processing for removing the focus for the object.
File name	gwin.h

GCSTimer

gwin.h

Outline	Timer process
Syntax	void GCSTimer(GBaseObject **pSelf,* unsigned short *usTimerID*)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned short usTimerID : timer ID
Return value	None
Details	Carries out the timer processing for the object.
File name	gwin.h

GCSUser

gwin.h

Outline	User process
Syntax	long GCSUser(GBaseObject **pSelf,* long *lLParam,* long *lUParam*)
Argument	(i) GBaseObject *pSelf : object for operation (i) long lLParam : appended information (lower-side) (i) long lUParam : appended information (upper-side)
Return value	-1 : not supported 0 or more : arbitrary
Details	Carries out the user processing for the object. Null processing.
File name	gwin.h

GCSSetFocusInformation

gwin.h

Outline	Set focus movement information
Syntax	void GCSSetFocusInformation(GBaseObject *pSelf, GFocusInformation *pFocusInfo)
Argument	(i) GBaseObject *pSelf : object for operation (i) GFocusInformation short *pFocusInfo : focus movement information
Return value	None
Details	Set focus movement information for the object. In the case of a virtual key code where the object is not received in a key operation for an object, focus movement follows this setting.
File name	gwin.h

GCSGetFocusInformation

gwin.h

Outline	Get focus movement information
Syntax	GFocusInformation *GCSGetFocusInformation(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Pointer to focus movement information
Details	Gets focus movement information for the object. In the case of a virtual key code where the object is not received in a key operation for an object, focus movement follows this setting.
File name	gwin.h

GCSChar

gwin.h

Outline	Character input process
Syntax	void GCSChar(GBaseObject **pSelf,* unsigned short *usCharCode,* unsigned long *ulStatus*)
Argument	(i) GBaseObject *pSelf : object for operation unsigned short usCharCode : character code unsigned long ulStatus : status
Return value	None
Details	Carries out the character input processing for the object. When one byte size of characters is set as the character code in usCharCode, processing is carried out for one-byte characters, and when two bytes size is set, processing is carried out two-byte characters.
File name	gwin.h

GCSClose

gwin.h

Outline	Issue object close process request
Syntax	void GCSClose(GBaseObject **pSelf*)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	None
Details	For a related parent object, a request for termination processing for the object is issued.
File name	gwin.h

GCSDeleteChild

gwin.h

Outline	Child object deletion process
Syntax	void GCSDeleteChild(GBaseObject *pSelf, GBaseObject *pChild)
Argument	(i) GBaseObject *pSelf : object for operation (i) GBaseObject *pChild : child object to be deleted
Return value	None
Details	Executes deletion processing for related child object.
File name	gwin.h

GCSSetMemorySpace

gwin.h

Outline	Set memory information
Syntax	void GCSSetMemorySpace(GBaseObject *pSelf, unsigned char ucType, unsigned char ucPlane)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned char ucType : memory management information type (i) unsigned char ucPlane : memory management information plane
Return value	None
Details	Sets the memory management information used by the object.
File name	gwin.h

GCSGetMemorySpace

gwin.h

Outline	Get memory information
Syntax	void GCSGetMemorySpace(GBaseObject *pSelf, unsigned char *pType, unsigned char *pPlane)
Argument	(i) GBaseObject *pSelf : object for operation (o) unsigned char *pType : memory management information type (o) unsigned char *pPlane : memory management information plane
Return value	None
Details	Gets the memory management information used by the object.
File name	gwin.h

GCSSetDrawForceStatus

gwin.h

Outline	Set forced drawing status
Syntax	void GCSSetDrawForceStatus(GBaseObject *pSelf, short fStatus)
Argument	(i) GBaseObject *pSelf : object for operation (i) short fStatus : forced drawing status FALSE : no forced drawing TRUE : forced drawing
Return value	None
Details	Sets the forced drawing status for the object.
File name	gwin.h

GCSGetDrawForceStatus

gwin.h

Outline	Get forced drawing status
Syntax	short GCSGetDrawForceStatus(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Forced drawing status FALSE : no forced drawing TRUE : forced drawing
Details	Get object forced drawing status.
File name	gwin.h

GCSSetRedrawStatus

gwin.h

Outline	Set redraw status
Syntax	void GCSSetRedrawStatus(GBaseObject *pSelf, short fStatus)
Argument	(i) GBaseObject *pSelf : object for operation (i) short fStatus : redraw status FALSE : no redrawing TRUE : redraw
Return value	None
Details	Sets the redrawing status for the object.
File name	gwin.h

GCSGetRedrawStatus

gwin.h

Outline	Get redraw status
Syntax	short GCSGetRedrawStatus(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Redraw status FALSE : no redrawing TRUE : redraw
Details	Gets the redrawing status for the object.
File name	gwin.h

GCSClearDrawStatus

gwin.h

Outline	Clear drawing status
Syntax	void GCSClearDrawStatus(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	None
Details	Clears the object drawing status.
File name	gwin.h

GCSAddDirtyRect

gwin.h

Outline	Client area redraw area registration (internal processing function)
Syntax	void GCSAddDirtyRect(GBaseObject * *pSelf,* GRect **pRefreshRect,* short *fErase*)
Argument	(i) GBaseObject *pSelf : object for operation (i) GRect *pRefreshRect : area for redrawing (client coordinates) (i) short fErase : background drawing present/none GW_REFRESH_BGERASE : background drawing GW_REFRESH_BGNOERASE : no background drawing
Return value	None
Details	Registers the redrawing area in object client coordinates. Background redrawing present/none is selected using fErase.
File name	gwin.h

GCSSetPrevTimerObject

gwin.h

Outline	Set previous object on timer bidirectional list
Syntax	void GCSSetPrevTimerObject(GBaseObject *pSelf, GBaseObject *pbwPrev)
Argument	(i) GBaseObject *pSelf : object for operation (i) GBaseObject *pbwPrev : object being set
Return value	None
Details	Sets the previous list object on the timer bidirectional list held in the object.
File name	gwin.h

GCSGetPrevTimerObject

gwin.h

Outline	Get previous object on timer bidirectional list
Syntax	GBaseObject* GCSGetPrevTimerObject(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Previous object on timer bidirectional list NULL : do not set Other than NULL : pointer to previous object
Details	Gets the previous list object on the timer bidirectional list held in the object.
File name	gwin.h

GCSSetNextTimerObject

gwin.h

Outline	Set next object on timer bidirectional list
Syntax	void GCSSetNextTimerObject(GBaseObject *pSelf, GBaseObject *pbwNext)
Argument	(i) GBaseObject *pSelf : object for operation (i) GBaseObject *pbwNext : object being set
Return value	None
Details	Sets the next list object on the timer bidirectional list held in the object.
File name	gwin.h

GCSGetNextTimerObject

gwin.h

Outline	Get next object on timer bidirectional list
Syntax	GBaseObject* GCSGetNextTimerObject(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Next object on timer bidirectional list NULL : do not set Other than NULL : pointer to next object
Details	Gets the next list object on the timer bidirectional list held in the object.
File name	gwin.h

GCSSetBlinkStatus

gwin.h

Outline	Set blink status
Syntax	void GCSSetBlinkStatus(GBaseObject **pSelf,* unsigned char *ucBlinkStatus*)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned char ucBlinkStatus : blink status GCBLINK_BLINK_OFF : OFF status GCBLINK_BLINK_ON : ON status
Return value	None
Details	Sets the object blinking status for the object.
File name	gwin.h

GCSGetBlinkStatus

gwin.h

Outline	Get blink status
Syntax	unsigned char GCSGetBlinkStatus(GBaseObject **pSelf*)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	blink status GCBLINK_BLINK_OFF : OFF status GCBLINK_BLINK_ON : ON status
Details	Gets the object blinking status for the object.
File name	gwin.h

GCSGetParentPanel

gwin.h

Outline	Get management panel object
Syntax	void GCSGetParentPanel(GBaseObject *pSelf, GBaseObject **pPanel, GBaseObject ** pWindow)
Argument	(i) GBaseObject *pSelf : object for operation (o) GBaseObject *pPanel : management panel object (o) GBaseObject *pWindow : management window object
Return value	None
Details	Gets the panel object for managing the object. Gets the window object matched to the object in the window.
File name	gwin.h

GCSSetPrevBlinkControl

gwin.h

Outline	Set previous object on blink bidirectional list
Syntax	void GCSSetPrevBlinkControl(GBaseObject *pSelf, GBaseObject *pvPrev)
Argument	(i) GBaseObject *pSelf : object for operation (i) GBaseObject *pvPrev : object being set
Return value	None
Details	Sets the previous list object on the blink bidirectional list held in the object.
File name	gwin.h

GCSGetPrevBlinkControl

gwin.h

Outline	Get previous object on blink bidirectional list
Syntax	GBaseObject* GCSGetPrevBlinkControl(GBaseObject *pSelf*)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Previous object on blink bidirectional list NULL : do not set Other than NULL : pointer to previous object
Details	Gets the previous list object on the blink bidirectional list held in the object.
File name	gwin.h

GCSSetNextBlinkControl

gwin.h

Outline	Set next object on blink bidirectional list
Syntax	void GCSSetNextBlinkControl(GBaseObject *pSelf,* GBaseObject *pvNext*)
Argument	(i) GBaseObject *pSelf : object for operation (i) GBaseObject *pvNext : object being set
Return value	None
Details	Sets the next list object on the blink bidirectional list held in the object.
File name	gwin.h

GCSGetNextBlinkControl

gwin.h

Outline	Get next object on blink bidirectional list
Syntax	GBaseObject* GCSGetNextBlinkControl(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Next object on blink bidirectional list NULL : do not set Other than NULL : pointer to next object
Details	Gets the next list object on the blink bidirectional list held in the object.
File name	gwin.h

GCSSetPrevScrollControl

gwin.h

Outline	Set previous object on scroll bidirectional list
Syntax	void GCSSetPrevScrollControl(GBaseObject *pSelf, GBaseObject *pvPrev)
Argument	(i) GBaseObject *pSelf : object for operation (i) GBaseObject *pbPrev : object being set
Return value	None
Details	Sets the previous list object on the scroll bidirectional list held in the object.
File name	gwin.h

GCSGetPrevScrollControl

gwin.h

Outline	Get previous object on scroll bidirectional list
Syntax	GBaseObject* GCSGetPrevScrollControl(GBaseObject *pSelf*)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	previous object on scroll bidirectional list NULL : do not set Other than NULL : pointer to previous object
Details	Gets the previous list object on the scroll bidirectional list held in the object.
File name	gwin.h

GCSSetNextScrollControl

gwin.h

Outline	Set next object on scroll bidirectional list
Syntax	void GCSSetNextScrollControl(GBaseObject * *pSelf,* GBaseObject * *pvNext*)
Argument	(i) GBaseObject *pSelf : object for operation (i) GBaseObject *pvNext : object being set
Return value	None
Details	Sets the next list object on the scroll bidirectional list held in the object.
File name	gwin.h

GCSGetNextScrollControl

gwin.h

Outline	Get next object on scroll bidirectional list
Syntax	GBaseObject* GCSGetNextScrollControl(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Next object on scroll bidirectional list NULL : do not set Other than NULL : pointer to next object
Details	Gets the next list object on the scroll bidirectional list held in the object.
File name	gwin.h

GCSLoadData

gwin.h

Outline	Load interpreter screen file
Syntax	int GCSLoadData(GBaseObject *pSelf, HGIO fp, GBaseWindowProperty *ptPropery, short fImport)
Argument	(i) GBaseObject *pSelf : object for operation (i) HGIO fp : file descriptor (i/o)GBaseWindowProperty *ptProperty : area storing the read property (i) short fImport : indicates whether to import or not
Return value	0 : normal end Other than 0 : error
Details	Reads the screen data from current position in interpreter screen file.
File name	gwin.h

GCSSetHWDCEnvironment

gwin.h

Outline	Set drawing environment
Syntax	void GCSSetHWDCEnvironment(GBaseObject *pSelf, HWDC hWDC)
Argument	(i) GBaseObject *pSelf : object for operation (i/o) HWDC hWDC : drawing handle
Return value	None
Details	Sets drawing environment to match object.
File name	gwin.h

GCSGetScreen

gwin.h

Outline	Get screen object
Syntax	GBaseObject* GCSGetScreen(GBaseObject *pSelf*)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	Pointer to screen object
Details	Gets screen object belonging to object.
File name	gwin.h

GCSSetGBaseObject

gwin.h

Outline	Set object management structure
Syntax	void GCSSetGBaseObject(GBaseObject *pSelf,* GBaseObject *pBaseObject*)
Argument	(i) GBaseObject *pSelf : object for operation (i) GBaseObject *pBaseObject : object management structure
Return value	FALSE : setting failed TRUE : setting succeeded
Details	Sets the management structure for the object being managed.
File name	gwin.h

GCSGetGBaseObject

gwin.h

Outline	Get object management structure
Syntax	GBaseObject* GCSGetGBaseObject(GBaseObject *pSelf)
Argument	(i) GBaseObject *pSelf : object for operation
Return value	NULL : management structure does not exist Other than NULL : pointer to management structure
Details	Gets the management structure for the object being managed.
File name	gwin.h

GCSFreeLoadData

gwin.h

Outline	Release memory acquired by LoadData
Syntax	int GCSFreeLoadData(GBaseObject *pSelf, GBaseWindowProperty *ptProperty)
Argument	(i) GBaseObject *pSelf : object for operation (i) GBaseWindowProperty *ptProperty : property
Return value	0 : normal end Other than 0 : error
Details	Releases the memory acquired by LoadData.
File name	gwin.h

GCSSetModalStatus

gwin.h

Outline	Set modal status
Syntax	void GCSSetModalStatus(GBaseObject **pWindow,* short *fStatus*)
Argument	(i) GBaseObject *pWindow : object for operation (i) short fStatus : window modal status FALSE : none TRUE : present
Return value	None
Details	Sets the modal window status.
File name	gwin.h

GCSGetModalStatus

gwin.h

Outline	Get modal window status
Syntax	short GCSGetModalStatus(GBaseObject **pWindow*)
Argument	(i) GBaseObject *pWindow : object for operation
Return value	FALSE : none TRUE : present
Details	Gets the modal window status.
File name	gwin.h

GCSGetRetValue

gwin.h

Outline	Get modal window return value
Syntax	short GCSGetRetValue(GBaseObject *pWindow)
Argument	(i) GBaseObject *pWindow : object for operation
Return value	Modal window return value
Details	Gets the return value returned when the modal window closes.
File name	gwin.h

GCSBeginModal

gwin.h

Outline	Start process for modal window
Syntax	void GCSBeginModal(GBaseObject *pWindow)
Argument	(i) GBaseObject *pWindow : object for operation
Return value	None
Details	Gives notice of the modal start for a window.
File name	gwin.h

GCSEndModal

gwin.h

Outline	End process for modal window
Syntax	void GCSEndModal(GBaseObject **pWindow,* short *nRetValue*)
Argument	(i) GBaseObject *pWindow : object for operation (i) short nRetValue : return value when modal window is closed
Return value	None
Details	Gives notice of the modal end for a window.
File name	gwin.h

GCSSetTitleBarStatus

gwin.h

Outline	Set title bar present/none status
Syntax	void GCSSetTitleBarStatus(GBaseObject **pWindow,* short *fStatus*)
Argument	(i) GBaseObject *pWindow : object for operation (i) short fStatus : title bar present/none status FALSE : none TRUE : present
Return value	None
Details	Sets the window title bar present or none. When switching on/off, the object area is registered as a redraw area.
File name	gwin.h

GCSGetTitleBarStatus

gwin.h

Outline	Get title bar present/none status
Syntax	short GCSGetTitleBarStatus(GBaseObject *pWindow)
Argument	(i) GBaseObject *pWindow : object for operation
Return value	FALSE : none TRUE : present
Details	Gets the title bar present/none status.
File name	gwin.h

GCSSetCloseButtonStatus

gwin.h

Outline	Set close button present/none status
Syntax	void GCSSetCloseButtonStatus(GBaseObject *pWindow, short fStatus)
Argument	(i) GBaseObject *pWindow : object for operation (i) short fStatus : close button present/none status FALSE : none TRUE : present
Return value	None
Details	Sets the window close button present or none. When switching present or none, the object area is registered as a redraw area.
File name	gwin.h

GCSGetCloseButtonStatus

gwin.h

Outline	Get close button present/none status
Syntax	short GCSGetCloseButtonStatus(GBaseObject *pWindow*)
Argument	(i) GBaseObject *pWindow : object for operation
Return value	FALSE : none TRUE : present
Details	Gets the close button present/none status.
File name	gwin.h

GCSSetEdgeStatus

gwin.h

Outline	Set window frame present/none status
Syntax	void GCSSetEdgeStatus(GBaseObject *pWindow,* short *fStatus*)
Argument	(i) GBaseObject *pWindow : object for operation (i) short fStatus : window frame present/none status FALSE : none TRUE : present
Return value	None
Details	Sets the window frame present or none. When switching present or none, the object area is registered as a redraw area.
File name	gwin.h

GCSGetEdgeStatus

gwin.h

Outline	Get window frame present/none status
Syntax	short GCSGetEdgeStatus(GBaseObject *pWindow*)
Argument	(i) GBaseObject *pWindow : object for operation
Return value	FALSE : none TRUE : present
Details	Gets the window frame present/none status.
File name	gwin.h

GCSAddChild

gwin.h

Outline	Add child object
Syntax	short GCSAddChild(GBaseObject *pPanel,* GBaseObject *pChild*)
Argument	(i) GBaseObject *pPanel : object for operation (i) GBaseObject *pChild : child object to be added
Return value	FALSE : addition failed TRUE : addition succeeded
Details	The child object specified by pChild is added to the child objects managed by the panel.
File name	gwin.h

GCSGetChildList

gwin.h

Outline	Get child object list
Syntax	HGBWL GCSGetChildList(GBaseObject *pPanel)
Argument	(i) GBaseObject *pPanel : object for operation
Return value	Child object list handle
Details	Gets child object list handle.
File name	gwin.h

GCSSetBackGroundDesign

gwin.h

Outline	Set background design
Syntax	void GCSSetBackGroundDesign(GBaseObject *pPanel, GDesign *pDesign)
Argument	(i) GBaseObject *pPanel : object for operation (i) GDesign* pDesign : background design
Return value	None
Details	Sets background design. After the background design is set, the panel area is registered as a redraw area.
File name	gwin.h

GCSGetBackGroundDesign

gwin.h

Outline	Get background design
Syntax	GDesign* GCSGetBackGroundDesign(GBaseObject *pPanel*)
Argument	(i) GBaseObject *pPanel : object for operation
Return value	Pointer to background design
Details	Gets pointer for background design.
File name	gwin.h

GCSCreateChildren

gwin.h

Outline	Process for creating child object
Syntax	short GCSCreateChildren(GBaseObject *pPanel*)
Argument	(i) GBaseObject *pPanel : object for operation
Return value	FALSE : creation failed TRUE : creation succeeded
Details	Creates child object managed by GCPanel object. No processing by GCPanel. Actual processing creates GCPanel in a derivative class.
File name	gwin.h

GCSDeleteAllChildren

gwin.h

Outline	Delete all child objects
Syntax	void GCSDeleteAllChildren(GBaseObject *pPanel*)
Argument	(i) GBaseObject *pPanel : object for operation
Return value	None
Details	Deletes all child objects.
File name	gwin.h

GCSGetChild

gwin.h

Outline	Search child object
Syntax	GBaseObject *GCSGetChild(GBaseObject *pPanel,* unsigned short *usID*)
Argument	(i) GBaseObject *pPanel : object for operation (i) unsigned short usID : ID for child object being searched
Return value	NULL : child object with specified ID was not searched. Other than NULL : pointer to child object with specified ID
Details	Searches for a GCPanel child object having the specified ID.
File name	gwin.h

GCSGetChildAll

gwin.h

Outline	Search child object search in frame
Syntax	GBaseObject *GCSGetChildAll(GBaseObject *pPanel, unsigned short usID)
Argument	(i) GBaseObject *pPanel : object for operation (i) unsigned short usID : ID for child object being searched
Return value	NULL : child object with specified ID was not searched. Other than NULL : pointer to child object with specified ID
Details	Searches for a GCPanel child object having the specified ID. When the child object is a frame, the search is also performed in the GCPanel for the page currently displayed in that frame.
File name	gwin.h

GCSSetActiveChild

gwin.h

Outline

Set active object

Syntax

```
short GCSSetActiveChild(
  GBaseObject    *pPanel,
  short          nSearch,
  GBaseObject    *pActiveChild
)
```

Argument

(i) GBaseObject *pPanel : object for operation
(i) short nSearch : method for specifying object
(i) GBaseObject *pChild : active child object

Return value

FALSE : setting failed
TRUE : setting succeeded

Details

When the panel status is active, the child object receiving the focus is set.
The setting methods are the method of directly specifying the object and the method of searching for the beginning and end of the child object list.
The following value is specified in nSearch.
GWINDOW_SEARCH_NON:

File name

gwin.h

GCSGetActiveChild

gwin.h

Outline	Get active object (only direct child object)
Syntax	GBaseObject* GCSGetActiveChild(GBaseObject *pPanel)
Argument	(i) GBaseObject *pPanel : object for operation
Return value	NULL : no setting Other than NULL : get active object
Details	When the panel status is active, the child object receiving the focus is acquired.
File name	gwin.h

GCSGetActiveControl

gwin.h

Outline	Get active control (including terminal child objects)
Syntax	GBaseObject* GCSGetActiveControl(GBaseObject *pPanel)
Argument	(i) GBaseObject *pPanel : object for operation
Return value	NULL : no setting Other than NULL : get active control
Details	When the panel status is active, the child object receiving the control is acquired.
File name	gwin.h

GCSSetBlinkOffInterval

gwin.h

Outline	Set blinking OFF interval
Syntax	void GCSSetBlinkOffInterval(GBaseObject *pPanel, long lInterval)
Argument	(i) GBaseObject *pPanel : object for operation (i) long lInterval : blinking OFF interval
Return value	None
Details	Sets the interval for blinking OFF.
File name	gwin.h

GCSGetBlinkOffInterval

gwin.h

Outline	Get blinking OFF interval
Syntax	long GCSGetBlinkOffInterval(GBaseObject *pPanel)
Argument	(i) GBaseObject *pPanel : object for operation
Return value	blinking OFF interval
Details	Gets the interval for blinking OFF.
File name	gwin.h

GCSSetBlinkOnInterval

gwin.h

Outline	Set blinking ON interval
Syntax	void GCSSetBlinkOnInterval(GBaseObject *pPanel, long lInterval)
Argument	(i) GBaseObject *pPanel : object for operation (i) long lInterval : blinking ON interval
Return value	None
Details	Sets the interval for blinking ON.
File name	gwin.h

GCSGetBlinkOnInterval

gwin.h

Outline	Get blinking ON interval
Syntax	long GCSGetBlinkOnInterval(GBaseObject *pPanel)
Argument	(i) GBaseObject *pPanel : object for operation
Return value	Blinking ON interval
Details	Gets the interval for blinking ON.
File name	gwin.h

GCSAddBlinkControl

gwin.h

Outline	Add caption blink control
Syntax	void GCSAddBlinkControl(GBaseObject **pPanel,* GBaseObject **pControl*)
Argument	(i) GBaseObject *pPanel : object for operation (i) GBaseObject *pControl : control to be added
Return value	None
Details	Among the controls in the panel, the control for caption blinking is added to the list.
File name	gwin.h

GCSRemoveBlinkControl

gwin.h

Outline	Delete caption blink control
Syntax	void GCSRemoveBlinkControl(GBaseObject **pPanel,* GBaseObject **pControl*)
Argument	(i) GBaseObject *pPanel : object for operation (i) GBaseObject *pControl : control to be deleted
Return value	None
Details	Among the controls in the panel, the control for terminating caption blink is deleted from the list.
File name	gwin.h

GCSAddScrollControl

gwin.h

Outline	Add caption scroll control
Syntax	void GCSAddScrollControl(GBaseObject **pPanel,* GBaseObject **pControl*)
Argument	(i) GBaseObject *pPanel : object for operation (i) GBaseObject *pControl : control to be added
Return value	None
Details	Among the controls in the panel, the control for caption scrolling is added to the list.
File name	gwin.h

GCSRemoveScrollControl

gwin.h

Outline	Delete caption scroll control
Syntax	void GCSRemoveScrollControl(GBaseObject **pPanel,* GBaseObject **pControl*)
Argument	(i) GBaseObject *pPanel : object for operation (i) GBaseObject *pControl : control to be deleted
Return value	None
Details	Among the controls in the panel, the control for terminating caption scroll is deleted from the list.
File name	gwin.h

GCSDrawChildren

gwin.h

Outline	Child object drawing process
Syntax	void GCSDrawChildren(GBaseObject **pPanel,* HWDC *pDC*)
Argument	(i) GBaseObject *pPanel : object for operation (i) HWDC pDC : drawing context
Return value	None
Details	Carries out child object drawing process. The drawing order is in order from the beginning of the list.
File name	gwin.h

GCSAttachPanel

gwin.h

Outline	Attach panel object
Syntax	GBaseObject *GCSAttachPanel(GBaseObject **pScreen,* GBaseObject **pNewPanel*)
Argument	(i) GBaseObject *pScreen : object for operation (i) GBaseObject *pNewPanel : related GCPanel class
Return value	Pointer to related panel object
Details	Attaches panel object to screen/window.
File name	gwin.h

GCSGetPanel

gwin.h

Outline	Get attached GCPanel object
Syntax	GBaseObject *GCSGetPanel(GBaseObject *pScreen*)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	Pointer to related panel
Details	Carries out acquisition of GCPanel object related to screen/window.
File name	gwin.h

GCSSetSystemFontID

gwin.h

Outline	Set system font resource ID
Syntax	void GCSSetSystemFontID(GBaseObject *pScreen,* unsigned short *usID*)
Argument	(i) GBaseObject *pScreen : object for operation (i) unsigned short usID : font resource ID
Return value	None
Details	Sets system font resource ID. After the system font resource ID is set, the entire screen is registered as a redraw area.
File name	gwin.h

GCSGetSystemFontID

gwin.h

Outline	Get system font resource ID
Syntax	unsigned short GCSGetSystemFontID(GBaseObject *pScreen)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	Font resource ID
Details	Gets system font resource ID.
File name	gwin.h

GCSSetSystemBorderID

gwin.h

Outline	Set system 3D border resource ID
Syntax	void GCSSetSystemBorderID(GBaseObject *pScreen, unsigned short usID)
Argument	(i) GBaseObject *pScreen : object for operation (i) unsigned short usID : 3D border resource ID
Return value	None
Details	Sets system 3D border resource ID. After the system 3D border resource ID is set, the entire screen is registered as a redraw area.
File name	gwin.h

GCSGetSystemBorderID

gwin.h

Outline	Get system 3D border resource ID
Syntax	unsigned short GCSGetSystemBorderID(GBaseObject *pScreen)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	3D border resource ID
Details	Gets system 3D border resource ID setting.
File name	gwin.h

GCSSetSystemForeColor

gwin.h

Outline	Set system foreground color
Syntax	void GCSSetSystemForeColor(GBaseObject *pScreen, GColor gcColor)
Argument	(i) GBaseObject *pScreen : object for operation (i) GColor gcColor : foreground color
Return value	None
Details	Sets system foreground color. After the system foreground color is set, the entire screen is registered as a redraw area.
File name	gwin.h

GCSGetSystemForeColor

gwin.h

Outline	Get system foreground color
Syntax	GColor GCSGetSystemForeColor(GBaseObject *pScreen)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	System foreground color
Details	Gets system foreground color.
File name	gwin.h

GCSSetSystemBackColor

gwin.h

Outline	Set system background color
Syntax	void GCSSetSystemBackColor(GBaseObject *pScreen, GColor gcColor)
Argument	(i) GBaseObject *pScreen : object for operation (i) GColor gcColor : background color
Return value	None
Details	Sets system background color. After the system background color is set, the entire screen is registered as a redraw area.
File name	gwin.h

GCSGetSystemBackColor

gwin.h

Outline	Get system background color
Syntax	GColor GCSGetSystemBackColor(GBaseObject *pScreen*)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	System background color
Details	Gets system background color.
File name	gwin.h

GCSSetSystemActiveColor

gwin.h

Outline	Set system active color
Syntax	void GCSSetSystemActiveColor(GBaseObject *pScreen,* GColor *gcColor*)
Argument	(i) GBaseObject *pScreen : object for operation (i) GColor gcColor : active color
Return value	None
Details	Sets system active color. After the system active color is set, the entire screen is registered as a redraw area.
File name	gwin.h

GCSGetSystemActiveColor

gwin.h

Outline	Get system active color
Syntax	GColor GCSGetSystemActiveColor(GBaseObject *pScreen*)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	System active color
Details	Gets system active color.
File name	gwin.h

GCSSetSystemInactiveColor

gwin.h

Outline	Set system inactive color
Syntax	void GCSSetSystemInactiveColor(GBaseObject *pScreen,* GColor *gcColor*)
Argument	(i) GBaseObject *pScreen : object for operation (i) GColor gcColor : inactive color
Return value	None
Details	Sets system inactive color. After the system inactive color is set, the entire screen is registered as a redraw area.
File name	gwin.h

GCSGetSystemInactiveColor

gwin.h

Outline	Get system inactive color
Syntax	GColor GCSGetSystemInactiveColor(GBaseObject *pScreen*)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	System inactive color
Details	Gets system inactive color.
File name	gwin.h

GCSGetFrame

gwin.h

Outline	Get frame
Syntax	GBaseObject *GCSGetFrame(GBaseObject *pScreen*)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	Pointer to frame managed by screen
Details	Gets pointer to frame being managed by the screen.
File name	gwin.h

GCSAddWindow

gwin.h

Outline	Add window
Syntax	short GCSAddWindow(GBaseObject *pScreen, GBaseObject *pWindow)
Argument	(i) GBaseObject *pScreen : object for operation (i) GBaseObject *pWindow : window object to be added
Return value	FALSE : addition failed TRUE : addition succeeded
Details	Add window to the frame on the screen.
File name	gwin.h

GCSChangeFocus

gwin.h

Outline	Change focus
Syntax	void GCSChangeFocus(GBaseObject *pScreen, GBaseObject *pNewObject)
Argument	(i) GBaseObject *pScreen : object for operation (i) GBaseObject *pNewObject : object acquiring the focus
Return value	None
Details	Focus given to specified object.
File name	gwin.h

GCSGetFocus

gwin.h

Outline	Get focus
Syntax	GBaseObject *GCSGetFocus(GBaseObject *pScreen)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	NULL : no focus Other than NULL : object having focus
Details	Gets the pointer to the object having the focus.
File name	gwin.h

GCSDispatchMessage

gwin.h

Outline	Event dispatch process
Syntax	long GCSDispatchMessage(GBaseObject *pScreen, GEventMessage *gemMsg)
Argument	(i) GBaseObject *pScreen : object for operation (i) GEventMessage *gemMsgt : event message
Return value	Supported event return values
Details	Calls a message procedure for each event according to the event.
File name	gwin.h

GCSGetWDC

gwin.h

Outline	Get drawing context
Syntax	HWDC GCSGetWDC(GBaseObject *pScreen)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	Pointer to drawing context
Details	Gets drawing context being managed by the screen.
File name	gwin.h

GCSGetHWDC

gwin.h

Outline	Get drawing context
Syntax	HWDC GCSGetHWDC(GBaseObject *pScreen)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	Pointer to drawing context
Details	Gets drawing context being managed by the screen.
File name	gwin.h

GCSSetPanelPlane

gwin.h

Outline	Set memory area panel plane management information
Syntax	void GCSSetPanelPlane(GBaseObject *pScreen, unsigned char ucIndex, unsigned char ucFlag)
Argument	(i) GBaseObject *pScreen : object for operation (i) unsigned char ucIndex : panel plane No. (i) unsigned char ucFlag : usage status (FALSE: unused, TRUE: used)
Return value	None
Details	Sets usage status for the panel plane in the memory area.
File name	gwin.h

GCSGetPanelPlane

gwin.h

Outline	Get memory area panel plane management information
Syntax	unsigned char GCSGetPanelPlane(GBaseObject *pScreen, unsigned char ucIndex)
Argument	(i) GBaseObject *pScreen : object for operation (i) unsigned char ucIndex : panel plane No.
Return value	Usage status FALSE : unused TRUE : used
Details	Gets usage status for the panel plane in the memory area.
File name	gwin.h

GCSSetWindowPlane

gwin.h

Outline	Set memory area window plane management information
Syntax	void GCSSetWindowPlane(GBaseObject **pScreen,* unsigned char *ucIndex,* unsigned char *ucFlag*)
Argument	(i) GBaseObject *pScreen : object for operation (i) unsigned char ucIndex : window plane No. (i) unsigned char ucFlag : usage status (FALSE: unused, TRUE: used)
Return value	None
Details	Sets usage status for the panel plane in the memory area.
File name	gwin.h

GCSGetWindowPlane

gwin.h

Outline	Get memory area window plane management information
Syntax	unsigned char GCSGetWindowPlane(GBaseObject **pScreen,* unsigned char *ucIndex*)
Argument	(i) GBaseObject *pScreen : object for operation (i) unsigned char ucIndex : panel plane No.
Return value	Usage status FALSE : unused TRUE : used
Details	Gets usage status for the window plane in the memory area.
File name	gwin.h

GCSSetExitStatus

gwin.h

Outline	Set exit information
Syntax	void GCSSetExitStatus(GBaseObject **pScreen,* short *fStatus*)
Argument	(i) GBaseObject *pScreen : object for operation (i) short fStatus : exit information (TRUE: exit)
Return value	None
Details	Sets the exit information for applications using the GUI library.
File name	gwin.h

GCSGetExitStatus

gwin.h

Outline	Get exit information
Syntax	short GCSGetExitStatus(GBaseObject **pScreen*)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	Exit information FALSE : in operation TRUE : exit
Details	Gets the exit information for applications using the GUI library.
File name	gwin.h

GCSSetSystemStatus

gwin.h

Outline	Set system information
Syntax	void GCSSetSystemStatus(GBaseObject *pScreen, unsigned long ulSystem, short fStatus)
Argument	(i) GBaseObject *pScreen : object for operation (i) unsigned long ulSystem : system flag (i) short fStatus : status (0: FALSE, 1: TRUE)
Return value	None
Details	Sets system status specified by ulSystem.
File name	gwin.h

GCSGetSystemStatus

gwin.h

Outline	Get system information
Syntax	short GCSGetSystemStatus(GBaseObject *pScreen, unsigned long ulSystem)
Argument	(i) GBaseObject *pScreen : object for operation (i) unsigned long ulSystem : system flag
Return value	Status 0 : FALSE 1 : TRUE
Details	Gets system status specified by ulSystem.
File name	gwin.h

GCSDoModal

gwin.h

Outline	Window modal display
Syntax	short GCSDoModal(GBaseObject *pScreen, GBaseObject *pWindow, long lLParam, long lUParam)
Argument	(i) GBaseObject *pScreen : object for operation (i) GBaseObject *pWindow : window object (i) long lLParam : additional information (lower-side) (i) long lUParam : additional information (upper-side)
Return value	Modal window return value
Details	Displays the window object specified by pWindow as a modal window.
File name	gwin.h

GCSSetPaletteStatus

gwin.h

Outline	Set system palette utilization status
Syntax	void GCSSetPaletteStatus(GBaseObject *pScreen, short nStatus)
Argument	(i) GBaseObject *pScreen : object for operation (i) short nStatus : palette usage status
Return value	None
Details	Sets the usage status of the palette. When FALSE is set, unused is the status. When TRUE is set, used is the status.
File name	gwin.h

GCSGetPaletteStatus

gwin.h

Outline	Get system palette utilization status
Syntax	short GCSGetPaletteStatus(GBaseObject *pScreen*)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	System palette usage status FALSE : unused TRUE : used
Details	Gets the usage status of the palette.
File name	gwin.h

GCSSetSystemPaletteID

gwin.h

Outline	Set system palette resource ID
Syntax	void GCSSetSystemPaletteID(GBaseObject *pScreen,* unsigned short *usID*)
Argument	(i) GBaseObject *pScreen : object for operation (i) unsigned short usID : palette resource ID
Return value	None
Details	Sets the palette resource ID used by the system palette.
File name	gwin.h

GCSGetSystemPaletteID

gwin.h

Outline	Get system palette resource ID
Syntax	unsigned short GCSGetSystemPaletteID(GBaseObject *pScreen)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	System palette resource ID
Details	Gets the palette resource ID used by the system palette.
File name	gwin.h

GCSSetPaletteColors

gwin.h

Outline	Set number of colors in system palette
Syntax	void GCSSetPaletteColors(GBaseObject *pScreen, unsigned short usPaletteColors)
Argument	(i) GBaseObject *pScreen : object for operation (i) unsigned short usPaletteColors : number of colors used by system palette
Return value	None
Details	Sets the number of colors used by the system palette.
File name	gwin.h

GCSGetPaletteColors

gwin.h

Outline	Get number of colors in system palette
Syntax	unsigned short GCSGetPaletteColors(GBaseObject *pScreen)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	Number of colors used by system palette
Details	Gets the number of colors used by the system palette.
File name	gwin.h

GCSAddTimerObject

gwin.h

Outline	Add object to system timer list
Syntax	void GCSAddTimerObject(GBaseObject *pScreen, GBaseObject *pObject)
Argument	(i) GBaseObject *pScreen : object for operation (i) GBaseObject *pObject : object to be added
Return value	None
Details	Adds the object specified by pObject to the system timer list. When registered into this system timer list, the object's function is call at a constant frequency with usTimerID==GCSCREEN_BASE_TIMER_ID.
File name	gwin.h

GCSRemoveTimerObject

gwin.h

Outline	Delete object from system timer list
Syntax	void GCSRemoveTimerObject(GBaseObject *pScreen, GBaseObject *pObject)
Argument	(i) GBaseObject *pScreen : object for operation (i) GBaseObject *pObject : object to be deleted
Return value	None
Details	Deletes the object specified by pObject from the system timer list.
File name	gwin.h

GCSGetWindow

gwin.h

Outline	Get window
Syntax	GBaseObject* GCSGetWindow(GBaseObject *pScreen, nsigned short usID)
Argument	(i) GBaseObject *pScreen : object for operation (i) unsigned short usID : ID of window being acquired
Return value	NULL : window for specified ID does not exist Other than NULL : window object having specified ID
Details	Gets the window having the ID specified by usID.
File name	gwin.h

GCSGetActiveWindow

gwin.h

Outline	Get active window
Syntax	GBaseObject* GCSGetActiveWindow(GBaseObject *pScreen)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	NULL : active window does not exist Other than NULL : active window object
Details	Gets the window having the ID specified by usID.
File name	gwin.h

GCSShowPanel

gwin.h

Outline	Display panel switching process
Syntax	short GCSShowPanel(GBaseObject *pScreen, long lPage)
Argument	(i) GBaseObject *pScreen : object for operation (i) long lPage : screen No. to be displayed
Return value	FALSE : switching failed TRUE : switching succeeded
Details	Switched to the page specifying the panel currently displayed in the object.
File name	gwin.h

GCSCreateInstance

gwin.h

Outline

Create instance process

Syntax

```
GBaseObject *GCSCreateInstance(
  GBaseObject                 *pScreen,
  unsigned short              usType,
  const GBaseWindowProperty   *pProperty,
  GBaseObject                 *pParent
)
```

Argument

(i) GBaseObject *pScreen	: object for operation
(i) unsigned short usType	: object type
(i) const GBaseWindowProperty *pImport	: object property
(i) GBaseObject *pParent	: parent object

Return value

NULL	: creation failed
Other than NULL	: created object

Details

Creates instance of object registered in the project.

File name

gwin.h

GCSPageLogOperation

gwin.h

Outline	Switch screen according to screen history
Syntax	short GCSPageLogOperation(GBaseObject *pScreen, short nOperation, short nShowWindow)
Argument	(i) GBaseObject *pScreen : object for operation (i) short nOperation : screen transition method (i) short nShowWindow : window recovery status
Return value	FALSE : failed TRUE : succeeded
Details	Switches screens according to the screen history.
File name	gwin.h

GCSDrawFrame

gwin.h

Outline	Related panel drawing process
Syntax	void GCSDrawFrame(GBaseObject *pScreen, HWDC pDC)
Argument	(i) GBaseObject *pScreen : object for operation (i) HWDC pDC : drawing context
Return value	None
Details	Carries out drawing process for frame object related to GCScreen object.
File name	gwin.h

GCSDrawPanel

gwin.h

Outline	Related GCPanel object drawing process
Syntax	void GCSDrawPanel(GBaseObject **pScreen,* HWDC *pDC*)
Argument	(i) GBaseObject *pScreen : object for operation (i) HWDC pDC : drawing context
Return value	None
Details	Carries out drawing process for panel object related to GCScreen object.
File name	gwin.h

GCSOnDrawWindow

gwin.h

Outline

Window drawing process

Syntax

```
void GCSOnDrawWindow(
  GBaseObject   *pScreen,
  HWDC          pDC,
  long          lStart,
  long          lEnd,
  long          lNow,
  GBaseObject   **pWinArray,
  GRect         *pClip
)
```

Argument

(i) GBaseObject *pScreen	: object for operation
(i) HWDC pDC	: drawing context
(i) long lStart	: depth of starting window
(i) long lEnd	: depth of ending window
(i) long lNow	: depth of current window
(i) GBaseObject **pWinArray	: drawing window array
(i) GRect *pClip	: drawing area

Return value

None

Details

Carries out drawing process for object client area.
Calls related GCPanel object drawing process in GCScreen object.

File name

gwin.h

GCSSetSystemPalette

gwin.h

Outline	Set system palette
Syntax	void GCSSetSystemPalette(GBaseObject *pScreen)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	None
Details	Carries out system palette setup according to the system palette settings. When no palette is used, ends without making palette settings. When the number of colors differs in the palette resource and system palette settings, the lower number of colors is set.
File name	gwin.h

GCSDrawBackGround

gwin.h

Outline	Background drawing process
Syntax	void GCSDrawBackGround(GBaseObject *pScreen, HWDC pDC)
Argument	(i) GBaseObject *pScreen : object for operation (i) HWDC pDC : drawing context
Return value	None
Details	Carries out drawing of GCScreen object background.
File name	gwin.h

GCSCreateGWindow

gwin.h

Outline	Create window
Syntax	short GCSCreateGWindow(GBaseObject *pScreen, long lPage)
Argument	(i) GBaseObject *pScreen : object for operation (i) long lPage : display specified window
Return value	FALSE : display failed TRUE : display succeeded
Details	Creates a window and displays it.
File name	gwin.h

GCSSetScreenManager

gwin.h

Outline	Set screen manager
Syntax	void GCSSetScreenManager(GBaseObject *pScreen, GBaseObject *pScreenManager)
Argument	(i) GBaseObject *pScreen : object for operation (i) GBaseObject *pScreenManager : screen manager object
Return value	None
Details	Sets up screen manager in screen objects for each module.
File name	gwin.h

GCSSetModuleID

gwin.h

Outline	Set module ID
Syntax	void GCSSetModuleID(GBaseObject **pScreen,* unsigned long *ulModuleID*)
Argument	(i) GBaseObject *pScreen : object for operation (i) unsigned long ulModuleID : module ID
Return value	None
Details	Sets up ID in screen objects for each module.
File name	gwin.h

GCSClearPageLog

gwin.h

Outline	Clear screen history
Syntax	void GCSClearPageLog(GBaseObject **pScreen*)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	None
Details	Clears the screen history.
File name	gwin.h

GCSAddPageLog

gwin.h

Outline	Add to screen history
Syntax	short GCSAddPageLog(GBaseObject *pScreen, long lPage)
Argument	(i) GBaseObject *pScreen : object for operation (i) long lPage : screen No. to be added
Return value	TRUE : addition succeeded FALSE : addition failed
Details	Adds to the screen history.
File name	gwin.h

GCSPrevPage

gwin.h

Outline	Issue request for displaying previous screen
Syntax	short GCSPrevPage(GBaseObject *pScreen, short nShowWindow)
Argument	(i) GBaseObject *pScreen : object for operation (i) short nShowWindow : restore window status FALSE : do not restore TRUE : restore
Return value	TRUE : succeeded FALSE : failed
Details	Issues a request to display the screen displayed previously based on the screen history information.
File name	gwin.h

GCSNextPage

gwin.h

Outline	Issue request for displaying next screen
Syntax	short GCSNextPage(GBaseObject *pScreen, short nShowWindow)
Argument	(i) GBaseObject *pScreen : object for operation (i) short nShowWindow : restore window status FALSE : do not restore TRUE : restore
Return value	TRUE : succeeded FALSE : failed
Details	Issues a request to display the screen displayed afterward based on the screen history information.
File name	gwin.h

GCSSetPageOffset

gwin.h

Outline	Set module page offset
Syntax	void GCSSetPageOffset(GBaseObject *pScreen, long lOffset)
Argument	(i) GBaseObject *pScreen : object for operation (i) long lOffset : page offset
Return value	None
Details	Sets the page offset for the module to which the screen belongs.
File name	gwin.h

GCSGetPageOffset

gwin.h

Outline	Get module page offset
Syntax	long GCSGetPageOffset(GBaseObject *pScreen)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	Page offset
Details	Gets the page offset for the module to which the screen belongs.
File name	gwin.h

GCSGetOperationLog

gwin.h

Outline	Get operation log
Syntax	short GCSGetOperationLog(GBaseObject *pScreen, unsigned short usIndex, GOperationInformation *pLog)
Argument	(i) GBaseObject *pScreen : object for operation (i) unsigned short usIndex : index No. (i) GOperationInformation *pLog : variable storing operation log being acquired
Return value	FALSE : failed TRUE : succeeded
Details	Gets the operation log in the position specified by usIndex.
File name	gwin.h

GCSClearOperationLog

gwin.h

Outline	Clear operation log
Syntax	short GCSClearOperationLog(GBaseObject *pScreen*)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	FALSE : failed TRUE : succeeded
Details	Clears the content of the operation log.
File name	gwin.h

GCSAddOperationLog

gwin.h

Outline	Add operation log (operation event)
Syntax	short GCSAddOperationLog(GBaseObject *pScreen,* long *lPage,* unsigned short *usID,* unsigned short *usMessage*)
Argument	(i) GBaseObject *pScreen : object for operation (i) long lPage : screen No. (i) unsigned short usID : object ID (i) unsigned short usMessage : event
Return value	FALSE : failed TRUE : succeeded
Details	Adds the operation log with the specified log content.
File name	gwin.h

GCSAddChangePageLog

gwin.h

Outline	Add operation log (screen switching)
Syntax	short GCSAddChangePageLog(GBaseObject *pScreen, long lOldPage, long lNewPage)
Argument	(i) GBaseObject *pScreen : object for operation (i) long lOldPage : screen No. prior to screen switching (i) long lNewPage : screen No. following screen switching
Return value	FALSE : failed TRUE : succeeded
Details	Adds the operation log with the specified log content.
File name	gwin.h

GCSGetPageNumber

gwin.h

Outline	Get screen No.
Syntax	long GCSGetPageNumber(GBaseObject *pScreen)
Argument	(i) GBaseObject *pScreen : object for operation
Return value	Currently displayed screen No.
Details	Returns the currently displayed screen No.
File name	gwin.h

GCSLoadProjectFile

gwin.h

Outline	Load interpreter project file
Syntax	int GCSLoadProjectFile(GBaseObject *pIScreen, GTCHAR *ptszProjectPath, short fShowStartPanel)
Argument	(i) GBaseObject *pIScreen; : object for operation (i) GTCHAR *ptszProjectPath : project file to be loaded (i) short fShowStartPanel : display/non-display of initial display page FALSE : non-display TRUE : display
Return value	0 : normal end Other than 0 : error
Details	Reads project data from the interpreter project file.
File name	gwin.h

GCSLoadGWindow

gwin.h

Outline	Window loading process
Syntax	GBaseObject* GCSLoadGWindow(GBaseObject *pScreen, long lPage)
Argument	(i) GBaseObject *pScreen : object for operation (i) long lPage : window No. to load
Return value	NULL : load failed Other than NULL : load succeeded
Details	Loads the window with the specified No.
File name	gwin.h

GCSModalGWindow

gwin.h

Outline	Modal window process
Syntax	short GCSModalGWindow(GBaseObject *pScreen, long lPage)
Argument	(i) GBaseObject *pScreen : object for operation (i) long lPage : modal window No.
Return value	Modal window return value
Details	Displays the window with the number specified as a modal window.
File name	gwin.h

GCSSetBlinkType

gwin.h

Outline	Set blink type
Syntax	void GCSSetBlinkType(GBaseObject **pControl,* unsigned char *ucBlinkType*)
Argument	(i) GBaseObject *pControl : object for operation (i) unsigned char ucBlinkType GCBLINK_CHANGE_VISIBLE_STRING : display/non-display of character string GCBLINK_CHANGE_STRING_COLOR : change character color GCBLINK_CHANGE_WHOLE_COLOR : change whole color
Return value	None
Details	Sets the object blink type.
File name	gwin.h

GCSGetBlinkType

gwin.h

Outline	Get blink type
Syntax	unsigned char GCSGetBlinkType(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	Blink type: GCBLINK_CHANGE_VISIBLE_STRING : display/non-display of character string GCBLINK_CHANGE_STRING_COLOR : change character color GCBLINK_CHANGE_WHOLE_COLOR : change whole color
Details	Gets the object blink type.
File name	gwin.h

GCSSetBlinkEnableStatus

gwin.h

Outline	Set blink operation
Syntax	void GCSSetBlinkEnableStatus(GBaseObject *pControl, unsigned char ucEnable)
Argument	(i) GBaseObject *pControl : object for operation (i) unsigned char ucEnable : blink operation setting GCBLINK_BLINK_DISABLE : disable GCBLINK_BLINK_ENABLE : enable
Return value	None
Details	Sets the object blink operation.
File name	gwin.h

GCSGetBlinkEnableStatus

gwin.h

Outline	Get blink operation
Syntax	unsigned char GCSGetBlinkEnableStatus(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	Setup of blink operation FALSE : disable TRUE : enable
Details	Gets the object blink operation.
File name	gwin.h

GCSStartCaptionBlink

gwin.h

Outline	Start blink operation
Syntax	void GCSStartCaptionBlink(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	None
Details	Starts the object blink operation.
File name	gwin.h

GCSStopCaptionBlink

gwin.h

Outline	Stop blink operation
Syntax	void GCSStopCaptionBlink(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	None
Details	Stops the object blink operation.
File name	gwin.h

GCSSetCaptionScrollEnable

gwin.h

Outline	Set scroll operation
Syntax	void GCSSetCaptionScrollEnable(GBaseObject *pControl, unsigned char ucEnable)
Argument	(i) GBaseObject *pControl : object for operation (i) unsigned char ucEnable : scroll operation setting GCSCROLL_SCROLL_DISABLE : disable GCSCROLL_SCROLL_ENABLE : enable GCSCROLL_SCROLL_OUTOFBOUNDS : enable when out of bounds
Return value	None
Details	Sets the object scroll operation.
File name	gwin.h

GCSGetCaptionScrollEnable

gwin.h

Outline	Get scroll operation
Syntax	unsigned char GCSGetCaptionScrollEnable(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	Scroll operation GCSCROLL_SCROLL_DISABLE : disable GCSCROLL_SCROLL_ENABLE : enable GCSCROLL_SCROLL_OUTOFBOUNDS : enable when out of bounds
Details	Gets the object scroll operation.
File name	gwin.h

GCSStartCaptionScroll

gwin.h

Outline	Start scroll operation
Syntax	void GCSStartCaptionScroll(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	None
Details	Starts the object scroll operation. The scroll operation must be set in GCSCROLL_SCROLL_ENABLE or GCSCROLL_SCROLL_OUTOFBOUNDS in advance.
File name	gwin.h

GCSStopCaptionScroll

gwin.h

Outline	Stop scroll operation
Syntax	void GCSStopCaptionScroll(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	None
Details	Stops the object scroll operation.
File name	gwin.h

GCSPauseCaptionScroll

gwin.h

Outline	Pause scroll operation
Syntax	void GCSPauseCaptionScroll(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	None
Details	Pauses the object scroll operation.
File name	gwin.h

GCSRestartCaptionScroll

gwin.h

Outline	Restart scroll operation
Syntax	void GCSRestartCaptionScroll(GBaseObject **pControl*)
Argument	(i) GBaseObject *pControl : object for operation
Return value	None
Details	Restart the object scroll operation.
File name	gwin.h

GCSSetCaptionScrollStatus

gwin.h

Outline	Set scroll operation status
Syntax	void GCSSetCaptionScrollStatus(GBaseObject **pControl,* unsigned char *ucStatus*)
Argument	(i) GBaseObject *pControl : object for operation (i) unsigned char ucStatus : scroll operation status
Return value	None
Details	Sets the object scroll operation status.
File name	gwin.h

GCSGetCaptionScrollStatus

gwin.h

Outline	Get scroll operation status
Syntax	unsigned char GCSGetCaptionScrollStatus(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	Scroll operation status
Details	Gets the object scroll operation status.
File name	gwin.h

GCSIsCaptionOutOfBounds

gwin.h

Outline	Confirm overflowing of caption character string
Syntax	short GCSIsCaptionOutOfBounds(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	Out of bounds status FALSE : not out of bounds TRUE : out of bounds
Details	Confirms whether or not the caption character string set in the control goes out of the bounds of the control caption display area.
File name	gwin.h

GCSSetCaptionScrollDelayTime

gwin.h

Outline	Set scroll delay time
Syntax	void GCSSetCaptionScrollDelayTime(GBaseObject *pControl, unsigned long ITime)
Argument	(i) GBaseObject *pControl : object for operation (i) unsigned long ulTime : delay time setting
Return value	None
Details	Sets the delay time for the object scroll starting time.
File name	gwin.h

GCSGetCaptionScrollDelayTime

gwin.h

Outline	Get scroll delay time
Syntax	unsigned long GCSGetCaptionScrollDelayTime(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	Delay time
Details	Gets the delay time for the object scroll starting time.
File name	gwin.h

GCSSetCaptionScrollRefreshTime

gwin.h

Outline	Set scroll refresh time
Syntax	void GCSSetCaptionScrollRefreshTime(GBaseObject **pControl,* unsigned long *lTime*)
Argument	(i) GBaseObject *pControl : object for operation (i) unsigned long ulTime : refresh time setting
Return value	None
Details	Sets the object scroll refresh time.
File name	gwin.h

GCSGetCaptionScrollRefreshTime

gwin.h

Outline	Get scroll refresh time
Syntax	unsigned long GCSGetCaptionScrollRefreshTime(GBaseObject **pControl*)
Argument	(i) GBaseObject *pControl : object for operation
Return value	Refresh time
Details	Gets the object scroll refresh time.
File name	gwin.h

GCSSetCaptionScrollMovementValue

gwin.h

Outline	Set scroll movement value
Syntax	void GCSSetCaptionScrollMovementValue(GBaseObject **pControl,* unsigned short *usMovementValue*)
Argument	(i) GBaseObject *pControl : object for operation (i) unsigned short usMovementValue : movement value setting
Return value	None
Details	Sets the object scroll movement value.
File name	gwin.h

GCSGetCaptionScrollMovementValue

gwin.h

Outline	Get scroll movement value
Syntax	unsigned short GCSGetCaptionScrollMovementValue(GBaseObject **pControl*)
Argument	(i) GBaseObject *pControl : object for operation
Return value	Movement value
Details	Gets the object scroll movement value.
File name	gwin.h

GCSSetCaptionScrollStartPosition

gwin.h

Outline	Set scroll start position
Syntax	void GCSSetCaptionScrollStartPosition(GBaseObject *pControl, unsigned char ucStartPosition)
Argument	(i) GBaseObject *pControl : object for operation (i) unsigned char ucStartPosition : start position setting GCSCROLL_START_NOW : Follow current caption setting. GCSCROLL_START_RIGHT : From right.
Return value	None
Details	Sets the object scroll start position.
File name	gwin.h

GCSGetCaptionScrollStartPosition

gwin.h

Outline	Get scroll start position
Syntax	unsigned char GCSGetCaptionScrollStartPosition(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	Start position GCSCROLL_START_NOW : Follow current caption setting. GCSCROLL_START_RIGHT : From right.
Details	Gets the object scroll start position.
File name	gwin.h

GCSSetCaptionScrollPosition

gwin.h

Outline	Set total scroll movement value
Syntax	void GCSSetCaptionScrollPosition(GBaseObject *pControl, long lPosition)
Argument	(i) GBaseObject *pControl : object for operation (i) unsigned short usMovementValue : total movement value setting
Return value	None
Details	Sets the total scroll movement value from the start position for the object.
File name	gwin.h

GCSGetCaptionScrollPosition

gwin.h

Outline	Get total scroll movement value
Syntax	long GCSGetCaptionScrollPosition(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	Total movement value
Details	Gets the total scroll movement value from the start position for the object.
File name	gwin.h

GCSUpdateScroll

gwin.h

Outline	Scroll update process
Syntax	void GCSUpdateScroll(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	None
Details	Carries out object scroll update process.
File name	gwin.h

GCSIsCaptionScrollFinish

gwin.h

Outline	Confirm caption scroll completion
Syntax	short GCSIsCaptionScrollFinish(GBaseObject *pControl)
Argument	(i) GBaseObject *pControl : object for operation
Return value	Scroll finish status FALSE : scrolling TRUE : scrolling finish
Details	Confirms whether the caption character string set in the control is being scrolled.
File name	gwin.h

GCSCallBackMessage

gwin.h

Outline	Call back message execution process
Syntax	long GCSCallBackMessage(GBaseObject *pControl, unsigned short usMessage, long lLParam, long lUParam)
Argument	(i) GBaseObject *pControl : object for operation (i) unsigned short usMessage : call back message type (i) long lLParam : appended information (lower-side) (i) long lUParam : appended information (upper-side)
Return value	Arbitrary
Details	For a parent class managing the object, executes call-back processing prepared for an object ID.
File name	gwin.h

GCSLoadScreenFile

gwin.h

Outline	Load interpreter screen file
Syntax	int GCSLoadScreenFile(GBaseObject *pIScreen, long nPageNo)
Argument	(i) GBaseObject *pIScreen : object for operation (i) long nPageNo : screen No. to load
Return value	0 : normal end Other than 0 : error
Details	Reads the interpreter screen file.
File name	gwin.h

GCSSetPenColor

gwin.h

Outline	Set line color
Syntax	void GCSSetPenColor(GBaseObject *pShape, GColor gcColor)
Argument	(i) GBaseObject *pShape : object for operation (i) GColor gcColor : line color
Return value	None
Details	Sets the line color for the drawing/line drawing. After the line color is set, the drawing/line drawing area is registered as a redraw area.
File name	gwin.h

GCSGetPenColor

gwin.h

Outline	Get line color
Syntax	GColor GCSGetPenColor(GBaseObject *pShape)
Argument	(i) GBaseObject *pShape : object for operation
Return value	Line color
Details	Gets the line color for the drawing/line drawing.
File name	gwin.h

GCSSetPenDash

gwin.h

Outline	Set line type
Syntax	void GCSSetPenDash(GBaseObject **pShape,* unsigned short *usPenDash*)
Argument	(i) GBaseObject *pShape : object for operation (i) unsigned short usPenDash : line type
Return value	None
Details	Sets the line type for the drawing/line drawing. One of the following values is set for usPenDash: 0 : solid line 1-8 : line type When a value outside of the range is set, there is no forced change of the setting. After the line type is set, the drawing/line drawing area is registered as a redraw area.
File name	gwin.h

GCSGetPenDash

gwin.h

Outline	Get line type
Syntax	unsigned short GCSGetPenDash(GBaseObject **pShape*)
Argument	(i) GBaseObject *pShape : object for operation
Return value	0 : solid line 1-8 : line type
Details	Gets the line type for the drawing/line drawing.
File name	gwin.h

GCSSetPenSize

gwin.h

Outline	Set line width
Syntax	void GCSSetPenSize(GBaseObject **pShape,* unsigned short *usPenSize*)
Argument	(i) GBaseObject *pShape : object for operation (i) unsigned short usPenSize : line width
Return value	None
Details	Sets the line width for the drawing/line drawing. After the line width is set, the drawing/line drawing area is registered as a redraw area.
File name	gwin.h

GCSGetPenSize

gwin.h

Outline	Get line width
Syntax	unsigned short GCSGetPenSize(GBaseObject **pShape*)
Argument	(i) GBaseObject *pShape : object for operation
Return value	Line width
Details	Gets the line width for the drawing/line drawing.
File name	gwin.h

GCSSetAngle

gwin.h

Outline	Set start angle and finish angle for sector and arc
Syntax	void GCSSetAngle(GBaseObject *pSArc, short nStartAngle, short nEndAngle)
Argument	(i) GBaseObject *pSArc : object for operation (i) short nStartAngle : start angle (i) short nEndAngle : end angle
Return value	None
Details	Sets the start angle and end angle for sector and arcs.
File name	gwin.h

GCSGetAngle

gwin.h

Outline	Get start angle and finish angle for sector and arc
Syntax	void GCSGetAngle(GBaseObject *pSArc, short *nStartAngle, short *nEndAngle)
Argument	(i) GBaseObject *pSArc : object for operation (o) short *nStartAngle : start angle (o) short *nEndAngle : end angle
Return value	None
Details	Gets the start angle and end angle for sector and arcs.
File name	gwin.h

GCSSetBrush

gwin.h

Outline	Set fill brush
Syntax	void GCSSetBrush(GBaseObject *pSRect, GBrush *pBrush)
Argument	(i) GBaseObject *pSRect : object for operation (i) GBrush *pBrush : brush being set
Return value	None
Details	Sets the brush used for drawing fill. After the brush is set, the drawing area is registered as a redraw area.
File name	gwin.h

GCSGetBrush

gwin.h

Outline	Get fill brush
Syntax	GBrush* GCSGetBrush(GBaseObject *pSRect)
Argument	(i) GBaseObject *pSRect : object for operation
Return value	Pointer to fill brush
Details	Gets the brush used for drawing fill.
File name	gwin.h

GCSSetStartPoint

gwin.h

Outline	Set line drawing starting point
Syntax	void GCSSetStartPoint(GBaseObject *pSLine, GPoint *pPoint)
Argument	(i) GBaseObject *pSLine : object for operation (i) GPoint *pPoint : line drawing starting point
Return value	None
Details	Sets the line drawing starting point.
File name	gwin.h

GCSGetStartPoint

gwin.h

Outline	Get line drawing starting point
Syntax	void GCSGetStartPoint(GBaseObject *pSLine, GPoint *pPoint)
Argument	(i) GBaseObject *pSLine : object for operation (o) GPoint *pPoint : variable storing line drawing starting point
Return value	None
Details	Gets the line drawing starting point.
File name	gwin.h

GCSSetEndPoint

gwin.h

Outline	Set line drawing end point
Syntax	void GCSSetEndPoint(GBaseObject *pSLine, GPoint *pPoint)
Argument	(i) GBaseObject *pSLine : object for operation (i) GPoint *pPoint : line drawing end point
Return value	None
Details	Sets the line drawing end point.
File name	gwin.h

GCSGetEndPoint

gwin.h

Outline	Get line drawing end point
Syntax	void GCSGetEndPoint(GBaseObject *pSLine, .GPoint *pPoint)
Argument	(i) GBaseObject *pSLine : object for operation (o) GPoint *pPoint : variable storing line drawing end point
Return value	None
Details	Gets the line drawing end point.
File name	gwin.h

GCSSetPoly

gwin.h

Outline	Set polygon/continuous line
Syntax	short GCSSetPoly(GBaseObject *pSPoly, GPoly *pPoly)
Argument	(i) GBaseObject *pSPoly : object for operation (i) GPoly *pPoly : polygon structure
Return value	FALSE : setting failed TRUE : setting succeeded
Details	Sets polygon/continuous line data. Vertex data has memory allocated and is held as internal data.
File name	gwin.h

GCSGetPoly

gwin.h

Outline	Get polygon/continuous line
Syntax	GPoly* GCSGetPoly(GBaseObject *pSPoly)
Argument	(i) GBaseObject *pSPoly : object for operation
Return value	Pointer to polygon structure
Details	Gets polygon/continuous line data.
File name	gwin.h

GCSSetDrawAttribute

gwin.h

Outline	Set drawing attributes for figure
Syntax	void GCSSetDrawAttribute(GBaseObject **pShape,* HWDC *pDC*)
Argument	(i) GBaseObject *pShape : object for operation (i) HWDC pDC : drawing handle
Return value	None
Details	Sets the drawing attributes. The drawing attributes set are perimeter color, perimeter line type, perimeter line width and fill brush.
File name	gwin.h

GCSButtonSetAction

gwin.h

Outline	Set button operations
Syntax	void GCSButtonSetAction(GBaseObject **pButton,* unsigned char *ucAction*)
Argument	(i) GBaseObject *pButton : object for operation (i) unsigned char ucAction : button operation
Return value	None
Details	Sets the button operation. One of the following values is set for ucAction: GBTN_ACT_NOACTION : no operation GBTN_ACT_MOMENTARY : momentary operation GBTN_ACT_ALTERNATE : alternative operation When a value outside of the range is set, there is no forced change of the setting.
File name	gwin.h

GCSButtonGetAction

gwin.h

Outline	Get button operations
Syntax	unsigned char GCSButtonGetAction(GBaseObject *pButton)
Argument	(i) GBaseObject *pButton : object for operation
Return value	GBTN_ACT_NOACTION : no operation GBTN_ACT_MOMENTARY : momentary operation GBTN_ACT_ALTERNATE : alternative operatio Besides the above : illegal operation setting
Details	Gets the button operation setting. When a setting outside the range is made, that setting value is returned.
File name	gwin.h

GCSButtonSetDisplay

gwin.h

Outline	Set button display
Syntax	void GCSButtonSetDisplay(GBaseObject **pButton,* unsigned char *ucDisplay*)
Argument	(i) GBaseObject *pButton : object for operation (i) unsigned char ucDisplay : button display
Return value	None
Details	Sets the button display. One of the following values is set for ucDisplay: GBTN_DISP_RECT : rectangle display GBTN_DISP_OVAL : round shape display GBTN_DISP_IMAGE : image display When a value outside of the range is set, there is no forced change of the setting. After the button display is set, the control area is registered as a redraw area.
File name	gwin.h

GCSButtonGetDisplay

gwin.h

Outline	Get button display
Syntax	unsigned char GCSButtonGetDisplay(GBaseObject *pButton)
Argument	(i) GBaseObject *pButton : object for operation
Return value	GBTN_DISP_RECT : rectangle display GBTN_DISP_OVAL : round shape display GBTN_DISP_IMAGE : image display Besides the above : illegal display setting
Details	Gets the button display setting. When a setting outside the range is made, that setting value is returned.
File name	gwin.h

GCSButtonSetBorderID

gwin.h

Outline	Set 3D border resource ID
Syntax	void GCSButtonSetBorderID(GBaseObject *pButton, unsigned short usID)
Argument	(i) GBaseObject *pButton : object for operation (i) unsigned short usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSButtonGetBorderID

gwin.h

Outline	Get 3D border resource ID
Syntax	unsigned short GCSButtonGetBorderID(GBaseObject *pButton)
Argument	(i) GBaseObject *pButton : object for operation
Return value	3D border resource ID
Details	Gets the 3D border resource ID.
File name	gwin.h

GCSButtonSetOnDesign

gwin.h

Outline	Set ON status design
Syntax	void GCSButtonSetOnDesign(GBaseObject *pButton, GDesign* pDesign)
Argument	(i) GBaseObject *pButton : object for operation (i) GDesign* pDesign : design
Return value	None
Details	Sets the design for ON status. After the design for ON status is set, the control area is registered as a redraw area.
File name	gwin.h

GCSButtonGetOnDesign

gwin.h

Outline	Get ON status design
Syntax	GDesign* GCSButtonGetOnDesign(GBaseObject *pButton)
Argument	(i) GBaseObject *pButton : object for operation
Return value	Pointer to design
Details	Gets the design for ON status.
File name	gwin.h

GCSButtonSetOffDesign

gwin.h

Outline	Set OFF status design
Syntax	void GCSButtonSetOffDesign(GBaseObject *pButton, GDesign *pDesign)
Argument	(i) GBaseObject *pButton : object for operation (i) GDesign* pDesign : design
Return value	None
Details	Sets the design for OFF status. After the design for OFF status is set, the control area is registered as a redraw area.
File name	gwin.h

GCSButtonGetOffDesign

gwin.h

Outline	Get OFF status design
Syntax	GDesign* GCSButtonGetOffDesign(GBaseObject *pButton)
Argument	(i) GBaseObject *pButton : object for operation
Return value	Pointer to design
Details	Gets the design for OFF status.
File name	gwin.h

GCSButtonSetFocusDesign

gwin.h

Outline	Set FOCUS status design
Syntax	void GCSButtonSetFocusDesign(GBaseObject *pButton, GDesign *pDesign)
Argument	(i) GBaseObject *pButton : object for operation (i) GDesign* pDesign : design
Return value	None
Details	Sets the design for FOCUS status. After the design for FOCUS status is set, the control area is registered as a redraw area.
File name	gwin.h

GCSButtonGetFocusDesign

gwin.h

Outline	Get FOCUS status design
Syntax	GDesign* GCSButtonGetFocusDesign(GBaseObject *pButton)
Argument	(i) GBaseObject *pButton : object for operation
Return value	Pointer to design
Details	Gets the design for FOCUS status.
File name	gwin.h

GCSButtonSetDisableDesign

gwin.h

Outline	Set Disable status design
Syntax	void GCSButtonSetDisableDesign(GBaseObject *pButton, GDesign *pDesign)
Argument	(i) GBaseObject *pButton : object for operation (i) GDesign* pDesign : design
Return value	None
Details	Sets the design for Disable status. After the design for Disable status is set, the control area is registered as a redraw area.
File name	gwin.h

GCSButtonGetDisableDesign

gwin.h

Outline	Get Disable status design
Syntax	GDesign* GCSButtonGetDisableDesign(GBaseObject *pButton*)
Argument	(i) GBaseObject *pButton : object for operation
Return value	Pointer to design
Details	Gets the design for Disable status.
File name	gwin.h

GCSButtonSetFontID

gwin.h

Outline	Set font resource ID
Syntax	void GCSButtonSetFontID(GBaseObject *pButton,* unsigned short *usID*)
Argument	(i) GBaseObject *pButton : object for operation (i) unsigned short usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSButtonGetFontID

gwin.h

Outline	Get font resource ID
Syntax	unsigned short GCSButtonGetFontID(GBaseObject *pButton)
Argument	(i) GBaseObject *pButton : object for operation
Return value	Font resource ID
Details	Gets the font resource ID.
File name	gwin.h

GCSButtonSetStringID

gwin.h

Outline	Set caption character string resource ID
Syntax	void GCSButtonSetStringID(GBaseObject *pButton, unsigned short usID)
Argument	(i) GBaseObject *pButton : object for operation (i) unsigned short usID : caption character string resource ID
Return value	None
Details	Sets the caption character string resource ID. After the caption character string resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSButtonGetStringID

gwin.h

Outline	Get caption character string resource ID
Syntax	unsigned short GCSButtonGetStringID(GBaseObject *pButton)
Argument	(i) GBaseObject *pButton : object for operation
Return value	Caption character string resource ID
Details	Gets the caption character string resource ID.
File name	gwin.h

GCSButtonSetCaption

gwin.h

Outline	Set caption information
Syntax	void GCSButtonSetCaption(GBaseObject *pButton, GCaption * pCaption)
Argument	(i) GBaseObject *pButton : object for operation (i) GCaption *pCaption : caption information
Return value	None
Details	Sets pCaption in caption information. After the caption information is set, the control area is registered as a redraw area.
File name	gwin.h

GCSButtonGetCaption

gwin.h

Outline	Get caption information
Syntax	GCaption *GCSButtonGetCaption(GBaseObject *pButton)
Argument	(i) GBaseObject *pButton : object for operation
Return value	Pointer to caption information
Details	Gets the caption information.
File name	gwin.h

GCSButtonSetFocusEffect

gwin.h

Outline	Set effect during focus
Syntax	void GCSButtonSetFocusEffect(GBaseObject *pButton, unsigned char ucFocusEffect)
Argument	(i) GBaseObject *pButton : object for operation (i) unsigned char ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect: GFOCUS_EFFECT_CHG_COLOR : design change GFOCUS_EFFECT_NON : no effect After the effect is set, the control area is registered as a redraw area.
File name	gwin.h

GCSButtonGetFocusEffect

gwin.h

Outline	Get effect during focus
Syntax	unsigned char GCSButtonGetFocusEffect(GBaseObject *pButton)
Argument	(i) GBaseObject *pButton : object for operation
Return value	GFOCUS_EFFECT_CHG_COLOR : design change GFOCUS_EFFECT_NON : no effect Besides the above : illegal display setting
Details	Gets the effect used during focus.
File name	gwin.h

GCSButtonSetStatus

gwin.h

Outline	Set object status
Syntax	void GCSButtonSetStatus(GBaseObject *pButton, unsigned char ucStatus)
Argument	(i) GBaseObject *pButton : object for operation (i) unsigned char ucStatus : set object status
Return value	None
Details	Sets the GCButton object status. After the GCButton object status is set, the control area is registered as a redraw area.
File name	gwin.h

GCSButtonGetStatus

gwin.h

Outline	Get object status
Syntax	unsigned char GCSButtonGetStatus(GBaseObject *pButton*)
Argument	(i) GBaseObject *pButton : object for operation
Return value	Object status
Details	Gets the GCButton object status.
File name	gwin.h

GCSCheckboxSetBorderID

gwin.h

Outline	Set 3D border resource ID
Syntax	void GCSCheckboxSetBorderID(GBaseObject *pCheckBox,* unsigned short *usID*)
Argument	(i) GBaseObject *pCheckBox : object for operation (i) unsigned short usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSCheckboxGetBorderID

gwin.h

Outline	Get 3D border resource ID
Syntax	unsigned short GCSCheckboxGetBorderID(GBaseObject *pCheckBox)
Argument	(i) GBaseObject *pCheckBox : object for operation
Return value	3D border resource ID
Details	Gets the 3D border resource ID setting.
File name	gwin.h

GCSCheckboxSetFontID

gwin.h

Outline	Set font resource ID
Syntax	void GCSCheckboxSetFontID(GBaseObject *pCheckBox, unsigned short usID)
Argument	(i) GBaseObject *pCheckBox : object for operation (i) unsigned short usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSCheckboxGetFontID

gwin.h

Outline	Get font resource ID
Syntax	unsigned short GCSCheckboxGetFontID(GBaseObject *pCheckBox)
Argument	(i) GBaseObject *pCheckBox : object for operation
Return value	Font resource ID
Details	Gets the font resource ID.
File name	gwin.h

GCSCheckboxSetStringID

gwin.h

Outline	Set caption character string resource ID
Syntax	void GCSCheckboxSetStringID(GBaseObject *pCheckBox, unsigned short usID)
Argument	(i) GBaseObject *pCheckBox : object for operation (i) unsigned short usID : caption character string resource ID.
Return value	None
Details	Sets the caption character string resource ID. After the caption character string resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSCheckboxGetStringID

gwin.h

Outline	Get caption character string resource ID
Syntax	unsigned short GCSCheckboxGetStringID(GBaseObject *pCheckBox)
Argument	(i) GBaseObject *pCheckBox : object for operation
Return value	Caption character string resource ID.
Details	Gets the caption character string resource ID.
File name	gwin.h

GCSCheckboxSetCaption

gwin.h

Outline	Set caption information
Syntax	void GCSCheckboxSetCaption(GBaseObject *pCheckBox, GCaption *pCaption)
Argument	(i) GBaseObject *pCheckBox : object for operation (i) GCaption *pCaption : caption information
Return value	None
Details	Sets pCaption in caption information All except character color are invalid. After the caption information is set, the control area is registered as a redraw area.
File name	gwin.h

GCSCheckboxGetCaption

gwin.h

Outline	Get caption information
Syntax	GCaption *GCSCheckboxGetCaption(GBaseObject *pCheckBox*)
Argument	(i) GBaseObject *pCheckBox : object for operation
Return value	Pointer to caption information
Details	Gets the caption information setting. All settings except character color are invalid.
File name	gwin.h

GCSCheckboxSetFocusEffect

gwin.h

Outline	Set effect during focus
Syntax	void GCSCheckboxSetFocusEffect(GBaseObject *pCheckBox,* unsigned char *ucFocusEffect*)
Argument	(i) GBaseObject *pCheckBox : object for operation (i) unsigned char ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect: GFOCUS_EFFECT_CHG_COLOR : design change GFOCUS_EFFECT_NON : no effect After the effect is set, the control area is registered as a redraw area.
File name	gwin.h

GCSCheckboxGetFocusEffect

gwin.h

Outline	Get effect during focus
Syntax	unsigned char GCSCheckboxGetFocusEffect(GBaseObject *pCheckBox)
Argument	(i) GBaseObject *pCheckBox : object for operation
Return value	GFOCUS_EFFECT_CHG_COLOR : design change GFOCUS_EFFECT_NON : no effect Besides the above : illegal display setting
Details	Gets the effect used during focus.
File name	gwin.h

GCSCheckboxSetStatus

gwin.h

Outline	Set object status
Syntax	void GCSCheckboxSetStatus(GBaseObject **pCheckBox,* unsigned char *ucStatus*)
Argument	(i) GBaseObject *pCheckBox : object for operation (i) unsigned char ucStatus : set object status
Return value	None
Details	Sets status for GCCheckBox object. One of the following values is set for ucStatus: GCHK_STATE_OFF : OFF status GCHK_STATE_ON : ON status GCHK_STATE_OFFSELECT : OFF selection status GCHK_STATE_ONSELECT : ON selection status When a setting outside of the range is made, there is no forced change of the setting.
File name	gwin.h

GCSCheckboxGetStatus

gwin.h

Outline	Get object status
Syntax	unsigned char GCSCheckboxGetStatus(GBaseObject *pCheckBox)
Argument	(i) GBaseObject *pCheckBox : object for operation
Return value	GCHK_STATE_OFF : OFF status GCHK_STATE_ON : ON status GCHK_STATE_OFFSELECT : OFF selection status GCHK_STATE_ONSELECT : ON selection status Besides the above : undefined status
Details	Gets status for GCCheckBox object.
File name	gwin.h

GCSCheckboxSetBoxSize

gwin.h

Outline	Set box size
Syntax	void GCSCheckboxSetBoxSize(GBaseObject *pCheckBox, unsigned short ucBoxSize)
Argument	(i) GBaseObject *pCheckBox : object for operation (i) unsigned short usID : check box size
Return value	None
Details	Sets the box size. After the box size is set, the control area is registered as a redraw area.
File name	gwin.h

GCSCheckboxGetBoxSize

gwin.h

Outline	Get box size
Syntax	unsigned short GCSCheckboxGetBoxSize(GBaseObject *pCheckBox)
Argument	(i) GBaseObject *pCheckBox : object for operation
Return value	Box size
Details	Gets the box size settings.
File name	gwin.h

GCSCheckboxSetBoxColor

gwin.h

Outline	Set box color
Syntax	void GCSCheckboxSetBoxColor(GBaseObject *pCheckBox, GColor gcColor)
Argument	(i) GBaseObject *pCheckBox : object for operation (i) GColor gcColor : box color
Return value	None
Details	Sets the box color. After the box color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSCheckboxGetBoxColor

gwin.h

Outline	Get box color
Syntax	GColor GCSCheckboxGetBoxColor(GBaseObject *pCheckBox)
Argument	(i) GBaseObject *pCheckBox : object for operation
Return value	Box color
Details	Gets the box color.
File name	gwin.h

GCSCheckboxSetFocusColor

gwin.h

Outline	Set background color during focus
Syntax	void GCSCheckboxSetFocusColor(GBaseObject *pCheckBox, GColor gcColor)
Argument	(i) GBaseObject *pCheckBox : object for operation (i) GColor gcColor : background color during focus
Return value	None
Details	Sets the background color when the effect during focus is GFOCUS_EFFECT_CHG_COLOR. After the background color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSCheckboxGetFocusColor

gwin.h

Outline	Get background color during focus
Syntax	GColor GCSCheckboxGetFocusColor(GBaseObject *pCheckBox*)
Argument	(i) GBaseObject *pCheckBox : object for operation
Return value	Background color during focus
Details	Gets the background color when the effect during focus is GFOCUS_EFFECT_CHG_COLOR.
File name	gwin.h

GCSCheckboxSetDisableCaptionColor

gwin.h

Outline	Set character color when disabled
Syntax	void GCSCheckboxSetDisableCaptionColor(GBaseObject *pCheckBox,* GColor *gcColor*)
Argument	(i) GBaseObject *pCheckBox : object for operation (i) GColor gcColor : character color when disabled
Return value	None
Details	Sets the character color during disabled control. After the character color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSCheckboxGetDisableCaptionColor

gwin.h

Outline	Get character color when disabled
Syntax	GColor GCSCheckboxGetDisableCaptionColor(GBaseObject *pCheckBox)
Argument	(i) GBaseObject *pCheckBox : object for operation
Return value	Character color when disabled
Details	Gets the character color during disabled control.
File name	gwin.h

GCSCheckboxSetDisableBoxColor

gwin.h

Outline	Set box color when disabled
Syntax	void GCSCheckboxSetDisableBoxColor(GBaseObject *pCheckBox, GColor gcColor)
Argument	(i) GBaseObject *pCheckBox : object for operation (i) GColor gcColor : box color
Return value	None
Details	Sets the box color during disabled control. After the box color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSCheckboxGetDisableBoxColor

gwin.h

Outline	Get box color when disabled
Syntax	GColor GCSCheckboxGetDisableBoxColor(GBaseObject *pCheckBox*)
Argument	(i) GBaseObject *pCheckBox : object for operation
Return value	Box color when disabled
Details	Gets the box color during disabled control.
File name	gwin.h

GCSEditSetBorderID

gwin.h

Outline	Set 3D border resource ID
Syntax	void GCSEditSetBorderID(GBaseObject *pEdit,* unsigned short *usID*)
Argument	(i) GBaseObject *pEdit : object for operation (i) unsigned short usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSEditGetBorderID

gwin.h

Outline	Get 3D border resource ID
Syntax	unsigned short GCSEditGetBorderID(GBaseObject *pEdit)
Argument	(i) GBaseObject *pEdit : object for operation
Return value	3D border resource ID
Details	Gets the 3D border resource ID setting.
File name	gwin.h

GCSEditSetFontID

gwin.h

Outline	Set font resource ID
Syntax	void GCSEditSetFontID(GBaseObject *pEdit, unsigned short usID)
Argument	(i) GBaseObject *pEdit : object for operation (i) unsigned short usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSEditGetFontID

gwin.h

Outline	Get font resource ID
Syntax	unsigned short GCSEditGetFontID(GBaseObject *pEdit)
Argument	(i) GBaseObject *pEdit : object for operation
Return value	Font resource ID
Details	Gets the font resource ID.
File name	gwin.h

GCSEditAddString

gwin.h

Outline	Add character string
Syntax	short GCSEditAddString(GBaseObject *pEdit, GTCHAR *pszString)
Argument	(i) GBaseObject *pEdit : object for operation (i) char *pszString : character string to be added
Return value	FALSE : processing failed TRUE : processing succeeded
Details	Adds the character string to the end.
File name	gwin.h

GCSEditSetString

gwin.h

Outline	Set display character string
Syntax	short GCSEditSetString(GBaseObject **pEdit,* const GTCHAR ** pString*)
Argument	(i) GBaseObject *pEdit : object for operation (i) const char *pString : display character string
Return value	FALSE : setting failed TRUE : setting succeeded
Details	Sets the display character string.
File name	gwin.h

GCSEditGetString

gwin.h

Outline	Get display character string
Syntax	short GCSEditGetString(GBaseObject **pEdit,* GTCHAR **pString,* long *lMaxCount*)
Argument	(i) GBaseObject *pEdit : object for operation (o) char *pString : buffer where display character string is stored (i) long lMaxCount : buffer size
Return value	FALSE : acquisition failed TRUE : acquisition succeeded
Details	Stores the display character string in pString.
File name	gwin.h

GCSEditSetTextColor

gwin.h

Outline	Set character display color
Syntax	void GCSEditSetTextColor(GBaseObject *pEdit, GColor gcColor)
Argument	(i) GBaseObject *pEdit : object for operation (i) GColor gcColor : color code for character display color
Return value	None
Details	Sets the character display color.
File name	gwin.h

GCSEditGetTextColor

gwin.h

Outline	Get character display color
Syntax	GColor GCSEditGetTextColor(GBaseObject *pEdit)
Argument	(i) GBaseObject *pEdit : object for operation
Return value	Color code for character display color
Details	Gets the character display color.
File name	gwin.h

GCSEditSetBackColor

gwin.h

Outline	Set background color
Syntax	void GCSEditSetBackColor(GBaseObject *pEdit, GColor gcColor)
Argument	(i) GBaseObject *pEdit : object for operation (i) GColor gcColor : color code for background color
Return value	None
Details	Sets the background color.
File name	gwin.h

GCSEditGetBackColor

gwin.h

Outline	Get background color
Syntax	GColor GCSEditGetBackColor(GBaseObject *pEdit)
Argument	(i) GBaseObject *pEdit : object for operation
Return value	Color code for background color
Details	Gets the background color.
File name	gwin.h

GCSEditSetBrush

gwin.h

Outline	Set brush
Syntax	void GCSEditSetBrush(GBaseObject *pEdit, GBrush *pBrush)
Argument	(i) GBaseObject *pEdit : object for operation (i) GBrush *pBrush : brush structure
Return value	TRUE : processing succeeded FALSE : processing failed
Details	Sets the brush.
File name	gwin.h

GCSEditDeleteString

gwin.h

Outline	Delete character string
Syntax	void GCSEditDeleteString(GBaseObject *pEdit)
Argument	(i) GBaseObject *pEdit : object for operation
Return value	None
Details	Deletes all the character strings.
File name	gwin.h

GCSEditAddLine

gwin.h

Outline	Add line
Syntax	short GCSEditAddLine(GBaseObject *pEdit, GTCHAR *pszLine)
Argument	(i) GBaseObject *pEdit : object for operation (i) GTCHAR *pszLine : line to be added
Return value	FALSE : processing failed TRUE : processing succeeded
Details	Adds the character string as a new line at the end.
File name	gwin.h

GCSEditInsertLine

gwin.h

Outline	Insert line
Syntax	short GCSEditInsertLine(GBaseObject *pEdit, GTCHAR *pszLine, unsigned long ulLine)
Argument	(i) GBaseObject *pEdit : object for operation (i) GTCHAR *pszLine : line to be inserted (i) unsigned long ulLine : line No.
Return value	0 : processing failed Other than 0 : number of lines inserted
Details	Inserts the character string as a new line at specified position.
File name	gwin.h

GCSEditDeleteLine

gwin.h

Outline	Delete line
Syntax	short GCSEditDeleteLine(GBaseObject *pEdit, unsigned long ulLine)
Argument	(i) GBaseObject *pEdit : object for operation (i) unsigned long ulLine : line No.
Return value	FALSE : processing failed TRUE : processing succeeded
Details	Deletes the specified line.
File name	gwin.h

GCSEditGetLineString

gwin.h

Outline	Get line character string
Syntax	short GCSEditGetLineString(GBaseObject *pEdit, unsigned long ulLine, GTCHAR *pszString, long lMaxCount)
Argument	(i) GBaseObject *pEdit : object for operation (i) unsigned long ulLine : line No. (i) GTCHAR *pszString : buffer where character string is saved (i) long lMaxCount : buffer size
Return value	FALSE : processing failed TRUE : processing succeeded
Details	Gets the character string for the specified line.
File name	gwin.h

GCSEditSetLineFeedCode

gwin.h

Outline	Set line feed character string
Syntax	void GCSEditSetLineFeedCode(GBaseObject *pEdit, unsigned char ucLineFeedCode)
Argument	(i) GBaseObject *pEdit : object for operation unsigned short usLineFeedType : line feed character type
Return value	None
Details	Sets the line feed character string type.
File name	gwin.h

GCSEditGetLineFeedCode

gwin.h

Outline	Get line feed character string
Syntax	unsigned char GCSEditGetLineFeedCode(GBaseObject *pEdit)
Argument	(i) GBaseObject *pEdit : object for operation
Return value	line feed character string type GEDIT_LF_TYPE_CRLF : \r\n GEDIT_LF_TYPE_LF : \n GEDIT_LF_TYPE_CR : \r
Details	Gets the line feed character string type.
File name	gwin.h

GCSEditSetCursor

gwin.h

Outline	Set cursor position
Syntax	short GCSEditSetCursor(GBaseObject *pEdit, unsigned long ulLine, unsigned long ulIndex)
Argument	(i) GBaseObject *pEdit : object for operation (i) unsigned long ulLine : line No. (i) unsigned long ulindex : specified index
Return value	FALSE : processing failed TRUE : processing succeeded
Details	Sets the cursor position to the specified position.
File name	gwin.h

GCSEditGetCursor

gwin.h

Outline	Get cursor position
Syntax	short GCSEditGetCursor(GBaseObject *pEdit, unsigned long *pulLine, unsigned long *pulIndex)
Argument	(i) GBaseObject *pEdit : object for operation (i) unsigned long *pulLine : line No. (i) unsigned long *pulindex : specified index
Return value	FALSE : processing failed TRUE : processing succeeded
Details	Gets the cursor position.
File name	gwin.h

GCSEditSetForeColor

gwin.h

Outline	Set foreground color
Syntax	void GCSEditSetForeColor(GBaseObject *pEdit, GColor gcColor)
Argument	(i) GBaseObject *pEdit : object for operation (i) GColor gcColor : color code for foreground color
Return value	None
Details	Sets the foreground color. After the foreground color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSEditGetForeColor

gwin.h

Outline	Get foreground color
Syntax	GColor GCSEditGetForeColor(GBaseObject *pEdit)
Argument	(i) GBaseObject *pEdit : object for operation
Return value	Color code for foreground color
Details	Gets the foreground color.
File name	gwin.h

GCSEditSetFillPattern

gwin.h

Outline	Set fill pattern
Syntax	void GCSEditSetFillPattern(GBaseObject *pEdit, short nFillPattern)
Argument	(i) GBaseObject *pEdit : object for operation (i) short nFillPattern : fill pattern
Return value	None
Details	Sets the fill pattern. After the fill pattern is set, the control area is registered as a redraw area.
File name	gwin.h

GCSEditGetFillPattern

gwin.h

Outline	Get fill pattern
Syntax	short GCSEditGetFillPattern(GBaseObject *pEdit)
Argument	(i) GBaseObject *pEdit : object for operation
Return value	Fill pattern
Details	Gets the fill pattern.
File name	gwin.h

GCSEditSetInsertMode

gwin.h

Outline	Set insert/overwrite mode
Syntax	void GCSEditSetInsertMode(GBaseObject **pEdit,* short *nIsInsert*)
Argument	(i) GBaseObject *pEdit : object for operation (i) short nIsInsert : set insert/overwrite mode TRUE : insert mode FALSE : overwrite mode
Return value	None
Details	Sets the insert/overwrite mode.
File name	gwin.h

GCSEditGetInsertMode

gwin.h

Outline	Get insert/overwrite mode
Syntax	short GCSEditGetInsertMode(GBaseObject **pEdit*)
Argument	(i) GBaseObject *pEdit : object for operation
Return value	TRUE : insert mode FALSE : overwrite mode
Details	Gets the insert/overwrite mode.
File name	gwin.h

GCSEditSetLineBrush

gwin.h

Outline	Set line brush
Syntax	short GCSEditSetLineBrush(GBaseObject **pEdit,* GBrush **pBrush,* unsigned long *ulLine*)
Argument	(i) GBaseObject *pEdit : object for operation (i) unsigned long ulLine : line No. (i) GBrush *pBrush : brush structure
Return value	None
Details	Sets the brush for the specified line.
File name	gwin.h

GCSEditGetLineBrush

gwin.h

Outline	Get line brush
Syntax	short GCSEditGetLineBrush(GBaseObject **pEdit,* GBrush **pBrush,* unsigned long *ulLine*)
Argument	(i) GBaseObject *pEdit : object for operation (i) unsigned long ulLine : line No. (i) GBrush *pBrush : brush structure
Return value	TRUE : processing succeeded FALSE : processing failed
Details	Gets the brush for the specified line.
File name	gwin.h

GCSEditSetLineTextColor

gwin.h

Outline	Set line text display color
Syntax	short GCSEditSetLineTextColor(GBaseObject **pEdit,* GColor *gcLineColor,* unsigned long *ulLine*)
Argument	(i) GBaseObject *pEdit : object for operation (i) unsigned long ulLine : line No. (i) GColor gcColor; : color code for text display color
Return value	None
Details	Sets the text display color for the specified line.
File name	gwin.h

GCSEditGetLineTextColor

gwin.h

Outline	Get line text display color
Syntax	GColor GCSEditGetLineTextColor(GBaseObject **pEdit,* unsigned long *ulLine*)
Argument	(i) GBaseObject *pEdit : object for operation
Return value	Color code for text display color for specified line
Details	Gets the text display color for the specified line.
File name	gwin.h

GCSEditGetLength

gwin.h

Outline	Get character length
Syntax	unsigned long GCSEditGetLength(GBaseObject *pEdit)
Argument	(i) GBaseObject *pEdit : object for operation
Return value	Number of characters
Details	Gets the current number of characters.
File name	gwin.h

GCSEditFind

gwin.h

Outline

Search forward

Syntax

```
short GCSEditFind(
  GBaseObject     *pEdit,
  GTCHAR          *pszString,
  unsigned long   ulStartLine,
  unsigned long   ulStartIndex,
  unsigned long   *pulEditFindLine,
  unsigned long   *pulEditFindPos
)
```

Argument

(i) GBaseObject *pEdit	: object for operation
(i) GTCHAR *pszString	: character string for search
(i) unsigned long ulStartLine	: starting line for search
(i) unsigned long ulStartIndex	: starting position for search
(i) unsigned long *pulFindLine	: search result line
(i) unsigned long *pulFindIndex	: search result position

Return value

FALSE	: search failed
TRUE	: search succeeded

Details

Searchs the character string for search from the search starting position toward the end. Returns the position of the character string found first.

File name

gwin.h

GCSEditReverseFind

gwin.h

Outline

Search backward

Syntax

```
short GCSEditReverseFind(
  GBaseObject     *pEdit,
  GTCHAR          *pszString,
  unsigned long   ulStartLine,
  unsigned long   ulStartIndex,
  unsigned long   *pulFindLine,
  unsigned long   *pulFindPos
)
```

Argument

(i) GBaseObject *pEdit	: object for operation
(i) GTCHAR *pszString	: character string for search
(i) unsigned long ulStartLine	: starting line for search
(i) unsigned long ulStartIndex	: starting position for search
(i) unsigned long *pulFindLine	: search results line
(i) unsigned long *pulFindIndex	: search result position

Return value

FALSE	: search failed
TRUE	: search succeeded

Details

Searchs the character string for search from the search starting position toward the beginning. Returns the position of the character string found first.

File name

gwin.h

GCSEditReplace

gwin.h

Outline

Replace

Syntax

```
short GCSEditReplace(
  GBaseObject   *pEdit,
  GTCHAR        *pszString,
  unsigned long  ulStartLine,
  unsigned long  ulStartIndex,
  unsigned long  ulSize
)
```

Argument

(i) GBaseObject *pEdit	: object for operation
(i) GTCHAR *pszString	: replacement character string
(i) unsigned long ulStartLine	: starting line for replacement
(i) unsigned long ulStartIndex	: starting position for replacement
(i) unsigned long ulSize	: replacement size

Return value

FALSE	: replacement failed
TRUE	: replacement succeeded

Details

Replaces a character string of the replacement size starting at the replacement starting position.

File name

gwin.h

GCSEditReplaceAll

gwin.h

Outline	Replace all
Syntax	unsigned long GCSEditReplaceAll(GBaseObject *pEdit, GTCHAR *pszReplace, GTCHAR *pszFind)
Argument	(i) GBaseObject *pEdit : object for operation (i) GTCHAR *pszReplace : replacement character string (i) GTCHAR *pszFind : character string for search
Return value	Number of replacements
Details	Searches the character string for search starting from the character string in the buffer and replaces it with the replacement character string.
File name	gwin.h

GCSEditInsertString

gwin.h

Outline	Insert character string
Syntax	short GCSEditInsertString(GBaseObject **pEdit,* GTCHAR **pszString,* unsigned long *ulLine,* unsigned long *ulIndex*)
Argument	(i) GBaseObject *pEdit : object for operation (i) char *pszString : character string to be inserted (i) unsigned long ulLine : line for insertion (i) unsigned long ulIndex : insertion position
Return value	FALSE : processing failed TRUE : processing succeeded
Details	Inserts character string.
File name	gwin.h

GCSEditGetLineCount

gwin.h

Outline	Get number of lines
Syntax	unsigned long GCSEditGetLineCount(GBaseObject **pEdit*)
Argument	(i) GBaseObject *pEdit : object for operation
Return value	Number of lines
Details	Gets the current number of lines.
File name	gwin.h

GCSEditGetBrush

gwin.h

Outline	Get brush
Syntax	short GCSEditGetBrush(GBaseObject *pEdit, GBrush *pBrush)
Argument	(i) GBaseObject *pEdit : object for operation (i) GBrush *pBrush : brush structure
Return value	TRUE : processing succeeded FALSE : processing failed
Details	Gets the brush.
File name	gwin.h

GCSEditRemoveString

gwin.h

Outline	Delete character string
Syntax	short GCSEditRemoveString(GBaseObject *pEdit, unsigned long ulRemoveSize)
Argument	(i) GBaseObject *pEdit : object for operation (i) unsigned long ulRemoveSize : size of character string for deletion
Return value	FALSE : processing failed TRUE : processing succeeded
Details	Deletes character string from the end.
File name	gwin.h

GCSHtmlbrowserSetBorderID

gwin.h

Outline	Set 3D border resource ID
Syntax	void GCSHtmlbrowserSetBorderID(GBaseObject **pHtmlBrowser,* unsigned short *usID*)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation (i) unsigned short usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSHtmlbrowserGetBorderID

gwin.h

Outline	Get 3D border resource ID
Syntax	unsigned short GCSHtmlbrowserGetBorderID(GBaseObject **pHtmlBrowser*)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation
Return value	3D border resource ID
Details	Gets the 3D border resource ID setting.
File name	gwin.h

GCSHtmlbrowserSetFontID

gwin.h

Outline	Set font resource ID
Syntax	void GCSHtmlbrowserSetFontID(GBaseObject **pHtmlBrowser,* unsigned short *usID*)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation (i) unsigned short usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSHtmlbrowserGetFontID

gwin.h

Outline	Get font resource ID
Syntax	unsigned short GCSHtmlbrowserGetFontID(GBaseObject **pHtmlBrowser*)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation
Return value	Font the resource ID
Details	Gets the font resource ID.
File name	gwin.h

GCSHtmlbrowserSetScrollBarWidth

gwin.h

Outline	Set scroll bar width
Syntax	void GCSHtmlbrowserSetScrollBarWidth(GBaseObject **pHtmlBrowser,* unsigned char *ucScroolBarWidth*)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation (i) unsigned char ucScrollBarWidth : scroll bar width
Return value	None
Details	Sets the scroll bar width. After the scroll bar width is set, the control area is registered as a redraw area.
File name	gwin.h

GCSHtmlbrowserGetScrollBarWidth

gwin.h

Outline	Get scroll bar width
Syntax	unsigned char GCSHtmlbrowserGetScrollBarWidth(GBaseObject **pHtmlBrowser*)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation
Return value	Scroll bar width
Details	Gets the scroll bar width.
File name	gwin.h

GCSHtmlbrowserSetTextColor

gwin.h

Outline	Set text display color
Syntax	void GCSHtmlbrowserSetTextColor(GBaseObject **pHtmlBrowser,* GColor *gcColor*)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation (i) GColor gcColor : color code for text display color
Return value	None
Details	Sets the text display color. After the text display color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSHtmlbrowserGetTextColor

gwin.h

Outline	Get text display color
Syntax	GColor GCSHtmlbrowserGetTextColor(GBaseObject **pHtmlBrowser*)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation
Return value	Color code for text display color
Details	Gets the text display color.
File name	gwin.h

GCSHtmlbrowserSetBackColor

gwin.h

Outline	Set background color
Syntax	void GCSHtmlbrowserSetBackColor(GBaseObject **pHtmlBrowser,* GColor *gcColor*)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation (i) GColor gcColor : color code for background color
Return value	None
Details	Sets the background color. After the background color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSHtmlbrowserGetBackColor

gwin.h

Outline	Get background color
Syntax	GColor GCSHtmlbrowserGetBackColor(GBaseObject **pHtmlBrowser*)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation
Return value	Color code for background color
Details	Gets the background color.
File name	gwin.h

GCSHtmlbrowserSetLinkColor

gwin.h

Outline	Set link color
Syntax	void GCSHtmlbrowserSetLinkColor(GBaseObject *pHtmlBrowser, GColor gcColor)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation (i) GColor gcColor : color code for link color
Return value	None
Details	Sets the link color. After the link color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSHtmlbrowserGetLinkColor

gwin.h

Outline	Get link color
Syntax	GColor GCSHtmlbrowserGetLinkColor(GBaseObject *pHtmlBrowser)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation
Return value	Color code for link color
Details	Gets the link color.
File name	gwin.h

GCSHtmlbrowserSetHtmlFileName

gwin.h

Outline	Set HTML file name
Syntax	void GCSHtmlbrowserSetHtmlFileName(GBaseObject **pHtmlBrowser,* GTCHAR **FileName*)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation (i) GTCHAR *FileName : HTML file name
Return value	None
Details	Sets the HTML file name. After the HTML file name is set, the read complete flag is cleared to 0 for reading again.
File name	gwin.h

GCSHtmlbrowserGetHtmlFileName

gwin.h

Outline	Get HTML file name
Syntax	GTCHAR* GCSHtmlbrowserGetHtmlFileName(GBaseObject **pHtmlBrowser*)
Argument	(i) GBaseObject *pHtmlBrowser : object for operation
Return value	HTML file name
Details	Gets the HTML file name.
File name	gwin.h

GCSLabelSetFontID

gwin.h

Outline	Set font resource ID
Syntax	void GCSLabelSetFontID(GBaseObject **pLabel,* unsigned short *usID*)
Argument	(i) GBaseObject *pLabel : object for operation (i) unsigned short usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSLabelGetFontID

gwin.h

Outline	Get font resource ID
Syntax	unsigned short GCSLabelGetFontID(GBaseObject **pLabel*)
Argument	(i) GBaseObject *pLabel : object for operation
Return value	Font resource ID
Details	Gets the font resource ID.
File name	gwin.h

GCSLabelSetStringID

gwin.h

Outline	Set caption character string resource ID
Syntax	void GCSLabelSetStringID(GBaseObject **pLabel,* unsigned short *usID*)
Argument	(i) GBaseObject *pLabel : object for operation (i) unsigned short usID : caption character string resource ID
Return value	None
Details	Sets the caption character string resource ID. After the caption character string resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSLabelGetStringID

gwin.h

Outline	Get caption character string resource ID
Syntax	unsigned short GCSLabelGetStringID(GBaseObject **pLabel*)
Argument	(i) GBaseObject *pLabel : object for operation
Return value	Caption character string resource ID
Details	Gets the caption character string resource ID.
File name	gwin.h

GCSLabelSetCaption

gwin.h

Outline	Set caption information
Syntax	void GCSLabelSetCaption(GBaseObject *pLabel, GCaption *pCaption)
Argument	(i) GBaseObject *pLabel : object for operation (i) GCaption *pCaption : caption information
Return value	None
Details	Sets pCaption in caption information. After the caption information is set, the control area is registered as a redraw area.
File name	gwin.h

GCSLabelGetCaption

gwin.h

Outline	Get caption information
Syntax	GCaption *GCSLabelGetCaption(GBaseObject *pLabel)
Argument	(i) GBaseObject *pLabel : object for operation
Return value	Pointer to caption information
Details	Gets the caption information setting.
File name	gwin.h

GCSListSetBorderID

gwin.h

Outline	Set 3D border resource ID
Syntax	void GCSListSetBorderID(GBaseObject **pList,* unsigned short *usID*)
Argument	(i) GBaseObject *pList : object for operation (i) unsigned short usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSListGetBorderID

gwin.h

Outline	Get 3D border resource ID
Syntax	unsigned short GCSListGetBorderID(GBaseObject **pList*)
Argument	(i) GBaseObject *pList : object for operation
Return value	3D border resource ID
Details	Gets the 3D border resource ID setting.
File name	gwin.h

GCSListSetFontID

gwin.h

Outline	Set font resource ID
Syntax	void GCSListSetFontID(GBaseObject **pList,* unsigned short *usID*)
Argument	(i) GBaseObject *pList : object for operation (i) unsigned short usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSListGetFontID

gwin.h

Outline	Get font resource ID
Syntax	unsigned short GCSListGetFontID(GBaseObject **pList*)
Argument	(i) GBaseObject *pList : object for operation
Return value	Font resource ID
Details	Gets the font resource ID.
File name	gwin.h

GCSListSetCaption

gwin.h

Outline	Set caption information
Syntax	void GCSListSetCaption(GBaseObject *pList, GCaption *pCaption)
Argument	(i) GBaseObject *pList : object for operation (i) GCaption *pCaption : caption information
Return value	None
Details	Sets pCaption in caption information. After the caption information is set, the control area is registered as a redraw area.
File name	gwin.h

GCSListGetCaption

gwin.h

Outline	Get caption information
Syntax	GCaption *GCSListGetCaption(GBaseObject *pList)
Argument	(i) GBaseObject *pList : object for operation
Return value	Pointer to caption information
Details	Gets the caption information setting.
File name	gwin.h

GCSListSetFocusColor

gwin.h

Outline	Set background color during focus
Syntax	void GCSListSetFocusColor(GBaseObject **pList,* GColor *gcColor*)
Argument	(i) GBaseObject *pList : object for operation (i) GColor gcColor : color code for background color during focus
Return value	None
Details	Sets the background color during focus. After the background color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSListGetFocusColor

gwin.h

Outline	Get background color during focus
Syntax	GColor GCSListGetFocusColor(GBaseObject **pList*)
Argument	(i) GBaseObject *pList : object for operation
Return value	Color code for background color during focus
Details	Gets the background color during focus.
File name	gwin.h

GCSListSetScrollButtonColor

gwin.h

Outline	Set scroll button color
Syntax	void GCSListSetScrollButtonColor(GBaseObject **pList,* GColor *gcColor*)
Argument	(i) GBaseObject *pList : object for operation (i) GColor gcColor : color code for scroll button color
Return value	None
Details	Sets the scroll button color. After the scroll button color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSListGetScrollButtonColor

gwin.h

Outline	Get scroll button color
Syntax	GColor GCSListGetScrollButtonColor(GBaseObject **pList*)
Argument	(i) GBaseObject *pList : object for operation
Return value	Color code
Details	Gets the scroll button color.
File name	gwin.h

GCSListSetScrollBarColor

gwin.h

Outline	Set scroll bar color
Syntax	void GCSListSetScrollBarColor(GBaseObject *pList, GColor gcColor)
Argument	(i) GBaseObject *pList : object for operation (i) GColor gcColor : color code for scroll bar color
Return value	None
Details	Sets the scroll bar color. After the scroll bar color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSListGetScrollBarColor

gwin.h

Outline	Get scroll bar color
Syntax	GColor GCSListGetScrollBarColor(GBaseObject *pList)
Argument	(i) GBaseObject *pList : object for operation
Return value	Color code
Details	Gets the scroll bar color.
File name	gwin.h

GCSListSetNormalColor

gwin.h

Outline	Set normal background color
Syntax	void GCSListSetNormalColor(GBaseObject *pList, GColor gcColor)
Argument	(i) GBaseObject *pList : object for operation (i) GColor gcColor : color code for normal background color
Return value	None
Details	Sets the normal background color. After the normal background color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSListGetNormalColor

gwin.h

Outline	Get normal background color
Syntax	GColor GCSListGetNormalColor(GBaseObject *pList)
Argument	(i) GBaseObject *pList : object for operation
Return value	Color code for normal background color
Details	Gets the normal background color.
File name	gwin.h

GCSListSetDisableColor

gwin.h

Outline	Set background color when disabled
Syntax	void GCSListSetDisableColor(GBaseObject **pList,* GColor *gcColor*)
Argument	(i) GBaseObject *pList : object for operation (i) GColor gcColor : color code for background color when disabled
Return value	None
Details	Sets the background color when disabled. After the background color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSListGetDisableColor

gwin.h

Outline	Get background color when disabled
Syntax	GColor GCSListGetDisableColor(GBaseObject **pList*)
Argument	(i) GBaseObject *pList : object for operation
Return value	Color code for background color setting when disabled
Details	Gets the background color when disabled.
File name	gwin.h

GCSListSetScrollBarWidth

gwin.h

Outline	Set scroll bar width
Syntax	void GCSListSetScrollBarWidth(GBaseObject *pList, unsigned char ucScroolBarWidth)
Argument	(i) GBaseObject *pList : object for operation (i) unsigned char ucScrollBarWidth : scroll bar width
Return value	None
Details	Sets the scroll bar width. After the scroll bar width is set, the control area is registered as a redraw area.
File name	gwin.h

GCSListGetScrollBarWidth

gwin.h

Outline	Get scroll bar width
Syntax	unsigned char GCSListGetScrollBarWidth(GBaseObject *pList)
Argument	(i) GBaseObject *pList : object for operation
Return value	Scroll bar width
Details	Gets the scroll bar width.
File name	gwin.h

GCSListSetSelectBarColor

gwin.h

Outline	Set selection bar color
Syntax	void GCSListSetSelectBarColor(GBaseObject **pList,* GColor *gcColor*)
Argument	(i) GBaseObject *pList : object for operation (i) GColor gcColor : color code for selection bar color
Return value	None
Details	Sets the selection bar color. After the selection bar color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSListGetSelectBarColor

gwin.h

Outline	Get selection bar color
Syntax	GColor GCSListGetSelectBarColor(GBaseObject **pList*)
Argument	(i) GBaseObject *pList : object for operation
Return value	Color code for selection bar
Details	Gets the selection bar color.
File name	gwin.h

GCSListSetMaxListLines

gwin.h

Outline	Set maximum number of lines in list
Syntax	void GCSListSetMaxListLines(GBaseObject **pList,* unsigned short *usMaxListLines*)
Argument	(i) GBaseObject *pList : object for operation (i) unsigned short usMaxListLines : maximum number 0 to 32767
Return value	None
Details	Sets the maximum number of lines for list control. The range of settings is 0 to 32767. When the maximum number of lines for display that is set is less than the number of lines in a particular list already registered, the character strings in the list that are over the setting are deleted.
File name	gwin.h

GCSListGetMaxListLines

gwin.h

Outline	Get maximum number of lines in list
Syntax	unsigned short GCSListGetMaxListLines(GBaseObject ** pList*)
Argument	(i) GBaseObject *pList : object for operation
Return value	Maximum number of lines in list
Details	Gets the maximum number of lines for list control.
File name	gwin.h

GCSListAddString

gwin.h

Outline	Add list character string
Syntax	short GCSListAddString(GBaseObject **pList,* GTCHAR **pszString*)
Argument	(i) GBaseObject *pList : object for operation (i) char *pszString : character string to be added
Return value	-1 : addition failed 0 or more : line added
Details	Adds a character string to the end of a list.
File name	gwin.h

GCSListGetListString

gwin.h

Outline	Get list character string
Syntax	short GCSListGetListString(GBaseObject **pList,* short *nIndex,* GTCHAR **pszString,* short *nLength*)
Argument	(i) GBaseObject *pList : object for operation (i) short nIndex : line No. to be acquired (o) char* pszString : buffer storing the acquired character string (i) short nLength : buffer size
Return value	FALSE : acquisition failed TRUE : acquisition succeeded
Details	Gets the character string for the specified line in a list.
File name	gwin.h

GCSListRemoveAllStrings

gwin.h

Outline	Delete all list character strings
Syntax	void GCSListRemoveAllStrings(GBaseObject **pList*)
Argument	(i) GBaseObject *pList : object for operation
Return value	None
Details	Deletes all the character strings in a list.
File name	gwin.h

GCSListSetCurrentSelect

gwin.h

Outline	Set selected line
Syntax	short GCSListSetCurrentSelect(GBaseObject **pList,* long *nSel*)
Argument	(i) GBaseObject *pList : object for operation (i) long nSel : set selected line -1 : no selected line 0 or more : specify selected line
Return value	FALSE : setting failed (illegal setting) TRUE : setting succeeded
Details	Sets the selected line in a list. When -1 is specified, a selection is not made. When the maximum number of lines is set, FALSE is returned for the return value.
File name	gwin.h

GCSListGetCurrentSelect

gwin.h

Outline	Get selected line
Syntax	long GCSListGetCurrentSelect(GBaseObject **pList*)
Argument	(i) GBaseObject *pList : object for operation
Return value	0 or more : current selected line -1 : no selection
Details	Gets the selected line in a list.
File name	gwin.h

GCSListSetVisibleScrollBarStatus

gwin.h

Outline	Set scroll bar display/non-display status
Syntax	void GCSListSetVisibleScrollBarStatus(GBaseObject **pList,* short *fStatus*)
Argument	(i) GBaseObject *pList : object for operation (i) short fStatus : display/non-display status FALSE : non-display TRUE : display
Return value	None
Details	Sets the display/non-display status for the scroll bar. When switching the display/non-display, the object area is registered as a redraw area.
File name	gwin.h

GCSListGetVisibleScrollBarStatus

gwin.h

Outline	Get scroll bar display/non-display status
Syntax	short GCSListGetVisibleScrollBarStatus(GBaseObject **pList*)
Argument	(i) GBaseObject *pList : object for operation
Return value	FALSE : non-display TRUE : display
Details	Gets the display/non-display status for the scroll bar.
File name	gwin.h

GCSListSetEnableScrollBarStatus

gwin.h

Outline	Set scroll bar controllable or not controllable status
Syntax	void GCSListSetEnableScrollBarStatus(GBaseObject **pList,* short *fStatus*)
Argument	(i) GBaseObject *pList : object for operation (i) short fStatus : controllable or not controllable status FALSE : not controllable TRUE : controllable
Return value	None
Details	Sets the controllable or not controllable status for the scroll bar. Controllability with mouse input or key input is determined according to this flag.
File name	gwin.h

GCSListGetEnableScrollBarStatus

gwin.h

Outline	Get scroll bar controllable or not controllable status
Syntax	short GCSListGetEnableScrollBarStatus(GBaseObject **pList*)
Argument	(i) GBaseObject *pList : object for operation
Return value	FALSE : not controllable TRUE : controllable
Details	Gets the controllable or not controllable status for the scroll bar.
File name	gwin.h

GCSListSetTopLine

gwin.h

Outline	Set list top line
Syntax	void GCSListSetTopLine(GBaseObject **pList,* long *lIndex*)
Argument	(i) GBaseObject *pList : object for operation (i) long lIndex : top line No.
Return value	None
Details	Changes the top line of the list to 1Position. When a value outside of the range is set, it is changed to the smallest line or the largest line.
File name	gwin.h

GCSListGetTopLine

gwin.h

Outline	Get list top line
Syntax	long GCSListGetTopLine(GBaseObject *pList)
Argument	(i) GBaseObject *pList : object for operation
Return value	Top line No.
Details	Gets the current first line in a list.
File name	gwin.h

GCSListInsertString

gwin.h

Outline	Insert list character string
Syntax	short GCSListInsertString(GBaseObject *pList, short nIndex, GTCHAR *pszString)
Argument	(i) GBaseObject *pList : object for operation (i) short nIndex : line No. for insertion (-1: add at the end) (i) char *pszString : character string to be added
Return value	-1 : addition failed 0 or more : line added
Details	Adds a character string to the specified line in a list. Specifies a line No. to be added to nIndex. When -1 is specified in nIndex, it is added to the end of the list. When the line specified by nIndex does not exist, it is an addition failure.
File name	gwin.h

GCSListGetLineCount

gwin.h

Outline	Get number of lines in list
Syntax	short GCSListGetLineCount(GBaseObject *pList)
Argument	(i) GBaseObject *pList : object for operation
Return value	Current number of lines in list
Details	Gets the current number of lines in a list.
File name	gwin.h

GCSListRemoveString

gwin.h

Outline	Delete list character string
Syntax	short GCSListRemoveString(GBaseObject *pList, short nIndex)
Argument	(i) GBaseObject *pList : object for operation (i) short nIndex : line No. for deletion
Return value	FALSE : deletion failed TRUE : deletion succeeded
Details	Deletes the character string for the specified line in a list.
File name	gwin.h

GCSProgressbarSetBackGroundDesign

gwin.h

Outline	Set background design
Syntax	void GCSProgressbarSetBackGroundDesign(GBaseObject *pProgressBar, GDesign *pDesign)
Argument	(i) GBaseObject *pProgressBar : object for operation (i) GDesign* pDesign : design
Return value	None
Details	Sets the background design for the progress bar. After the background design is set, the control area is registered as a redraw area.
File name	gwin.h

GCSProgressbarGetBackGroundDesign

gwin.h

Outline	Get background design
Syntax	GDesign* GCSProgressbarGetBackGroundDesign(GBaseObject *pProgressBar)
Argument	(i) GBaseObject *pProgressBar : object for operation
Return value	Pointer to design
Details	Gets the background design for the progress bar.
File name	gwin.h

GCSProgressbarSetBarDesign

gwin.h

Outline	Set bar design
Syntax	void GCSProgressbarSetBarDesign(GBaseObject *pProgressBar, GDesign *pDesign)
Argument	(i) GBaseObject *pProgressBar : object for operation (i) GDesign* pDesign : design
Return value	None
Details	Sets the design for the progress bar. After the bar design is set, the control area is registered as a redraw area.
File name	gwin.h

GCSProgressbarGetBarDesign

gwin.h

Outline	Get bar design
Syntax	GDesign* GCSProgressbarGetBarDesign(GBaseObject *pProgressBar)
Argument	(i) GBaseObject *pProgressBar : object for operation
Return value	Pointer to design
Details	Gets the design for the progress bar.
File name	gwin.h

GCSProgressbarSetRange

gwin.h

Outline	Set progress bar range
Syntax	void GCSProgressbarSetRange(GBaseObject *pProgressBar, short nMin, short nMax)
Argument	(i) GBaseObject *pProgressBar : object for operation (i) short lMin : minimum value (i) short lMax : maximum value
Return value	None
Details	Sets the range for the progress bar. When the smallest value and largest value settings are inverted, they are reversed.
File name	gwin.h

GCSProgressbarGetRange

gwin.h

Outline	Get progress bar range
Syntax	void GCSProgressbarGetRange(GBaseObject *pProgressBar, short *pMin, short *pMax)
Argument	(i) GBaseObject *pProgressBar : object for operation (o) short *lMin : variable storing smallest value (o) short *lMax : variable storing largest value
Return value	None
Details	Gets the range for the progress bar.
File name	gwin.h

GCSProgressbarSetDirection

gwin.h

Outline	Set fill direction
Syntax	void GCSProgressbarSetDirection(GBaseObject **pProgressBar,* unsigned char *ucDirection*)
Argument	(i) GBaseObject *pProgressBar : object for operation (i) unsigned char ucDirection : fill direction
Return value	None
Details	Sets the fill direction for the progress bar. One of the followings is set for ucDirection: GPROG_DIR_LEFT2RIGHT : left to right GPROG_DIR_RIGHT2LEFT : right to left GPROG_DIR_TOP2BOTTOM : top to bottom GPROG_DIR_BOTTOM2TOP : bottom to top When a value outside of the range is set, there is no forced change of the setting.
File name	gwin.h

GCSProgressbarGetDirection

gwin.h

Outline	Get fill direction
Syntax	unsigned char GCSProgressbarGetDirection(GBaseObject *pProgressBar)
Argument	(i) GBaseObject *pProgressBar : object for operation
Return value	GPROG_DIR_LEFT2RIGHT : left to right GPROG_DIR_RIGHT2LEFT : right to left GPROG_DIR_TOP2BOTTOM : top to bottom GPROG_DIR_BOTTOM2TOP : bottom to top Besides the above : illegal display setting
Details	Gets the fill direction setting.
File name	gwin.h

GCSProgressbarGetValue

gwin.h

Outline	Get current progress bar value
Syntax	short GCSProgressbarGetValue(GBaseObject *pProgressBar)
Argument	(i) GBaseObject *pProgressBar : object for operation
Return value	Current value
Details	Gets the current value for the progress bar.
File name	gwin.h

GCSProgressbarSetValue

gwin.h

Outline	Set current progress bar value
Syntax	void GCSProgressbarSetValue(GBaseObject *pProgressBar, long nValue)
Argument	(i) GBaseObject *pProgressBar : object for operation (i) long lValue : current value
Return value	None
Details	Sets the current value for the progress bar. The permissible range for the setting is the range specified by SetRange, and values outside of the range become the maximum value or the minimum value.
File name	gwin.h

GCSRadiobuttonSetBorderID

gwin.h

Outline	Set 3D border resource ID
Syntax	void GCSRadiobuttonSetBorderID(GBaseObject *pRadioButton, unsigned short usID)
Argument	(i) GBaseObject *pRadioButton : object for operation (i) unsigned short usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSRadiobuttonGetBorderID

gwin.h

Outline	Get 3D border resource ID
Syntax	unsigned short GCSRadiobuttonGetBorderID(GBaseObject **pRadioButton*)
Argument	(i) GBaseObject *pRadioButton : object for operation
Return value	3D border resource ID
Details	Gets the 3D border resource ID setting.
File name	gwin.h

GCSRadiobuttonSetFontID

gwin.h

Outline	Set font resource ID
Syntax	void GCSRadiobuttonSetFontID(GBaseObject **pRadioButton,* unsigned short *usID*)
Argument	(i) GBaseObject *pRadioButton : object for operation (i) unsigned short usID : font resource ID
Return value	None
Details	Sets the font resource ID. After font resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSRadiobuttonGetFontID

gwin.h

Outline	Get font resource ID
Syntax	unsigned short GCSRadiobuttonGetFontID(GBaseObject **pRadioButton*)
Argument	(i) GBaseObject *pRadioButton : object for operation
Return value	font resource ID
Details	Gets the font resource ID.
File name	gwin.h

GCSRadiobuttonSetStringID

gwin.h

Outline	Set caption character string resource ID
Syntax	void GCSRadiobuttonSetStringID(GBaseObject **pRadioButton,* unsigned short *usID*)
Argument	(i) GBaseObject *pRadioButton : object for operation (i) unsigned short usID : caption character string resource ID
Return value	None
Details	Sets the caption character string resource ID. After the caption character string resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSRadiobuttonGetStringID

gwin.h

Outline	Get caption character string resource ID
Syntax	unsigned short GCSRadiobuttonGetStringID(GBaseObject **pRadioButton*)
Argument	(i) GBaseObject *pRadioButton : object for operation
Return value	Caption character string resource ID
Details	Gets the caption character string resource ID.
File name	gwin.h

GCSRadiobuttonSetCaption

gwin.h

Outline	Set caption information
Syntax	void GCSRadiobuttonSetCaption(GBaseObject **pRadioButton,* GCaption **pCaption*)
Argument	(i) GBaseObject *pRadioButton : object for operation (i) GCaption *pCaption : caption information
Return value	None
Details	Sets the pCaption in caption information. All except character color are disabled. After the caption information is set, the control area is registered as a redraw area.
File name	gwin.h

GCSRadiobuttonGetCaption

gwin.h

Outline	Get caption information
Syntax	GCaption *GCSRadiobuttonGetCaption(GBaseObject *pRadioButton*)
Argument	(i) GBaseObject *pRadioButton : object for operation
Return value	Pointer to caption information
Details	Gets the caption information setting. All settings except character color are disabled.
File name	gwin.h

GCSRadiobuttonSetFocusEffect

gwin.h

Outline	Set effect during focus
Syntax	void GCSRadiobuttonSetFocusEffect(GBaseObject *pRadioButton,* unsigned char *ucFocusEffect*)
Argument	(i) GBaseObject *pRadioButton : object for operation (i) unsigned char ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect: GFOCUS_EFFECT_CHG_COLOR : design change GFOCUS_EFFECT_NON : no effect After the effect is set, the control area is registered as a redraw area.
File name	gwin.h

GCSRadiobuttonGetFocusEffect

gwin.h

Outline	Get effect during focus
Syntax	unsigned char GCSRadiobuttonGetFocusEffect(GBaseObject * pRadioButton)
Argument	(i) GBaseObject *pRadioButton : object for operation
Return value	GFOCUS_EFFECT_CHG_COLOR : design change GFOCUS_EFFECT_NON : no effect Besides the above : illegal display setting
Details	Gets the effect used during focus.
File name	gwin.h

GCSRadiobuttonSetStatus

gwin.h

Outline	Set object status
Syntax	void GCSRadiobuttonSetStatus(GBaseObject *pRadioButton, unsigned char ucStatus)
Argument	(i) GBaseObject *pRadioButton : object for operation (i) unsigned char ucStatus : set object status
Return value	None
Details	Sets the GCRadioButton object status. One of the following values is set for ucStatus: GRADIO_STATE_OFF : OFF status GRADIO_STATE_ON : ON status GRADIO_STATE_OFFSELECT : OFF selection status When GRADIO_STATE_ON is set, other controls within the same screen are set to OFF.
File name	gwin.h

GCSRadiobuttonGetStatus

gwin.h

Outline	Get object status
Syntax	unsigned char GCSRadiobuttonGetStatus(GBaseObject **pRadioButton*)
Argument	(i) GBaseObject *pRadioButton : object for operation
Return value	GRADIO_STATE_OFF : OFF status GRADIO_STATE_ON : ON status GRADIO_STATE_OFFSELECT : OFF selection status Besides the above : undefined status
Details	Gets the GCRadioButton object status.
File name	gwin.h

GCSRadiobuttonSetBoxSize

gwin.h

Outline	Set box size
Syntax	void GCSRadiobuttonSetBoxSize(GBaseObject **pRadioButton,* unsigned short *ucBoxSize*)
Argument	(i) GBaseObject *pRadioButton : object for operation (i) unsigned short ucBoxSize : check box size
Return value	None
Details	Sets the box size. After the box size is set, the control area is registered as a redraw area.
File name	gwin.h

GCSRadiobuttonGetBoxSize

gwin.h

Outline	Get box size
Syntax	unsigned short GCSRadiobuttonGetBoxSize(GBaseObject *pRadioButton)
Argument	(i) GBaseObject *pRadioButton : object for operation
Return value	Box size
Details	Gets the box size settings.
File name	gwin.h

GCSRadiobuttonSetBoxColor

gwin.h

Outline	Set box color
Syntax	void GCSRadiobuttonSetBoxColor(GBaseObject *pRadioButton, GColor gcColor)
Argument	(i) GBaseObject *pRadioButton : object for operation (i) GColor gcColor : box color
Return value	None
Details	Sets the box color. After the box color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSRadiobuttonGetBoxColor

gwin.h

Outline	Get box color
Syntax	GColor GCSRadiobuttonGetBoxColor(GBaseObject *pRadioButton)
Argument	(i) GBaseObject *pRadioButton : object for operation
Return value	Box color
Details	Gets the box color.
File name	gwin.h

GCSRadiobuttonSetFocusColor

gwin.h

Outline	Set background color during focus
Syntax	void GCSRadiobuttonSetFocusColor(GBaseObject *pRadioButton, GColor gcColor)
Argument	(i) GBaseObject *pRadioButton : object for operation (i) GColor gcColor : background color during focus
Return value	None
Details	Sets the background color when the effect during focus is GFOCUS_EFFECT_CHG_COLOR. After the background color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSRadiobuttonGetFocusColor

gwin.h

Outline	Get background color during focus
Syntax	GColor GCSRadiobuttonGetFocusColor(GBaseObject *pRadioButton)
Argument	(i) GBaseObject *pRadioButton : object for operation
Return value	Background color during focus
Details	Gets the background color when the effect during focus is GFOCUS_EFFECT_CHG_COLOR.
File name	gwin.h

GCSRadiobuttonSetDisableCaptionColor

gwin.h

Outline	Set character color when disabled
Syntax	void GCSRadiobuttonSetDisableCaptionColor(GBaseObject *pRadioButton, GColor gcColor)
Argument	(i) GBaseObject *pRadioButton : object for operation (i) GColor gcColor : character color when disabled
Return value	None
Details	Sets the character color during disabled control. After the character color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSRadiobuttonGetDisableCaptionColor

gwin.h

Outline	Get character color when diabled
Syntax	GColor GCSRadiobuttonGetDisableCaptionColor(GBaseObject **pRadioButton*)
Argument	(i) GBaseObject *pRadioButton : object for operation
Return value	Character color when disabled
Details	Gets the character color during disabled control.
File name	gwin.h

GCSRadiobuttonSetDisableBoxColor

gwin.h

Outline	Set box color when disabled
Syntax	void GCSRadiobuttonSetDisableBoxColor(GBaseObject **pRadioButton,* GColor *gcColor*)
Argument	(i) GBaseObject *pRadioButton : object for operation (i) GColor gcColor : box color
Return value	None
Details	Sets the box color during disabled control. After the box color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSRadiobuttonGetDisableBoxColor

gwin.h

Outline	Get box color when disabled
Syntax	GColor GCSRadiobuttonGetDisableBoxColor(GBaseObject **pRadioButton*)
Argument	(i) GBaseObject *pRadioButton : object for operation
Return value	Box color when disabled
Details	Gets the box color during disabled control.
File name	gwin.h

GCSRadiobuttonSetRadioGroup

gwin.h

Outline	Set radio group No.
Syntax	void GCSRadiobuttonSetRadioGroup(GBaseObject **pRadioButton,* unsigned short *usRadioGroup*)
Argument	(i) GBaseObject *pRadioButton : object for operation unsigned short usRadioGroup : radio group No.
Return value	None
Details	Sets the group No. for radio buttons. The radio button control in the screen having the same group No. carries out an exclusion operation.
File name	gwin.h

GCSRadiobuttonGetRadioGroup

gwin.h

Outline	Get radio group No.
Syntax	unsigned short GCSRadiobuttonGetRadioGroup(GBaseObject *pRadioButton)
Argument	(i) GBaseObject *pRadioButton : object for operation
Return value	Radio group No.
Details	Gets the group No. for radio buttons.
File name	gwin.h

GCSScrollbarexSetDisplay

gwin.h

Outline	Set display type
Syntax	void GCSScrollbarexSetDisplay(GBaseObject *pScrollBarEx, unsigned char ucDisplay)
Argument	(i) GBaseObject *pScrollBarEx : object for operation (i) unsigned char ucDisplay : display type
Return value	None
Details	Sets the display type. One of the following values is set for ucDisplay: GSBAR_DISP_NORMAL : normal display GSBAR_DISP_IMAGE : image display When a value outside of the range is set, there is no forced change of the setting. After the display type is set, the control area is registered as a redraw area.
File name	gwin.h

GCSScrollbarexGetDisplay

gwin.h

Outline	Get display type
Syntax	unsigned char GCSScrollbarexGetDisplay(GBaseObject **pScrollBarEx*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation
Return value	GSBAR_DISP_NORMAL : normal display GSBAR_DISP_IMAGE : image display Besides the above : illegal display setting
Details	Gets the display type setting. When a setting outside the range is made, that setting value is returned.
File name	gwin.h

GCSScrollbarexSetDirection

gwin.h

Outline	Set scroll bar direction
Syntax	void GCSScrollbarexSetDirection(GBaseObject **pScrollBarEx,* unsigned char *ucDirection*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation (i) unsigned char ucDirection : scroll bar direction
Return value	None
Details	Sets the scroll bar direction. One of the following values is set for ucDisplay: GSBAR_VERTICAL : vertical direction GSBAR_HORIZONTAL : horizontal direction When a value outside of the range is set, there is no forced change of the setting. After the scroll bar direction is set, the control area is registered as a redraw area.
File name	gwin.h

GCSScrollbarexGetDirection

gwin.h

Outline	Get scroll bar direction
Syntax	unsigned char GCSScrollbarexGetDirection(GBaseObject **pScrollBarEx*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation
Return value	GSBAR_VERTICAL : vertical direction GSBAR_HORIZONTAL : horizontal direction Besides the above : illegal display setting
Details	Gets the scroll bar direction setting. When a setting outside the range is made, that setting value is returned.
File name	gwin.h

GCSScrollbarexSetPageSize

gwin.h

Outline	Set page size
Syntax	void GCSScrollbarexSetPageSize(GBaseObject **pScrollBarEx,* unsigned long *ulPageSize*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation (i) unsigned long ulPageSize : page size
Return value	None
Details	Sets the page size. After the page size is set, the control area is registered as a redraw area.
File name	gwin.h

GCSScrollbarexGetPageSize

gwin.h

Outline	Get page size
Syntax	unsigned long GCSScrollbarexGetPageSize(GBaseObject **pScrollBarEx*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation
Return value	Page size
Details	Gets the page size.
File name	gwin.h

GCSScrollbarexSetRange

gwin.h

Outline	Set scroll bar range
Syntax	void GCSScrollbarexSetRange(GBaseObject **pScrollBar* long *nMin* long *nMax*)
Argument	(i) GBaseObject *pScrollBar : object for operation (i) long lMin : minimum value (i) long lMax : maximum value
Return value	None
Details	Sets the range for the scroll bar. When the smallest value and largest value settings are inverted, they are reversed.
File name	gwin.h

GCSScrollbarexGetRange

gwin.h

Outline	Get scroll bar range
Syntax	void GCSScrollbarexGetRange(GBaseObject *pScrollBar long* pMin long* pMax)
Argument	(i) GBaseObject *pScrollBar : object for operation (i) long *pMin : pointer to variable storing smallest value (i) long *pMax : pointer to variable storing largest value
Return value	None
Details	Gets the range for the scroll bar.
File name	gwin.h

GCSScrollbarexSetScrollArrowColor

gwin.h

Outline	Set scroll arrow color
Syntax	void GCSScrollbarexSetScrollArrowColor(GBaseObject *pScrollBarEx, GColor gcColor)
Argument	(i) GBaseObject *pScrollBarEx : object for operation (i) GColor gcColor : color code for scroll arrow color
Return value	None
Details	Sets the scroll bar arrow color. After the scroll bar arrow color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSScrollbarexGetScrollArrowColor

gwin.h

Outline	Get scroll arrow color
Syntax	GColor GCSScrollbarexGetScrollArrowColor(GBaseObject **pScrollBarEx*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation
Return value	Color code
Details	Gets the scroll bar arrow color.
File name	gwin.h

GCSScrollbarexSetScrollPosition

gwin.h

Outline	Set scroll bar current position
Syntax	void GCSScrollbarexSetScrollPosition(GBaseObject **pScrollBar* long *lPosition*)
Argument	(i) GBaseObject *pScrollBar : object for operation (i) long lPosition : current position
Return value	None
Details	Sets the scroll bar current position. When a value outside of the maximum/minimum value is set, that setting value is set to maximum/minimum value.
File name	gwin.h

GCSScrollbarexGetScrollPosition

gwin.h

Outline	Get scroll bar current position
Syntax	long GCSScrollbarexGetScrollPosition(GBaseObject **pScrollBar*)
Argument	(i) GBaseObject *pScrollBar : object for operation
Return value	Current position
Details	Gets the scroll bar current position.
File name	gwin.h

GCSScrollbarexSetPrevBtnOnImgID

gwin.h

Outline	Set design (upper) resource ID when button is ON
Syntax	void GCSScrollbarexSetPrevBtnOnImgID(GBaseObject **pScrollBarEx,* unsigned short *usID*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation (i) unsigned short usID : design (upper) resource ID when button is ON
Return value	None
Details	Sets the design (upper) resource ID when the button is ON. After the design resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSScrollbarexSetPrevBtnOffImgID

gwin.h

Outline	Set design (upper) resource ID when button is OFF
Syntax	void GCSScrollbarexSetPrevBtnOffImgID(GBaseObject *pScrollBarEx, unsigned short usID)
Argument	(i) GBaseObject *pScrollBarEx : object for operation (i) unsigned short usID : design (upper) resource ID when button is OFF
Return value	None
Details	Sets the design (upper) resource ID when the button is OFF. After the design resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSScrollbarexGetPrevBtnOnImgID

gwin.h

Outline	Get design (upper) resource ID when button is ON
Syntax	unsigned short GCSScrollbarexGetPrevBtnOnImgID(GBaseObject *pScrollBarEx)
Argument	(i) GBaseObject *pScrollBarEx : object for operation
Return value	Design (upper) resource ID when button is ON
Details	Gets the design (upper) resource ID setting when the button is ON.
File name	gwin.h

GCSScrollbarexGetPrevBtnOffImgID

gwin.h

Outline	Get design (upper) resource ID when button is OFF
Syntax	unsigned short GCSScrollbarexGetPrevBtnOffImgID(GBaseObject **pScrollBarEx*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation
Return value	Design (upper) resource ID when button is OFF
Details	Gets the design (upper) resource ID setting when the button is OFF.
File name	gwin.h

GCSScrollbarexSetNextBtnOnImgID

gwin.h

Outline	Set design (lower) resource ID when button is ON
Syntax	void GCSScrollbarexSetNextBtnOnImgID(GBaseObject **pScrollBarEx,* unsigned short *usID*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation (i) unsigned short usID : design (lower) resource ID when button is ON
Return value	None
Details	Sets the design (lower) resource ID when the button is ON. After the design resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSScrollbarexSetNextBtnOffImgID

gwin.h

Outline	Set design (lower) resource ID when button is OFF
Syntax	void GCSScrollbarexSetNextBtnOffImgID(GBaseObject **pScrollBarEx,* unsigned short *usID*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation (i) unsigned short usID : design (lower) resource ID when button is OFF
Return value	None
Details	Sets the design (lower) resource ID when the button is OFF. After the design resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSScrollbarexGetNextBtnOnImgID

gwin.h

Outline	Get design (lower) resource ID when button is ON
Syntax	unsigned short GCSScrollbarexGetNextBtnOnImgID(GBaseObject **pScrollBarEx*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation
Return value	Design (lower) resource ID when button is ON
Details	Gets the design (lower) resource ID setting when the button is ON.
File name	gwin.h

GCSScrollbarexGetNextBtnOffImgID

gwin.h

Outline	Get design (lower) resource ID when button is OFF
Syntax	unsigned short GCSScrollbarexGetNextBtnOffImgID(GBaseObject *pScrollBarEx)
Argument	(i) GBaseObject *pScrollBarEx : object for operation
Return value	Design (lower) resource ID when button is OFF
Details	Gets the design (lower) resource ID setting when the button is OFF.
File name	gwin.h

GCSScrollbarexSetBackGroundColor

gwin.h

Outline	Set background color
Syntax	void GCSScrollbarexSetBackGroundColor(GBaseObject *pScrollBarEx, GColor gcColor)
Argument	(i) GBaseObject *pScrollBarEx : object for operation (i) GColor gcColor : color code for background color
Return value	None
Details	Sets the background color. After the background color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSScrollbarexGetBackGroundColor

gwin.h

Outline	Get background color
Syntax	GColor GCSScrollbarexGetBackGroundColor(GBaseObject *pScrollBarEx*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation
Return value	Color code
Details	Gets the background color.
File name	gwin.h

GCSScrollbarexSetOutLineBorderID

gwin.h

Outline	Set outline 3D border
Syntax	void GCSScrollbarexSetOutLineBorderID(GBaseObject *pScrollBarEx,* unsigned short *usID*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation (i) unsigned short usID : 3D border resource ID
Return value	None
Details	Sets the outline 3D border.
File name	gwin.h

GCSScrollbarexGetOutLineBorderID

gwin.h

Outline	Get outline 3D border
Syntax	unsigned short GCSScrollbarexGetOutLineBorderID(GBaseObject **pScrollBarEx*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation
Return value	Outline 3D border resource ID
Details	Gets outline 3D border resource ID.
File name	gwin.h

GCSScrollbarexSetPinchWidth

gwin.h

Outline	Set pinch width
Syntax	void GCSScrollbarexSetPinchWidth(GBaseObject **pScrollBarEx,* unsigned short *nWidth*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation (i) unsigned short nWidth : pinch width
Return value	None
Details	Sets the pinch width.
File name	gwin.h

GCSScrollbarexGetPinchWidth

gwin.h

Outline	Get pinch width
Syntax	unsigned short GCSScrollbarexGetPinchWidth(GBaseObject *pScrollBarEx)
Argument	(i) GBaseObject *pScrollBarEx : object for operation
Return value	Pinch width
Details	Gets the pinch width.
File name	gwin.h

GCSScrollbarexSetPinchColor

gwin.h

Outline	Set pinch color
Syntax	void GCSScrollbarexSetPinchColor(GBaseObject *pScrollBarEx, GColor gcColor)
Argument	(i) GBaseObject *pScrollBarEx : object for operation (i) GColor gcColor : pinch color
Return value	None
Details	Sets the pinch color.
File name	gwin.h

GCSScrollbarexGetPinchColor

gwin.h

Outline	Get pinch color
Syntax	GColor GCSScrollbarexGetPinchColor(GBaseObject *pScrollBarEx)
Argument	(i) GBaseObject *pScrollBarEx : object for operation
Return value	Pinch color
Details	Gets the pinch color.
File name	gwin.h

GCSScrollbarexSetBarEnableStatus

gwin.h

Outline	Set bar display/non-display
Syntax	void GCSScrollbarexSetBarEnableStatus(GBaseObject *pScrollBarEx, short nStatus)
Argument	(i) GBaseObject *pScrollBarEx : object for operation (i) short nStatus : scroll bar display/non-display status TRUE : display FALSE : non-display
Return value	None
Details	Sets the display/non-display for the bar.
File name	gwin.h

GCSScrollbarexGetBarEnableStatus

gwin.h

Outline	Get bar display/non-display
Syntax	short GCSScrollbarexGetBarEnableStatus(GBaseObject **pScrollBarEx*)
Argument	(i) GBaseObject *pScrollBarEx : object for operation
Return value	TRUE : display FALSE : non-display
Details	Gets the display/non-display setting for the bar.
File name	gwin.h

GCSTextboxSetBorderID

gwin.h

Outline	Set 3D border resource ID
Syntax	void GCSTextboxSetBorderID(GBaseObject **pTextBox,* unsigned short *usID*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) unsigned short usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSTextboxGetBorderID

gwin.h

Outline	Get 3D border resource ID
Syntax	unsigned short GCSTextboxGetBorderID(GBaseObject *pTextBox)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	3D border resource ID
Details	Gets the 3D border resource ID setting.
File name	gwin.h

GCSTextboxSetFontID

gwin.h

Outline	Set font resource ID
Syntax	void GCSTextboxSetFontID(GBaseObject *pTextBox, unsigned short usID)
Argument	(i) GBaseObject *pTextBox : object for operation (i) unsigned short usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSTextboxGetFontID

gwin.h

Outline	Get font resource ID
Syntax	unsigned short GCSTextboxGetFontID(GBaseObject *pTextBox*)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	Font resource ID
Details	Gets the font resource ID.
File name	gwin.h

GCSTextboxSetCaption

gwin.h

Outline	Set caption information
Syntax	void GCSTextboxSetCaption(GBaseObject *pTextBox,* GCaption *pCaption*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) GCaption *pCaption : caption information
Return value	None
Details	Sets the pCaption in caption information. After the caption information is set, the control area is registered as a redraw area.
File name	gwin.h

GCSTextboxGetCaption

gwin.h

Outline	Get caption information
Syntax	GCaption *GCSTextboxGetCaption(GBaseObject *pTextBox*)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	Pointer to caption information
Details	Gets the caption information setting.
File name	gwin.h

GCSTextboxSetFocusEffect

gwin.h

Outline	Set effect during focus
Syntax	void GCSTextboxSetFocusEffect(GBaseObject *pTextBox,* unsigned char *ucFocusEffect*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) unsigned char ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect: GFOCUS_EFFECT_CHG_COLOR : design change GFOCUS_EFFECT_NON : no effect After the effect is set, the control area is registered as a redraw area.
File name	gwin.h

GCSTextboxGetFocusEffect

gwin.h

Outline	Get effect during focus
Syntax	unsigned char GCSTextboxGetFocusEffect(GBaseObject *pTextBox)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	GFOCUS_EFFECT_CHG_COLOR : design change GFOCUS_EFFECT_NON : no effect Besides the above : illegal display setting
Details	Gets the effect used during focus.
File name	gwin.h

GCSTextboxSetFocusColor

gwin.h

Outline	Set background color during focus
Syntax	void GCSTextboxSetFocusColor(GBaseObject *pTextBox, GColor gcColor)
Argument	(i) GBaseObject *pTextBox : object for operation (i) GColor gcColor : color code for background color during focus
Return value	None
Details	Sets the background color during focus. After the background color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSTextboxGetFocusColor

gwin.h

Outline	Get background color during focus
Syntax	GColor GCSTextboxGetFocusColor(GBaseObject *pTextBox*)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	Color code for background color during focus
Details	Gets the background color during focus.
File name	gwin.h

GCSTextboxSetNormalColor

gwin.h

Outline	Set normal background color
Syntax	void GCSTextboxSetNormalColor(GBaseObject *pTextBox,* GColor *gcColor*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) GColor gcColor : color code for normal background color
Return value	None
Details	Sets the normal background color. After the background color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSTextboxGetNormalColor

gwin.h

Outline	Get normal background color
Syntax	GColor GCSTextboxGetNormalColor(GBaseObject *pTextBox*)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	Color code for normal background color
Details	Gets the normal background color.
File name	gwin.h

GCSTextboxSetDisableColor

gwin.h

Outline	Set background color when disabled
Syntax	void GCSTextboxSetDisableColor(GBaseObject *pTextBox,* GColor *gcColor*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) GColor gcColor : color code for background color when disabled
Return value	None
Details	Sets the background color when disabled. After the background color is set, the control area is registered as a redraw area.
File name	gwin.h

GCSTextboxGetDisableColor

gwin.h

Outline	Get background color when disabled
Syntax	GColor GCSTextboxGetDisableColor(GBaseObject **pTextBox*)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	Color code for background color setting when disabled
Details	Gets the background color when disabled.
File name	gwin.h

GCSTextboxSetTextType

gwin.h

Outline	Set character string type
Syntax	void GCSTextboxSetTextType(GBaseObject **pTextBox,* unsigned char *ucType*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) unsigned char ucType : character string type
Return value	None
Details	Sets the character string type. One of the following values is set for ucType: GTEXT_TYPE_STRING : character string GTEXT_TYPE_SHORT : short value GTEXT_TYPE_USHORT : unsigned short value GTEXT_TYPE_LONG : long value GTEXT_TYPE_ULONG : unsigned long value GTEXT_TYPE_FLOAT : float value
File name	gwin.h

GCSTextboxGetTextType

gwin.h

Outline	Get character string type
Syntax	unsigned char GCSTextboxGetTextType(GBaseObject *pTextBox)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	GTEXT_TYPE_STRING : character string GTEXT_TYPE_SHORT : short value GTEXT_TYPE_USHORT : unsigned short value GTEXT_TYPE_LONG : long value GTEXT_TYPE_ULONG : unsigned long value GTEXT_TYPE_FLOAT : float value Besides the above : illegal setting
Details	Gets the display character string type.
File name	gwin.h

GCSTextboxSetFormatID

gwin.h

Outline	Set character string display format resource ID
Syntax	void GCSTextboxSetFormatID(GBaseObject *pTextBox, unsigned short usID)
Argument	(i) GBaseObject *pTextBox : object for operation (i) unsigned short usID : character string resource ID as display format
Return value	None
Details	Sets the character string resource ID used as the format when display in a text box. After the character string resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSTextboxGetFormatID

gwin.h

Outline	Get character string display format resource ID
Syntax	unsigned short GCSTextboxGetFormatID(GBaseObject *pTextBox*)
Argument	(i) GBaseObject *pTextBox :object for operation
Return value	Character string resource ID
Details	Gets the character string resource ID used as the display format.
File name	gwin.h

GCSTextboxSetPasswordStatus

gwin.h

Outline	Set password display status
Syntax	void GCSTextboxSetPasswordStatus(GBaseObject *pTextBox,* short *fStatus*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) short fStatus : password display status FALSE : normal display status TRUE : password display status
Return value	None
Details	Sets whether the password will be displayed for text boxes or not. After the password display status is set, the control area is registered as a redraw area.
File name	gwin.h

GCSTextboxGetPasswordStatus

gwin.h

Outline	Get password display status
Syntax	short GCSTextboxGetPasswordStatus(GBaseObject *pTextBox)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	FALSE : normal display status TRUE : password display status
Details	Gets whether the test box display status is normal display status or the password display status.
File name	gwin.h

GCSTextboxSetCommaStatus

gwin.h

Outline	Set comma display status
Syntax	void GCSTextboxSetCommaStatus(GBaseObject *pTextBox, short fStatus)
Argument	(i) GBaseObject *pTextBox : object for operation (i) short fStatus : comma display status FALSE : normal display status TRUE : comma display status
Return value	None
Details	Sets whether commas will be displayed for text boxes or not. After the comma display status is set, the control area is registered as a redraw area.
File name	gwin.h

GCSTextboxGetCommaStatus

gwin.h

Outline	Get comma display status
Syntax	short GCSTextboxGetCommaStatus(GBaseObject *pTextBox*)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	FALSE : normal display status TRUE : comma display status
Details	Gets the setting as to whether the test box display status is normal display status or the comma display status.
File name	gwin.h

GCSTextboxSetRefuseInputNumberStatus

gwin.h

Outline	Set numerical input disabled status
Syntax	void GCSTextboxSetRefuseInputNumberStatus(GBaseObject *pTextBox,* short *fStatus*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) short fStatus : number input status FALSE : input enabled TRUE : input disabled
Return value	None
Details	Sets whether numerical input will be rejected for text boxes or not.
File name	gwin.h

GCSTextboxGetRefuseInputNumberStatus

gwin.h

Outline	Get numerical input disabled status
Syntax	short GCSTextboxGetRefuseInputNumberStatus(GBaseObject *pTextBox)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	FALSE : input enabled TRUE : input disabled
Details	Gets the setting as to whether numerical input will be rejected for text boxes or not.
File name	gwin.h

GCSTextboxSetRefuseInputSmallLetterStatus

gwin.h

Outline	Set one-byte lower case character input disabled status
Syntax	void GCSTextboxSetRefuseInputSmallLetterStatus(GBaseObject *pTextBox, short fStatus)
Argument	(i) GBaseObject *pTextBox : object for operation (i) short fStatus : one-byte lower case character input status FALSE : input enabled TRUE : input disabled
Return value	None
Details	Sets whether one-byte lower case character input will be rejected for text boxes or not.
File name	gwin.h

GCSTextboxGetRefuseInputSmallLetterStatus

gwin.h

Outline	Get one-byte lower case character input disabled status
Syntax	short GCSTextboxGetRefuseInputSmallLetterStatus(GBaseObject *pTextBox)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	FALSE : input enabled TRUE : input disabled
Details	Gets the setting as to whether one-byte lower case character input will be rejected for text boxes or not.
File name	gwin.h

GCSTextboxSetRefuseInputCapitalLetterStatus

gwin.h

Outline	Set one-byte upper case character input disabled status
Syntax	void GCSTextboxSetRefuseInputCapitalLetterStatus(GBaseObject *pTextBox, short fStatus)
Argument	(i) GBaseObject *pTextBox : object for operation (i) short fStatus : one-byte upper case character input status FALSE : input enabled TRUE : input disabled
Return value	None
Details	Sets whether one-byte upper case character input will be rejected for text boxes or not.
File name	gwin.h

GCSTextboxGetRefuseInputCapitalLetterStatus

gwin.h

Outline	Get one-byte upper case character input disabled status
Syntax	short GCSTextboxGetRefuseInputCapitalLetterStatus(GBaseObject *pTextBox*)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	FALSE : input enabled TRUE : input disabled
Details	Gets whether one-byte upper case character input will be rejected for text boxes or not.
File name	gwin.h

GCSTextboxSetRefuseInputSymbolLetterStatus

gwin.h

Outline	Set one-byte symbol input disabled status
Syntax	void GCSTextboxSetRefuseInputSymbolLetterStatus(GBaseObject *pTextBox,* short *fStatus*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) short fStatus : one-byte symbol input status FALSE : input enabled TRUE : input disabled
Return value	None
Details	Sets whether one-byte symbol input will be rejected for text boxes or not.
File name	gwin.h

GCSTextboxGetRefuseInputSymbolLetterStatus

gwin.h

Outline	Get one-byte symbol input disabld status
Syntax	short GCSTextboxGetRefuseInputSymbolLetterStatus(GBaseObject **pTextBox*)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	FALSE : input enabled TRUE : input disabled
Details	Gets whether one-byte symbol input will be rejected for text boxes or not.
File name	gwin.h

GCSTextboxSetRefuseInputFullLetterStatus

gwin.h

Outline	Set two-byte character input disabled status
Syntax	void GCSTextboxSetRefuseInputFullLetterStatus(GBaseObject **pTextBox,* short *fStatus*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) short fStatus : two-byte character input status FALSE : input enabled TRUE : input disabled
Return value	None
Details	Sets whether two-byte character input will be rejected for text boxes or not.
File name	gwin.h

GCSTextboxGetRefuseInputFullLetterStatus

gwin.h

Outline	Get two-byte character input disabled status
Syntax	short GCSTextboxGetRefuseInputFullLetterStatus(GBaseObject *pTextBox)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	FALSE : input enabled TRUE : input disabled
Details	Gets setting as to whether two-byte character input will be rejected for text boxes or not.
File name	gwin.h

GCSTextboxSetCheckMaxStatus

gwin.h

Outline	Set maximum value check status
Syntax	void GCSTextboxSetCheckMaxStatus(GBaseObject *pTextBox, short fStatus)
Argument	(i) GBaseObject *pTextBox : object for operation (i) short fStatus : setting for maximum value check status FALSE : check disabled TRUE : check enabled
Return value	None
Details	Sets whether the maximum value will be checked with SetValue and GetValue for text boxes.
File name	gwin.h

GCSTextboxGetCheckMaxStatus

gwin.h

Outline	Get maximum value check status
Syntax	short GCSTextboxGetCheckMaxStatus(GBaseObject *pTextBox)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	FALSE : check disabled TRUE : check enabled
Details	Gets the setting as to whether the maximum value will be checked with SetValue and GetValue from text boxes.
File name	gwin.h

GCSTextboxSetCheckMinStatus

gwin.h

Outline	Set minimum value check status
Syntax	void GCSTextboxSetCheckMinStatus(GBaseObject *pTextBox, short fStatus)
Argument	(i) GBaseObject *pTextBox : object for operation (i) short fStatus : setting for minimum value check status FALSE : check disabled TRUE : check enabled
Return value	None
Details	Sets whether the maximum value will be checked with SetValue and GetValue for text boxes.
File name	gwin.h

GCSTextboxGetCheckMinStatus

gwin.h

Outline	Get minimum value check status
Syntax	short GCSTextboxGetCheckMinStatus(GBaseObject *pTextBox)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	FALSE : check disabled TRUE : check enabled
Details	Gets the setting as to whether the minimum value will be checked with SetValue and GetValue from text boxes.
File name	gwin.h

GCSTextboxSetMaxGValue

gwin.h

Outline	Set maximum value
Syntax	void GCSTextboxSetMaxGValue(GBaseObject *pTextBox, GValue gvValue)
Argument	(i) GBaseObject *pTextBox : object for operation (i) GValue gvValue : maximum value
Return value	None
Details	Sets the maximum value when checking the maximum value with SetValue and GetValue for text boxes.
File name	gwin.h

GCSTextboxGetMaxGValue

gwin.h

Outline	Get maximum value
Syntax	GValue GCSTextboxGetMaxGValue(GBaseObject **pTextBox*)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	Maximum value (GValue)
Details	Gets the maximum value when checking the maximum value for SetValue and GetValue from text boxes.
File name	gwin.h

GCSTextboxSetMinGValue

gwin.h

Outline	Set minimum value
Syntax	void GCSTextboxSetMinGValue(GBaseObject **pTextBox,* GValue *gvValue*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) GValue gvValue : minimum value
Return value	None
Details	Sets the minimum value when checking the minimum value with SetValue and GetValue for text boxes.
File name	gwin.h

GCSTextboxGetMinGValue

gwin.h

Outline	Get minimum value
Syntax	GValue GCSTextboxGetMinGValue(GBaseObject **pTextBox*)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	Minimum value (GValue)
Details	Gets the minimum value when checking the minimum value with SetValue and GetValue from text boxes.
File name	gwin.h

GCSTextboxSetString

gwin.h

Outline	Set display character string
Syntax	short GCSTextboxSetString(GBaseObject **pTextBox,* const GTCHAR **pString*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) const char *pString : display character string
Return value	FALSE : setting failed TRUE : setting succeeded
Details	Sets the display character string.
File name	gwin.h

GCSTextboxGetString

gwin.h

Outline	Get display character string
Syntax	short GCSTextboxGetString(GBaseObject **pTextBox*, GTCHAR **pString*, long *lMaxCount*)
Argument	(i) GBaseObject *pTextBox : object for operation (o) char *pString : buffer storing display character string (i) long lMaxCount : buffer size
Return value	FALSE : acquisition failed TRUE : acquisition succeeded
Details	Stores the display character string in pString.
File name	gwin.h

GCSTextboxSetGValue

gwin.h

Outline	Set display value
Syntax	short GCSTextboxSetGValue(GBaseObject **pTextBox,* GValue *gvValue*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) GValue gvValue : setting value
Return value	0 : normal value GTEXT_ERROR_UNDER_MIN_LIMIT : less than minimum value GTEXT_ERROR_OVER_MAX_LIMIT : greater than maximum value
Details	Sets the value displayed in a text box. Numbers are converted to character strings according to the display format. At this time, the maximum value and minimum value are checked, and if it is outside of the range, an error is returned in the return value. Error checks are performed by: GTEXT_TYPE_SHORT GTEXT_TYPE_USHORT GTEXT_TYPE_LONG
File name	gwin.h

GCSTextboxGetGValue

gwin.h

Outline	Get display value
Syntax	short GCSTextboxGetGValue(GBaseObject **pTextBox,* GValue **gvValue*)
Argument	(i) GBaseObject *pTextBox : object for operation (i) GValue *gvValue : GValue variable storing the value
Return value	0 : normal value GTEXT_ERROR_UNDER_MIN_LIMIT : less than minimum value GTEXT_ERROR_OVER_MAX_LIMIT : greater than maximum value
Details	Gets the value displayed in a text box. Character strings are converted to numbers according to the display format. At this time, the maximum value and minimum value are checked, and if it is outside of the range, an error is returned in the return value. Error checks are performed by: GTEXT_TYPE_SHORT GTEXT_TYPE_USHORT
File name	gwin.h

GCSTextboxSetStringBuffer

gwin.h

Outline	Set display buffer
Syntax	short GCSTextboxSetStringBuffer(GBaseObject *pTextBox, unsigned long ulBufferSize, GTCHAR *pStringBuffer)
Argument	(i) GBaseObject *pTextBox : object for operation (i) unsigned long ulBufferSize : buffer size (i) char *pStringBuffer : character string buffer
Return value	FALSE : setting failed TRUE : setting succeeded
Details	Sets the display buffer. Allocates memory internally when pStringBuffer is NULL. When a character string buffer is specified, that buffer is set.
File name	gwin.h

GCSTextboxGetStringBuffer

gwin.h

Outline	Get display buffer
Syntax	GTCHAR* GCSTextboxGetStringBuffer(GBaseObject *pTextBox, unsigned long *ulBufferSize)
Argument	(i) GBaseObject *pTextBox : object for operation (i) unsigned long *ulBufferSize : variable storing the buffer size
Return value	Display buffer address
Details	Gets the display buffer.
File name	gwin.h

GCSTextboxSetBackGroundPattern

gwin.h

Outline	Set background fill status
Syntax	void GCSTextboxSetBackGroundPattern(GBaseObject *pTextBox, short nBackGroundPattern)
Argument	(i) GBaseObject *pTextBox : object for operation (i) short nBackGroundPattern : background fill status
Return value	None
Details	Sets the background color fill status. NO_DRAW and FILL_BACK_COLOR are the corresponding fill statuses.
File name	gwin.h

GCSTextboxGetBackGroundPattern

gwin.h

Outline	Get background fill status
Syntax	short GCSTextboxGetBackGroundPattern(GBaseObject *pTextBox)
Argument	(i) GBaseObject *pTextBox : object for operation
Return value	Background fill status NO_DRAW : no fill FILL_BACK_COLOR : background fill
Details	Sets the background color fill status. NO_DRAW and FILL_BACK_COLOR are the corresponding fill statuses.
File name	gwin.h

GCSBWLInit

gwin.h

Outline	Initialization
Syntax	void GCSBWLInit(HGBWL *hList*)
Argument	(i) HGBWL hList : list handle
Return value	None
Details	Initialization
File name	gwin.h

GCSBWLAdd

gwin.h

Outline	Addition
Syntax	short GCSBWLAdd(HGBWL *hList,* GBaseObject **pgbwBaseWindow*)
Argument	(i) HGBWL hList : list handle (i) GbaseObject *pgbwBaseWindow : object to be added
Return value	FALSE : addition failed TRUE : addition succeeded
Details	Addition
File name	gwin.h

GCSBWLRemove

gwin.h

Outline	Deletion
Syntax	short GCSBWLRemove(HGBWL *hList*, GBaseObject **pgbwBaseWindow*)
Argument	(i) HGBWL hList : list handle (i) GBaseObject *pgbwBaseWindow : object to be deleted
Return value	FALSE : deletion failed TRUE : deletion succeeded
Details	Deletion
File name	gwin.h

GCSBWLClear

gwin.h

Outline	All clear
Syntax	void GCSBWLClear(HGBWL *hList*)
Argument	(i) HGBWL hList : list handle
Return value	None
Details	Clears all.
File name	gwin.h

GCSBWLMoveFirst

gwin.h

Outline	Move to first
Syntax	short GCSBWLMoveFirst(HGBWL *hList,* GBaseObject **pgbwBaseWindow*)
Argument	(i) HGBWL hList : list handle (i) GBaseObject *pgbwBaseWindow : object to be moved
Return value	FALSE : specified object does not exist TRUE : move succeeded
Details	Moves to the first.
File name	gwin.h

GCSBWLMoveLast

gwin.h

Outline	Move to last
Syntax	short GCSBWLMoveLast(HGBWL *hList,* GBaseObject **pgbwBaseWindow*)
Argument	(i) HGBWL hList : list handle (i) GbaseObject *pgbwBaseWindow : object to be moved
Return value	FALSE : specified object does not exist TRUE : move succeeded
Details	Moves to the last.
File name	gwin.h

GCSBWLGetFirst

gwin.h

Outline	Get first object
Syntax	GBaseObject* GCSBWLGetFirst(HGBWL *hList*)
Argument	(i) HGBWL hList : list handle
Return value	NULL : object does not exist Other than NULL : first object
Details	Gets the first object.
File name	gwin.h

GCSBWLGetLast

gwin.h

Outline	Get last object
Syntax	GBaseObject* GCSBWLGetLast(HGBWL *hList*)
Argument	(i) HGBWL hList :list handle
Return value	NULL : object does not exist Other than NULL : last object
Details	Gets the last object.
File name	gwin.h

GCSBWLFindListItem

gwin.h

Outline	Find object
Syntax	short GCSBWLFindListItem(HGBWL *hList,* GBaseObject **pBaseWindow*)
Argument	(i) HGBWL hList : list handle (i) GbaseObject *pBaseWindow : object being searched for
Return value	FALSE : none TRUE : present
Details	Finds the object.
File name	gwin.h

GCSBWLFindFirstItem

gwin.h

Outline	Move search position to first item
Syntax	short GCSBWLFindFirstItem(HGBWL *hList*)
Argument	(i) HGBWL hList : list handle
Return value	FALSE : move failed TRUE : move succeeded
Details	Moves the search to the first item.
File name	gwin.h

GCSBWLFindNextItem

gwin.h

Outline	Acquire object at search position and move search position to next
Syntax	short GCSBWLFindNextItem(HGBWL *hList,* GBaseObject ***pgbwBaseWindow*)
Argument	(i) HGBWL hList : list handle (o) GBaseObject **pgbwBaseWindow : store object to be acquired
Return value	FALSE : move failed TRUE : move succeeded
Details	Gets the object at search position and moves search position to the next.
File name	gwin.h

GCSBWLFindLastItem

gwin.h

Outline	Move search position to last item
Syntax	short GCSBWLFindLastItem(HGBWL *hList*)
Argument	(i) HGBWL hList : list handle
Return value	FALSE : move failed TRUE : move succeeded
Details	Moves the search position to the last item.
File name	gwin.h

GCSBWLFindPrevItem

gwin.h

Outline	Acquire object at search position and move search position to previous
Syntax	short GCSBWLFindPrevItem(HGBWL *hList,* GBaseObject ***pgbwBaseWindow*)
Argument	(i) HGBWL hList : list handle (o) GBaseObject **pgbwBaseWindow : store object to be acquired
Return value	FALSE : move failed TRUE : move succeeded
Details	Gets the object at search position and moves search position to the previous.
File name	gwin.h

GCSBWLCount

gwin.h

Outline	Get number of list items
Syntax	long GCSBWLCount(HGBWL *hList*)
Argument	(i) HGBWL hList : list handle
Return value	Number of list items
Details	Gets the number of list items.
File name	gwin.h

GCSBWLSetItemPosition

gwin.h

Outline	Set current list search position
Syntax	void GCSBWLSetItemPosition(HGBWL *hList,* void **pvItemPosition*)
Argument	(i) HGBWL hList : list handle (i) void *pvItemPosition : list search position
Return value	None
Details	Sets the current list search position.
File name	gwin.h

GCSBWLGetItemPosition

gwin.h

Outline	Get current list search position
Syntax	void* GCSBWLGetItemPosition(HGBWL *hList*)
Argument	(i) HGBWL hList : list handle
Return value	Pointer showing current list search position
Details	Gets the current list search position.
File name	gwin.h

GCSBWLNew

gwin.h

Outline	Create list object
Syntax	HGBWL GCSBWLNew()
Argument	None
Return value	List handle
Details	Creates the list object.
File name	gwin.h

GCSBWLDelete

gwin.h

Outline	Delete list object
Syntax	void GCSBWLDelete(HGBWL *hList*)
Argument	(i) HGBWL hList : list handle
Return value	None
Details	Deletes the list object.
File name	gwin.h

GCSWDCGetHGDRAW

gwin.h

Outline	Get drawing handle
Syntax	HGDRAW GCSWDCGetHGDRAW(HWDC *hWDC*)
Argument	(i) HWDC hWDC : device context handle
Return value	None
Details	Gets the drawing handle (HGDRAW).
File name	gwin.h

GCSWDCOffsetOrigin

gwin.h

Outline	Move position of origin
Syntax	void GCSWDCOffsetOrigin(HWDC *hWDC*, short *nX*, short *nY*)
Argument	(i) HWDC hWDC : device context handle (i) short nX : X coordinate offset amount (i) short nY : Y coordinate offset amount
Return value	None
Details	Moves the position of the origin.
File name	gwin.h

GCSWDCSetClipRect

gwin.h

Outline	Define clipping area
Syntax	void GCSWDCSetClipRect(HWDC *hWDC,* GRect **pClip*)
Argument	(i) HWDC hWDC : device context handle (i) GRect *pClip : clipping area
Return value	None
Details	Defines clipping area as pClip.
File name	gwin.h

GCSWDCGetClipRect

gwin.h

Outline	Get clipping area
Syntax	void GCSWDCGetClipRect(HWDC *hWDC,* GRect **pClip*)
Argument	(i) HWDC hWDC : device context handle (i) GRect *pClip : clipping area
Return value	None
Details	Gets the clipping area.
File name	gwin.h

GCSWDCInceptClipRect

gwin.h

Outline	Define clipping for the part overlapping the current clipping area
Syntax	short GCSWDCInceptClipRect(HWDC *hWDC,* GRect **pClip,* GRect **pOldClip*)
Argument	(i) HWDC hWDC : device context handle (i) GRect *pClip : clipping area (o) GRect *pOldClip : previous clipping area
Return value	FALSE : no clipping area TRUE : clipping area present
Details	Defines the clipping area as the part overlapping between the current clipping area and pClip.
File name	gwin.h

GCSWDCBeginDraw

gwin.h

Outline	Start drawing process
Syntax	void GCSWDCBeginDraw(HWDC *hWDC,* GRect **pRefreshRect*)
Argument	(i) HWDC hWDC : device context handle (i) GRect *pRefreshRect : area for redrawing
Return value	None
Details	Starts the drawing process.
File name	gwin.h

GCSWDCEndDraw

gwin.h

Outline	End drawing process
Syntax	void GCSWDCEndDraw(HWDC *hWDC*)
Argument	(i) HWDC hWDC : device context handle
Return value	None
Details	Ends the drawing process.
File name	gwin.h

GCSWDCNew

gwin.h

Outline	Create device context
Syntax	HWDC GCSWDCNew(HGDRAW *hDraw*)
Argument	(i) HGDRAW hDraw : drawing handle
Return value	Device context Null : creation failed Other than Null : device context created
Details	Creates the device context.
File name	gwin.h

GCSWDCDelete

gwin.h

Outline	Delete device context
Syntax	void GCSWDCDelete(HWDC *hWDC*)
Argument	(i) HWDC hWDC : device context handle
Return value	None
Details	Deletes the device context.
File name	gwin.h

GCSWDCSetOrigin

gwin.h

Outline	Set position of origin
Syntax	void GCSWDCSetOrigin(HWDC *hWDC,* short *nX,* short *nY*)
Argument	(i) HWDC hWDC : device context handle (i) short nX : X coordinate (i) short nY : Y coordinate
Return value	None
Details	Sets the position of the origin.
File name	gwin.h

GCSPictureSetDisplay

gwin.h

Outline	Set picture display
Syntax	void GCSPictureSetDisplay(GBaseObject **pPicture* unsigned char *ucDisplay*)
Argument	(i) GBaseObject *pPicture : object for operation (i) unsigned char ucDisplay : picture display
Return value	None
Details	Sets the picture display. One of the following values is set for ucDisplay: GPCTR_DISP_RECT : rectangle display GPCTR_DISP_OVAL : (oval) round shape display GPCTR_DISP_IMAGE : image display When a value outside of the range is set, there is no forced change of the setting.
File name	gwin.h

GCSPictureGetDisplay

gwin.h

Outline	Get picture display
Syntax	unsigned char GCSPictureGetDisplay(GBaseObject **pPicture*)
Argument	(i) GBaseObject *pPicture : object for operation
Return value	GPCTR_DISP_RECT : rectangle display GPCTR_DISP_OVAL : (oval) round shape display GPCTR_DISP_IMAGE : image display Besides the above : illegal display setting
Details	Gets the picture display setting. When a setting outside the range is made, that setting value is returned.
File name	gwin.h

GCSPictureSetBorderID

gwin.h

Outline	Set 3D border resource ID
Syntax	void GCSPictureSetBorderID(GBaseObject **pPicture* unsigned short *usID*)
Argument	(i) GBaseObject *pPicture : object for operation (i) unsigned short usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSPictureGetBorderID

gwin.h

Outline	Get 3D border resource ID
Syntax	unsigned short GCSPictureGetBorderID(GBaseObject **pPicture*)
Argument	(i) GBaseObject *pPicture : object for operation
Return value	3D border resource ID
Details	Gets the 3D border resource ID.
File name	gwin.h

GCSPictureSetFontID

gwin.h

Outline	Set font resource ID
Syntax	void GCSPictureSetFontID(GBaseObject **pPicture* unsigned short *usID*)
Argument	(i) GBaseObject *pPicture : object for operation (i) unsigned short usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSPictureGetFontID

gwin.h

Outline	Get font resource ID
Syntax	unsigned short GCSPictureGetFontID(GBaseObject **pPicture*)
Argument	(i) GBaseObject *pPicture : object for operation
Return value	Font resource ID
Details	Gets the font resource ID.
File name	gwin.h

GCSPictureSetStringID

gwin.h

Outline	Set caption character string resource ID
Syntax	void GCSPictureSetStringID(GBaseObject **pPicture* unsigned short *usID*)
Argument	(i) GBaseObject *pPicture : object for operation (i) unsigned short usID :caption character string resource ID
Return value	None
Details	Sets the caption character string resource ID. After the caption character string resource ID is set, the control area is registered as a redraw area.
File name	gwin.h

GCSPictureGetStringID

gwin.h

Outline	Get caption character string resource ID
Syntax	unsigned short GCSPictureGetStringID(GBaseObject **pPicture*)
Argument	(i) GBaseObject *pPicture : object for operation
Return value	Caption character string resource ID
Details	Gets the caption character string resource ID.
File name	gwin.h

GCSPictureSetCaption

gwin.h

Outline	Set caption information
Syntax	void GCSPictureSetCaption(GBaseObject *pPicture* GCaption *pCaption*)
Argument	(i) GBaseObject *pPicture : object for operation (i) GCaption *pCaption : pointer to caption information
Return value	None
Details	Sets pCaption in caption information. After the caption information is set, the control area is registered as a redraw area.
File name	gwin.h

GCSPictureGetCaption

gwin.h

Outline	Get caption information
Syntax	GCaption *GCSPictureGetCaption(GBaseObject *pPicture*)
Argument	(i) GBaseObject *pPicture : object for operation
Return value	Pointer to caption information
Details	Gets the caption information.
File name	gwin.h

GCSPictureSetStatus

gwin.h

Outline	Set object status
Syntax	void GCSPictureSetStatus(GBaseObject **pPicture* unsigned char *ucStatus*)
Argument	(i) GBaseObject *pPicture : object for operation (i) unsigned char ucStatus : set object status
Return value	None
Details	Sets the GCButton object status. After the GCButton object status is set, the control area is registered as a redraw area.
File name	gwin.h

GCSPictureGetStatus

gwin.h

Outline	Get object status
Syntax	unsigned char GCSPictureGetStatus(GBaseObject **pPicture*)
Argument	(i) GBaseObject *pPicture : object for operation
Return value	Object status
Details	Gets the GCButton object status.
File name	gwin.h

GCSPictureSetMaxStatus

gwin.h

Outline	Set maximus picture status
Syntax	void GCSPictureSetMaxStatus(GBaseObject *pPicture* unsigned char *ucStatus*)
Argument	(i) GBaseObject *pPicture : object for operation (i) unsigned char ucMaxStatus : number of maximus picture status
Return value	None
Details	Sets number of maximus picture status. The picture design shows that the status exist for the number of this setting.
File name	gwin.h

GCSPictureGetMaxStatus

gwin.h

Outline	Get maximus picture status
Syntax	unsigned char GCSPictureGetMaxStatus(GBaseObject *pPicture*)
Argument	(i) GBaseObject *pPicture : object for operation
Return value	Number of maximus picture status
Details	Gets number of maximus picture status.
File name	gwin.h

GCSPictureSetStatusDesign

gwin.h

Outline	Set picture status design
Syntax	void GCSPictureSetStatusDesign(GBaseObject *pPicture GDesign *pDesign)
Argument	(i) GBaseObject *pPicture : object for operation (i) GDesign *pDesign : pointer to status design arrangement
Return value	None
Details	Sets the pointer to picture status design arrangement. After the pointer is set, the control area is registered as a redraw area.
File name	gwin.h

GCSPictureGetStatusDesign

gwin.h

Outline	Get picture status design
Syntax	GDesign* GCSPictureGetStatusDesign(GBaseObject *pPicture)
Argument	(i) GBaseObject *pPicture : object for operation
Return value	NULL : no setting Other than NULL : Pointer to status design arrangement
Details	Gets the pointer to picture status design arrangement.
File name	gwin.h

GCSChangeActiveFocus

gwin.h

Outline	Change active focus
Syntax	void GCSChangeActiveFocus(GBaseObject **pPanel,* GBaseObject **pObject*)
Argument	(i) GBaseObject *pPanel : pointer to management panel object (i) GBaseObject *pObject : object setting the focus
Return value	None
Details	Changes the active focus to specified object.
File name	gwin.h

GCSNCPLCButtonSetAction

gncwin.h

Outline	Set PLC button operations
Syntax	void GCSNCPLCButtonSetAction(GBaseObject **pPLCButton,* unsigned char *ucAction*)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) unsigned char ucAction : PLC button operation
Return value	None
Details	Sets the PLC button operation. One of the following values is set for ucAction: GPLCBTN_ACT_NOACTION : no operation GPLCBTN_ACT_MOMENTARY : momentary operation GPLCBTN_ACT_ALTERNATE : alternative operation When a value outside of the range is set, there is no forced change of the setting.
File name	gncwin.h

GCSNCPLCButtonGetAction

gncwin.h

Outline	Get PLC button operations
Syntax	unsigned char GCSNCPLCButtonGetAction(GBaseObject **pPLCButton*)
Argument	(i) GBaseObject *pPLCButton object for operation
Return value	GPLCBTN_ACT_NOACTION : no operation GPLCBTN_ACT_MOMENTARY : momentary operation GPLCBTN_ACT_ALTERNATE : alternative operation Besides the above : illegal operation setting
Details	Gets the PLC button operation setting. When a setting outside the range is made, that setting value is returned.
File name	gncwin.h

GCSNCPLCButtonSetDisplay

gncwin.h

Outline	Set PLC button display
Syntax	void GCSNCPLCButtonSetDisplay(GBaseObject *pPLCButton, unsigned char ucDisplay)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) unsigned char ucDisplay : PLC button display
Return value	None
Details	Sets the PLC button display. One of the following values is set for ucDisplay: GPLCBTN_DISP_RECT : rectangle display GPLCBTN_DISP_OVAL : round shape display GPLCBTN_DISP_IMAGE : image display When a value outside of the range is set, there is no forced change of the setting. After the button display is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetDisplay

gncwin.h

Outline	Get PLC button display
Syntax	unsigned char GCSNCPLCButtonGetDisplay(GBaseObject **pPLCButton*)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	GPLCBTN_DISP_RECT : rectangle display GPLCBTN_DISP_OVAL : round shape display GPLCBTN_DISP_IMAGE : image display Besides the above : illegal display setting
Details	Gets the PLC button display setting. When a setting outside the range is made, that setting value is returned.
File name	gncwin.h

GCSNCPLCButtonSetBorderID

gncwin.h

Outline	Set 3D border resource ID
Syntax	void GCSNCPLCButtonSetBorderID(GBaseObject **pPLCButton,* unsigned short *usID*)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) unsigned short usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetBorderID

gncwin.h

Outline	Get 3D border resource ID
Syntax	unsigned short GCSNCPLCButtonGetBorderID(GBaseObject **pPLCButton*)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	3D border resource ID
Details	Gets the 3D border resource ID.
File name	gncwin.h

GCSNCPLCButtonSetOnDesign

gncwin.h

Outline	Set ON status design
Syntax	void GCSNCPLCButtonSetOnDesign(GBaseObject **pPLCButton,* GDesign **pDesign*)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) GDesign *pDesign : pointer to design
Return value	None
Details	Sets the design for ON status. After the design for ON status is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetOnDesign

gncwin.h

Outline	Get ON status design
Syntax	GDesign *GCSNCPLCButtonGetOnDesign(GBaseObject *pPLCButton)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	pointer to design
Details	Gets the design for ON status.
File name	gncwin.h

GCSNCPLCButtonSetOffDesign

gncwin.h

Outline	Set OFF status design
Syntax	void GCSNCPLCButtonSetOffDesign(GBaseObject *pPLCButton, GDesign *pDesign)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) GDesign *pDesign : pointer to design
Return value	None
Details	Sets the design for OFF status. After the design for OFF status is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetOffDesign

gncwin.h

Outline	Get OFF status design
Syntax	GDesign *GCSNCPLCButtonGetOffDesign(GBaseObject *pPLCButton)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	pointer to design
Details	Gets the design for OFF status.
File name	gncwin.h

GCSNCPLCButtonSetOnOnDesign

gncwin.h

Outline	Set ONON status design
Syntax	void GCSNCPLCButtonSetOnOnDesign(GBaseObject *pPLCButton, GDesign *pDesign)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) GDesign *pDesign : pointer to design
Return value	None
Details	Sets the design for ONON status. After the design for ONON status is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetOnOnDesign

gncwin.h

Outline	Get ONON status design
Syntax	GDesign *GCSNCPLCButtonGetOnOnDesign(GBaseObject *pPLCButton)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	pointer to design
Details	Gets the design for ONON status.
File name	gncwin.h

GCSNCPLCButtonSetOnOffDesign

gncwin.h

Outline	Set ONOFF status design
Syntax	void GCSNCPLCButtonSetOnOffDesign(GBaseObject *pPLCButton, GDesign *pDesign)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) GDesign *pDesign : pointer to design
Return value	None
Details	Sets the design for ONOFF status. After the design for ONOFF status is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetOnOffDesign

gncwin.h

Outline	Get ONOFF status design
Syntax	GDesign *GCSNCPLCButtonGetOnOffDesign(GBaseObject *pPLCButton)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	pointer to design
Details	Gets the design for ONOFF status.
File name	gncwin.h

GCSNCPLCButtonSetFocusDesign

gncwin.h

Outline	Set FOCUS status design
Syntax	void GCSNCPLCButtonSetFocusDesign(GBaseObject *pPLCButton, GDesign *pDesign)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) GDesign *pDesign : pointer to design
Return value	None
Details	Sets the design for FOCUS status. After the design for FOCUS status is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetFocusDesign

gncwin.h

Outline	Get FOCUS status design
Syntax	GDesign *GCSNCPLCButtonGetFocusDesign(GBaseObject *pPLCButton)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	pointer to design
Details	Gets the design for FOCUS status.
File name	gncwin.h

GCSNCPLCButtonSetDisableDesign

gncwin.h

Outline	Set Disable status design
Syntax	void GCSNCPLCButtonSetDisableDesign(GBaseObject *pPLCButton, GDesign *pDesign)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) GDesign *pDesign : pointer to design
Return value	None
Details	Sets the design for Disable status. After the design for Disable status is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetDisableDesign

gncwin.h

Outline	Get Disable status design
Syntax	GDesign *GCSNCPLCButtonGetDisableDesign(GBaseObject *pPLCButton*)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	pointer to design
Details	Gets the design for Disable status.
File name	gncwin.h

GCSNCPLCButtonSetFontID

gncwin.h

Outline	Set font resource ID
Syntax	void GCSNCPLCButtonSetFontID(GBaseObject *pPLCButton,* unsigned short *usID*)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) unsigned short usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetFontID

gncwin.h

Outline	Get font resource ID
Syntax	unsigned short GCSNCPLCButtonGetFontID(GBaseObject **pPLCButton*)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	Font resource ID
Details	Gets the font resource ID.
File name	gncwin.h

GCSNCPLCButtonSetOnStringID

gncwin.h

Outline	Set ON status caption character string resource ID
Syntax	void GCSNCPLCButtonSetOnStringID(GBaseObject **pPLCButton,* unsigned short *usID*)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) unsigned short usID : caption character string resource ID
Return value	None
Details	Sets the ON status caption character string resource ID. After the ON status caption character string resource ID is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetOnStringID

gncwin.h

Outline	Get ON status caption character string resource ID
Syntax	unsigned short GCSNCPLCButtonGetOnStringID(GBaseObject **pPLCButton*)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	Caption character string resource ID
Details	Gets the ON status caption character string resource ID.
File name	gncwin.h

GCSNCPLCButtonSetOffStringID

gncwin.h

Outline	Set OFF status caption character string resource ID
Syntax	void GCSNCPLCButtonSetOffStringID(GBaseObject **pPLCButton,* unsigned short *usID*)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) unsigned short usID : caption character string resource ID
Return value	None
Details	Sets the OFF status caption character string resource ID. After the OFF status caption character string resource ID is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetOffStringID

gncwin.h

Outline	Get OFF status caption character string resource ID
Syntax	unsigned short GCSNCPLCButtonGetOffStringID(GBaseObject **pPLCButton*)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	Caption character string resource ID
Details	Gets the OFF status caption character string resource ID.
File name	gncwin.h

GCSNCPLCButtonSetOnOnStringID

gncwin.h

Outline	Set ONON status caption character string resource ID
Syntax	void GCSNCPLCButtonSetOnOnStringID(GBaseObject **pPLCButton,* unsigned short *usID*)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) unsigned short usID : caption character string resource ID
Return value	None
Details	Sets the ONON status caption character string resource ID. After the ONON status caption character string resource ID is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetOnOnStringID

gncwin.h

Outline	Get ONON status caption character string resource ID
Syntax	unsigned short GCSNCPLCButtonGetOnOnStringID(GBaseObject **pPLCButton*)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	Caption character string resource ID
Details	Gets the ONON status caption character string resource ID.
File name	gncwin.h

GCSNCPLCButtonSetOnOffStringID

gncwin.h

Outline	Set ONOFF status caption character string resource ID
Syntax	void GCSNCPLCButtonSetOnOffStringID(GBaseObject **pPLCButton,* unsigned short *usID*)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) unsigned short usID : caption character string resource ID
Return value	None
Details	Sets the ONOFF status caption character string resource ID. After the ONOFF status caption character string resource ID is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetOnOffStringID

gncwin.h

Outline	Get ONOFF status caption character string resource ID
Syntax	unsigned short GCSNCPLCButtonGetOnOffStringID(GBaseObject **pPLCButton*)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	Caption character string resource ID
Details	Gets the ONOFF status caption character string resource ID.
File name	gncwin.h

GCSNCPLCButtonSetCaption

gncwin.h

Outline	Set caption information
Syntax	void GCSNCPLCButtonSetCaption(GBaseObject **pPLCButton,* GCaption **pCaption*)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) GCaption *pCaption : pointer to caption information
Return value	None
Details	Sets pCaption to the caption information. After the caption information is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetCaption

gncwin.h

Outline	Get caption information
Syntax	GCaption *GCSNCPLCButtonGetCaption(GBaseObject *pPLCButton)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	pointer to caption information
Details	Gets the caption information setting.
File name	gncwin.h

GCSNCPLCButtonSetFocusEffect

gncwin.h

Outline	Set effect during focus
Syntax	void GCSNCPLCButtonSetFocusEffect(GBaseObject *pPLCButton, unsigned char ucFocusEffect)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) unsigned char ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect: GFOCUS_EFFECT_CHG_COLOR : design change GFOCUS_EFFECT_NON : no effect After the effect during focus is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetFocusEffect

gncwin.h

Outline	Get effect during focus
Syntax	unsigned char GCSNCPLCButtonGetFocusEffect(GBaseObject *pPLCButton)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	GFOCUS_EFFECT_CHG_COLOR : design change GFOCUS_EFFECT_NON : no effect Besides the above : illegal display setting
Details	Gets the effect used during focus.
File name	gncwin.h

GCSNCPLCButtonSetStatus

gncwin.h

Outline	Set object status
Syntax	void GCSNCPLCButtonSetStatus(GBaseObject *pPLCButton, unsigned char ucStatus)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) unsigned char ucStatus : Set object status
Return value	None
Details	Sets the GCNCPLCButton object ON/OFF status. After the GCNCPLCButton object ON/OFF is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCButtonGetStatus

gncwin.h

Outline	Get object status
Syntax	unsigned char GCSNCPLCButtonGetStatus(GBaseObject *pPLCButton)
Argument	(i) GBaseObject *pPLCButton : object for operation
Return value	Object status
Details	Gets the GCNCPLCButton object status.
File name	gncwin.h

GCSNCPLCButtonSetDeviceType

gncwin.h

Outline	Set PLC device operations
Syntax	short GCSNCPLCButtonSetDeviceType(GBaseObject *pPLCButton, short nIndex, unsigned char ucDeviceType)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) short nIndex : index No. (i) unsigned char ucDeviceType : PLC device type
Return value	FALSE : Abnormal (index No. illegal) TRUE : Normal
Details	Sets the PLC device type.
File name	gncwin.h

GCSNCPLCButtonGetDeviceType

gncwin.h

Outline	Get PLC device operations
Syntax	short GCSNCPLCButtonGetDeviceType (GBaseObject *pPLCButton, short nIndex, unsigned char *pDeviceType)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) short nIndex : index No. (i) unsigned char *pDeviceType : PLC device type
Return value	FALSE : Abnormal (index No. illegal) TRUE : Normal
Details	Gets the PLC device type.
File name	gncwin.h

GCSNCPLCButtonSetDevice

gncwin.h

Outline	Set PLC device
Syntax	short GCSNCPLCButtonSetDevice (GBaseObject *pPLCButton, short nIndex, char *szDevice)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) short nIndex : index No. (i) char *szDevice : PLC device
Return value	FALSE : Abnormal (index No. illegal) TRUE : Normal
Details	Sets the PLC device.
File name	gncwin.h

GCSNCPLCButtonGetDevice

gncwin.h

Outline	Get PLC device
Syntax	short GCSNCPLCButtonGetDevice(GBaseObject *pPLCButton, short nIndex, char *szDevice)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) short nIndex : index No. (i) char *szDevice : PLC device
Return value	FALSE : Abnormal (index No. illegal) TRUE : Normal
Details	Gets the PLC device.
File name	gncwin.h

GCSNCPLCButtonSetDeviceVal

gncwin.h

Outline	Set PLC device value
Syntax	short GCSNCPLCButtonSetDeviceVal(GBaseObject *pPLCButton, short nIndex, unsigned char ucDeviceVal e)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) short nIndex : index No. (i) unsigned char ucDeviceVal : PLC device value
Return value	FALSE : Abnormal (index No. illegal) TRUE : Normal
Details	Sets the PLC device value.
File name	gncwin.h

GCSNCPLCButtonGetDeviceVal

gncwin.h

Outline	Get PLC device value
Syntax	short GCSNCPLCButtonGetDeviceVal(GBaseObject *pPLCButton, short nIndex, unsigned char *pDeviceVal)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) short nIndex : index No. (i) unsigned char *pDeviceVal : PLC device value
Return value	FALSE : Abnormal (index No. illegal) TRUE : Normal
Details	Gets the PLC device value.
File name	gncwin.h

GCSNCPLCTextboxSetBorderID

gncwin.h

Outline	Set 3D border resource ID
Syntax	void GCSNCPLCTextboxSetBorderID(GBaseObject *pPLCButton, unsigned char usID)
Argument	(i) GBaseObject *pPLCButton : object for operation (i) unsigned char usID : 3D border resource ID
Return value	None
Details	Sets the 3D border resource ID. After the 3D border resource ID is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCTextboxGetBorderID

gncwin.h

Outline	Get 3D border resource ID
Syntax	unsigned short GCSNCPLCTextboxGetBorderID(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	3D border resource ID
Details	Gets the 3D border resource ID.
File name	gncwin.h

GCSNCPLCTextboxSetFontID

gncwin.h

Outline	Set font resource ID
Syntax	void GCSNCPLCTextboxSetFontID(GBaseObject *pPLCTextBox, unsigned short usID
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) unsigned short usID : font resource ID
Return value	None
Details	Sets the font resource ID. After the font resource ID is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCTextboxGetFontID

gncwin.h

Outline	Get font resource ID
Syntax	unsigned short GCSNCPLCTextboxGetFontID(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	Font resource ID
Details	Gets the font resource ID.
File name	gncwin.h

GCSNCPLCTextboxSetCaption

gncwin.h

Outline	Set caption information
Syntax	void GCSNCPLCTextboxSetCaption(GBaseObject *pPLCTextBox, GCaption *pCaption)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) GCaption *pCaption : pointer to caption information
Return value	None
Details	Sets pCaption to the caption information. After the caption information is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCTextboxGetCaption

gncwin.h

Outline	Get caption information
Syntax	GCaption *GCSNCPLCTextboxGetCaption(GBaseObject *pPLCTextBox*)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	pointer to caption information
Details	Gets the caption information setting.
File name	gncwin.h

GCSNCPLCTextboxSetFocusEffect

gncwin.h

Outline	Set effect during focus
Syntax	void GCSNCPLCTextboxSetFocusEffect(GBaseObject *pPLCTextBox,* unsigned char *ucFocusEffect*)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) unsigned char ucFocusEffect : effect during focus
Return value	None
Details	Sets the effect during focus. One of the following values is set for ucFocusEffect: GFOCUS_EFFECT_CHG_COLOR : design change GFOCUS_EFFECT_NON : no effect After the effect during focus is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCTextboxGetFocusEffect

gncwin.h

Outline	Get effect during focus
Syntax	unsigned char GCSNCPLCTextboxGetFocusEffect(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	GFOCUS_EFFECT_CHG_COLOR : design change GFOCUS_EFFECT_NON : no effect Besides the above : illegal display setting
Details	Gets the effect used during focus.
File name	gncwin.h

GCSNCPLCTextboxSetFocusColor

gncwin.h

Outline	Set background color during focus
Syntax	void GCSNCPLCTextboxSetFocusColor(GBaseObject *pPLCTextBox, GColor gcColor)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) GColor gcColor : color code for background color during focus
Return value	None
Details	Sets the background color during focus. After the background color during focus is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCTextboxGetFocusColor

gncwin.h

Outline	Get background color during focus
Syntax	GColor GCSNCPLCTextboxGetFocusColor(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	Color code for background color during focus
Details	Gets the background color during focus.
File name	gncwin.h

GCSNCPLCTextboxSetNormalColor

gncwin.h

Outline	Set normal background color
Syntax	void GCSNCPLCTextboxSetNormalColor(GBaseObject *pPLCTextBox, GColor gcColor)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) GColor gcColor : color code for normal background color
Return value	None
Details	Sets the normal background color. After the normal background color is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCTextboxGetNormalColor

gncwin.h

Outline	Get normal background color
Syntax	GColor GCSNCPLCTextboxGetNormalColor(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	Color code for normal background color
Details	Gets the normal background color.
File name	gncwin.h

GCSNCPLCTextboxSetDisableColor

gncwin.h

Outline	Set background color when disabled
Syntax	void GCSNCPLCTextboxSetDisableColor(GBaseObject *pPLCTextBox GColor gcColor)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) GColor gcColor : color code for background color when disabled
Return value	None
Details	Sets the background color when disabled. After the background color when disabled is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCTextboxGetDisableColor

gncwin.h

Outline	Get background color when disabled
Syntax	GColor GCSNCPLCTextboxGetDisableColor(GBaseObject *pPLCTextBox*)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	Color code for background color setting when disabled
Details	Gets the background color when disabled.
File name	gncwin.h

GCSNCPLCTextboxSetTextType

gncwin.h

Outline	Set character string type
Syntax	void GCSNCPLCTextboxSetTextType(GBaseObject *pPLCTextBox,* unsigned char *ucType*)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) unsigned char ucType : character string type
Return value	None
Details	Sets the display character string type. One of the following values is set for ucType: GPLCTEXT_TYPE_DEC : Signed decimal integer GPLCTEXT_TYPE_UDEC : Unsigned decimal integer GPLCTEXT_TYPE_HEX : Hexadecimal integer GPLCTEXT_TYPE_FLOAT : Real number After the character string type is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCTextboxGetTextType

gncwin.h

Outline	Get character string type
Syntax	unsigned char GCSNCPLCTextboxGetTextType(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	GPLCTEXT_TYPE_DEC : Signed decimal integer GPLCTEXT_TYPE_UDEC : Unsigned decimal integer GPLCTEXT_TYPE_HEX : Hexadecimal integer GPLCTEXT_TYPE_FLOAT : Real number Besides the above : illegal setting
Details	Gets the display character string type.
File name	gncwin.h

GCSNCPLCTextboxSetPasswordStatus

gncwin.h

Outline	Set password display status
Syntax	void GCSNCPLCTextboxSetPasswordStatus(GBaseObject *pPLCTextBox, short fStatus)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) short fStatus : password display status FALSE : normal display status TRUE : password display status
Return value	None
Details	Sets whether the password will be displayed for text boxes or not. After the password display status is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCTextboxGetPasswordStatus

gncwin.h

Outline	Get password display status
Syntax	short GCSNCPLCTextboxGetPasswordStatus(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	FALSE : normal display status TRUE : password display status
Details	Gets whether the test box display status is normal display status or the password display status.
File name	gncwin.h

GCSNCPLCTextboxSetCommaStatus

gncwin.h

Outline	Set comma display status
Syntax	void GCSNCPLCTextboxSetCommaStatus(GBaseObject *pPLCTextBox, short fStatus)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) short fStatus : comma display status FALSE : normal display status TRUE : comma display status
Return value	None
Details	Sets whether commas will be displayed or not for text boxes. After the comma display status is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCTextboxGetCommaStatus

gncwin.h

Outline	Get comma display status
Syntax	short GCSNCPLCTextboxGetCommaStatus(GBaseObject *pPLCTextBox*)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	FALSE : normal display status TRUE : comma display status
Details	Gets the setting as to whether the test box display status is normal display status or the comma display status.
File name	gncwin.h

GCSNCPLCTextboxSetCheckMaxStatus

gncwin.h

Outline	Set maximum value check status
Syntax	void GCSNCPLCTextboxSetCheckMaxStatus(GBaseObject *pPLCTextBox,* short *fStatus*)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) short fStatus : setting for maximum value check status FALSE : check disabled TRUE : check enabled
Return value	None
Details	Sets the status of the maximum value check during SetValue/GetValue for the PLC text box.
File name	gncwin.h

GCSNCPLCTextboxGetCheckMaxStatus

gncwin.h

Outline	Get maximum value check status
Syntax	short GCSNCPLCTextboxGetCheckMaxStatus(GBaseObject **pPLCTextBox*)
Argument	(i) GBaseObject *pPLCTextBox
Return value	FALSE : check disabled TRUE : check enabled
Details	Gets the status of the maximum value check during SetValue/GetValue from the PLC text box.
File name	gncwin.h

GCSNCPLCTextboxSetCheckMinStatus

gncwin.h

Outline	Set minimum value check status
Syntax	void GCSNCPLCTextboxSetCheckMinStatus(GBaseObject **pPLCTextBox,* short *fStatus*)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) short fStatus : setting for minimum value check status FALSE : check disabled TRUE : check enabled
Return value	None
Details	Sets the status of the maximum value check during SetValue/GetValue for the PLC text box.
File name	gncwin.h

GCSNCPLCTextboxGetCheckMinStatus

gncwin.h

Outline	Get minimum value check status
Syntax	short GCSNCPLCTextboxGetCheckMinStatus(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	FALSE : check disabled TRUE : check enabled
Details	Gets the status of the minimum value check during SetValue/GetValue from the PLC text box.
File name	gncwin.h

GCSNCPLCTextboxSetMaxGValue

gncwin.h

Outline	Set maximum value
Syntax	void GCSNCPLCTextboxSetMaxGValue(GBaseObject *pPLCTextBox, GValue gvValue)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) GValue gvValue : maximum value
Return value	None
Details	Sets the value of the maximum value check during SetValue/GetValue for the PLC text box.
File name	gncwin.h

GCSNCPLCTextboxGetMaxGValue

gncwin.h

Outline	Get maximum value
Syntax	GValue GCSNCPLCTextboxGetMaxGValue(GBaseObject *pPLCTextBox*)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	Maximum value (GValue)
Details	Gets the value of the maximum value check during SetValue/GetValue from the PLC text box.
File name	gncwin.h

GCSNCPLCTextboxSetMinGValue

gncwin.h

Outline	Set minimum value
Syntax	void GCSNCPLCTextboxSetMinGValue(GBaseObject *pPLCTextBox,* GValue *gvValue*)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) GValue gvValue : minimum value
Return value	None
Details	Sets the value of the minimum value check during SetValue/GetValue for the PLC text box.
File name	gncwin.h

GCSNCPLCTextboxGetMinGValue

gncwin.h

Outline	Get minimum value
Syntax	GValue GCSNCPLCTextboxGetMinGValue(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	Minimum value (GValue)
Details	Gets the value of the minimum value check during SetValue/GetValue from the PLC text box.
File name	gncwin.h

GCSNCPLCTextboxSetString

gncwin.h

Outline	Set display character string
Syntax	short GCSNCPLCTextboxSetString(GBaseObject *pPLCTextBox, const GTCHAR *pString)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) const char *pString : display character string
Return value	FALSE : setting failed TRUE : setting succeeded
Details	Sets the display character string.
File name	gncwin.h

GCSNCPLCTextboxGetString

gncwin.h

Outline	Get display character string
Syntax	short GCSNCPLCTextboxGetString(GBaseObject *pPLCTextBox, GTCHAR *pString, long lMaxCount)
Argument	(i) GbaseObject *pPLCTextBox : object for operation (o) char *pString : buffer to store display character string (i) long lMaxCount : buffer size
Return value	FALSE : acquisition failed TRUE : acquisition succeeded
Details	Stores the display character string in pString.
File name	gncwin.h

GCSNCPLCTextboxSetGValue

gncwin.h

Outline	Set display value
Syntax	short GCSNCPLCTextboxSetGValue(GBaseObject **pPLCTextBox,* GValue *gvValue*)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) GValue gvValue : setting value
Return value	0 : normal value GTEXT_ERROR_UNDER_MIN_LIMIT : less than minimum value GTEXT_ERROR_OVER_MAX_LIMIT : greater than maximum value
Details	Sets the value displayed in a text box. Numbers are converted to character strings according to the display format. At this time, the maximum value and minimum value are checked, and if it is outside of the range, an error is returned in the return value. The error check is executed as follows: GTEXT_TYPE_SHORT GTEXT_TYPE_USHORT GTEXT_TYPE_LONG GTEXT_TYPE_ULONG After the value displayed in a text box is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCTextboxGetGValue

gncwin.h

Outline

Get display value

Syntax

```
short GCSNCPLCTextboxGetGValue(
  GBaseObject    *pPLCTextBox,
  GValue         *gvValue
)
```

Argument

(i) GBaseObject *pPLCTextBox : object for operation
(i) GValue *gvValue : pointer to GValue variable stored value

Return value

0	: normal value
GTEXT_ERROR_UNDER_MIN_LIMIT	: less than minimum value
GTEXT_ERROR_OVER_MAX_LIMIT	: greater than maximum value

Details

Gets the value displayed in a text box.
Character strings are converted to numbers according to the display format.
At this time, the maximum value and minimum value are checked, and if it is outside of the range, an error is returned in the return value. The error check is executed as follows:
GTEXT_TYPE_SHORT
GTEXT_TYPE_USHORT
GTEXT_TYPE_LONG
GTEXT_TYPE_ULONG

File name

gncwin.h

GCSNCPLCTextboxSetStringBuffer

gncwin.h

Outline	Set display buffer
Syntax	short GCSNCPLCTextboxSetStringBuffer(GBaseObject *pPLCTextBox, unsigned long ulBufferSize, GTCHAR *pStringBuffer)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) unsigned long ulBufferSize : buffer size (i) char *pStringBuffer : character string buffer
Return value	FALSE : setting failed TRUE : setting succeeded
Details	Sets the display buffer. Allocates memory internally when pStringBuffer is NULL. When a character string buffer is specified, that buffer is set.
File name	gncwin.h

GCSNCPLCTextboxGetStringBuffer

gncwin.h

Outline	Get display buffer
Syntax	GTCHAR *GCSNCPLCTextboxGetStringBuffer(GBaseObject *pPLCTextBox, unsigned long *ulBufferSize)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) unsigned long *ulBufferSize : pointer to variable stored buffer size
Return value	Display buffer address
Details	Gets the display buffer.
File name	gncwin.h

GCSNCPLCTextboxSetBackGroundPattern

gncwin.h

Outline	Set background fill status
Syntax	void GCSNCPLCTextboxSetBackGroundPattern(GBaseObject *pPLCTextBox, short nBackGroundPattern)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) short nBackGroundPattern : background fill status
Return value	None
Details	Sets the background color fill status. NO_DRAW and FILL_BACK_COLOR are the corresponding fill statuses.
File name	gncwin.h

GCSNCPLCTextboxGetBackGroundPattern

gncwin.h

Outline	Get background fill status
Syntax	short GCSNCPLCTextboxGetBackGroundPattern(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	Background fill status NO_DRAW : no fill FILL_BACK_COLOR : background fill
Details	Sets the background color fill status. NO_DRAW and FILL_BACK_COLOR are the corresponding fill statuses.
File name	gncwin.h

GCSNCPLCTextboxSetSelectStringStartPos

gncwin.h

Outline	Set selected character string start position
Syntax	void GCSNCPLCTextboxSetSelectStringStartPos(GBaseObject *pPLCTextBox, unsigned short usStartPos)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) unsigned short usStartPos : selected character string start position
Return value	None
Details	Sets the selected character string start position.
File name	gncwin.h

GCSNCPLCTextboxGetSelectStringStartPos

gncwin.h

Outline	Get selected character string start position
Syntax	unsigned short GCSNCPLCTextboxGetSelectStringStartPos(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	Selected character string start position
Details	Gets the selected character string start position.
File name	gncwin.h

GCSNCPLCTextboxSetSelectStringEndPos

gncwin.h

Outline	Set selected character string end position
Syntax	void GCSNCPLCTextboxSetSelectStringEndPos(GBaseObject *pPLCTextBox, unsigned short usEndPos)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) unsigned short usEndPos : selected character string end position
Return value	None
Details	Sets the selected character string end position.
File name	gncwin.h

GCSNCPLCTextboxGetSelectStringEndPos

gncwin.h

Outline	Get selected character string end position
Syntax	unsigned short GCSNCPLCTextboxGetSelectStringEndPos(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	Selected character string end position
Details	Gets the selected character string end position.
File name	gncwin.h

GCSNCPLCTextboxSetIntegerWidth

gncwin.h

Outline	Set integer part width
Syntax	void GCSNCPLCTextboxSetIntegerWidth(GBaseObject *pPLCTextBox, unsigned char ucWidth)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) unsigned char ucWidth : number of integer part figures
Return value	None
Details	Sets the number of integer part figures.
File name	gncwin.h

GCSNCPLCTextboxGetIntegerWidth

gncwin.h

Outline	Get integer part width
Syntax	unsigned char GCSNCPLCTextboxGetIntegerWidth(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	The number of integer part figures
Details	Gets the number of integer part figures.
File name	gncwin.h

GCSNCPLCTextboxSetDecimalWidth

gncwin.h

Outline	Set decimal part width
Syntax	void GCSNCPLCTextboxSetDecimalWidth(GBaseObject *pPLCTextBox, unsigned char ucWidth)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) unsigned char ucWidth : number of decimal part figures
Return value	None
Details	Sets the number of decimal part figures.
File name	gncwin.h

GCSNCPLCTextboxGetDecimalWidth

gncwin.h

Outline	Get decimal part width
Syntax	unsigned char GCSNCPLCTextboxGetDecimalWidth (GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	The number of decimal part figures
Details	Gets the number of decimal part figures.
File name	gncwin.h

GCSNCPLCTextboxSetScale

gncwin.h

Outline	Set scale
Syntax	void GCSNCPLCTextboxSetScale(GBaseObject *pPLCTextBox, double *pScale)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) double *pScale : scale
Return value	None
Details	Sets the scale.
File name	gncwin.h

GCSNCPLCTextboxGetScale

gncwin.h

Outline	Get scale
Syntax	void GCSNCPLCTextboxGetScale(GBaseObject *pPLCTextBox, double *pScale)
Argument	(i) GbaseObject *pPLCTextBox : object for operation (i) double *pScale : scale
Return value	None
Details	Gets the scale.
File name	gncwin.h

GCSNCPLCTextboxSetOffset

gncwin.h

Outline	Set offset
Syntax	void GCSNCPLCTextboxSetOffset(GBaseObject *pPLCTextBox, long lOffset)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) long lOffset : offset
Return value	None
Details	Sets the offset.
File name	gncwin.h

GCSNCPLCTextboxGetOffset

gncwin.h

Outline	Get offset
Syntax	long GCSNCPLCTextboxGetOffset(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	Offset
Details	Gets the offset.
File name	gncwin.h

GCSNCPLCTextboxSetDevice

gncwin.h

Outline	Set PLC device
Syntax	short GCSNCPLCTextboxSetDevice(GBaseObject *pPLCTextBox, char *szDevice)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) char *szDevice : PLC device
Return value	TRUE : acquisition succeeded FALSE : acquisition failed
Details	Sets the PLC device.
File name	gncwin.h

GCSNCPLCTextboxGetDevice

gncwin.h

Outline	Get PLC device
Syntax	short GCSNCPLCTextboxGetDevice(GBaseObject *pPLCTextBox, char *szDevice)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) char *szDevice : PLC device
Return value	TRUE : acquisition succeeded FALSE : acquisition failed
Details	Gets the PLC device.
File name	gncwin.h

GCSNCPLCTextboxSetTextSize

gncwin.h

Outline	Set size
Syntax	void GCSNCPLCTextboxSetTextSize(GBaseObject *pPLCTextBox, unsigned long ulSize)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) unsigned long ulSize : PLC device size
Return value	None
Details	Sets the PLC device size. One of the following values is set for ucType: GPLCTEXT_SIZE_2 : 2 byte GPLCTEXT_SIZE_4 : 4 byte After the PLC device size is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCTextboxGetTextSize

gncwin.h

Outline	Get size
Syntax	unsigned long GCSNCPLCTextboxGetTextSize(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	GPLCTEXT_SIZE_2 : 2 byte GPLCTEXT_SIZE_4 : 4 byte Besides the above : illegal setting
Details	Gets the PLC device size.
File name	gncwin.h

GCSNCPLCTextboxSetZeroSuppressStatus

gncwin.h

Outline	Set zero suppress display status
Syntax	void GCSNCPLCTextboxSetZeroSuppressStatus(GBaseObject *pPLCTextBox, short fStatus)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) short fStatus : zero suppress display status FALSE : normal display status TRUE : zero suppress display status
Return value	None
Details	Sets whether the zero suppress will be displayed for text boxes or not. After the zero suppress display status is set, the control area is registered as a redraw area.
File name	gncwin.h

GCSNCPLCTextboxGetZeroSuppressStatus

gncwin.h

Outline	Get zero suppress display status
Syntax	short GCSNCPLCTextboxGetZeroSuppressStatus(GBaseObject *pPLCTextBox)
Argument	(i) GBaseObject *pPLCTextBox : object for operation
Return value	FALSE : normal display status TRUE : zero suppress display status
Details	Gets whether the test box display status is normal display status or the zero suppress display status.
File name	gncwin.h

GCSNCTableSetCellForeColor

gncwin.h

Outline	Set foreground color of each cell
Syntax	void GCSNCTableSetCellForeColor(GBaseObject *pSelf, GColor gcColor, unsigned short usLine, unsigned short usRow, unsigned short *pusErrCode)
Argument	(i) GBaseObject *pPLCTextBox : object for operation (i) Gcolor gcColor : color (i) unsigned short usLine : line No. (0 to) (i) unsigned short usRow : row No. (0 to) (o) unsigned short *pusErrCode : error code GERR_NOERR : normal GERR_NCTABLE_RANGEOVER : cell No. outside range
Return value	None
Details	Sets the foreground color of each cell.
File name	gncwin.h

GCSNCTableSetCellBackColor

gncwin.h

Outline

Set background color of each cell

Syntax

```
void GCSNCTableSetCellBackColor(
  GBaseObject     *pSelf,
  GColor          gcColor,
  unsigned short  usLine,
  unsigned short  usRow,
  unsigned short  *pusErrCode
)
```

Argument

(i) GBaseObject *pSelf : object for operation
(i) Gcolor gcColor : color
(i) unsigned short usLine : line No. (0 to)
(i) unsigned short usRow : row No. (0 to)
(o) unsigned short *pusErrCode : error code
GERR_NOERR : normal
GERR_NCTABLE_RANGEOVER : cell No. outside range

Return value

None

Details

Sets the background color of each cell.

File name

gncwin.h

GCSNCTableSetCellString

gncwin.h

Outline	Set character string of each cell
Syntax	void GCSNCTableSetCellString(GBaseObject **pSelf,* const GTCHAR **pString,* unsigned short *usLine,* unsigned short *usRow,* unsigned short **pusErrCode*)
Argument	(i) GBaseObject *pSelf : object for operation (i) const GTCHAR *pString : set character string (0 to 128 characters) (i) unsigned short usLine : line No. (0 to) (i) unsigned short usRow : row No. (0 to) (o) unsigned short *pusErrCode : error code GERR_NOERR : normal GERR_NCTABLE_RANGEOVER : cell No. outside range GERR_NCTABLE_MEMORYERROR : securing memory failed
Return value	None
Details	Sets the character string of each cell.
File name	gncwin.h

GCSNCTableGetCellString

gncwin.h

Outline

Get character string of each cell

Syntax

```
void GCSNCTableGetCellString(
  GBaseObject     *pSelf,
  GTCHAR          *pString,
  long            lMaxCount,
  unsigned short  usLine,
  unsigned short  usRow,
  unsigned short  *pusErrCode
)
```

Argument

(i) GBaseObject *pSelf	: object for operation
(o) GTCHAR *pString	: character string to be acquired
(i) long lMaxCount	: number of characters to be acquired
(i) unsigned short usLine	: line No. (0 to)
(i) unsigned short usRow	: row No. (0 to)
(o) unsigned short *pusErrCode	: error code
GERR_NOERR	: normal
GERR_NCTABLE_RANGEOVER	: cell No. outside range
GERR_NCTABLE_WRONGSTRINGGETSIZE	: character string size to be acquired illegal

Return value

None

Details

Gets the character string of each cell.

File name

gncwin.h

GCSNCTableSetSubCursorNum

gncwin.h

Outline	Set sub-cursor position
Syntax	void GCSNCTableSetSubCursorNum(GBaseObject **pSelf,* unsigned short *usLine,* unsigned short *usRow,* unsigned short **pusErrCode*)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned short usLine : line No. (0 to) (i) unsigned short usRow : row No. (0 to) (o) unsigned short *pusErrCode : error code GERR_NOERR : normal GERR_NCTABLE_DISABLESUBCURSOR : sub-cursor invalid GERR_NCTABLE_RANGEOVER : cell No. outside range
Return value	None
Details	Sets the sub-cursor position.
File name	gncwin.h

GCSNCTableGetSubCursorNum

gncwin.h

Outline	Get sub-cursor position
Syntax	void GCSNCTableGetSubCursorNum(GBaseObject **pSelf,,* unsigned short **uspLine,* unsigned short **usRow,* unsigned short **pusErrCode*)
Argument	(i) GBaseObject *pSelf : object for operation (i) unsigned short *uspLine : line No. (0 to) (i) unsigned short *usRow : row No. (0 to) (o) unsigned short *pusErrCode : error code GERR_NOERR : normal GERR_NCTABLE_DISABLESUBCURSOR : sub-cursor invalid
Return value	None
Details	Gets the sub-cursor position.
File name	gncwin.h

GCSNCTableSetCellWidth

gncwin.h

Outline

Set cell width

Syntax

```
void GCSNCTableSetCellWidth(
  GBaseObject     *pSelf,
  unsigned short  usRowNum,
  unsigned short  usWidth,
  unsigned short  usRatioMode,
  unsigned short  *pusErrCode
)
```

Argument

(i) GbaseObject *pSelf : object for operation
(i) unsigned short usRowNum : row No. (0 to)
(i) unsigned short usWidth : row width
(i) unsigned short usRatioMode : row proportion mode (0:proportion 1:pixel)
(o) unsigned short *pusErrCode : error code

GERR_NOERR	: normal
GERR_NCTABLE_RANGEOVER	: cell No. outside range

Return value

None

Details

Sets the cell width.

File name

gncwin.h

GCSNCTableSetCellHight

gncwin.h

Outline	Set cell height
Syntax	void GCSNCTableSetCellHight (GBaseObject **pSelf,* unsigned short *usHight,* unsigned short **pusErrCode*)
Argument	(i) GbaseObject *pSelf : object for operation (i) unsigned short usHight : Hight (o) unsigned short *pusErrCode : error code GERR_NOERR : normal
Return value	None
Details	Sets the cell height.
File name	gncwin.h

GCSNCTableGetCellNumFromPoint

gncwin.h

Outline Get cell No. including pertinent coordinates

Syntax

```
void GCSNCTableGetCellNumFromPoint(
  GBaseObject     *pSelf,
  GPoint          gptPoint,,
  unsigned short  *uspLine,
  unsigned short  *uspRow,
  unsigned short  *pusErrCode
)
```

Argument

(i) GbaseObject *pSelf : object for operation
(i) GPoint gptPoint : coordinates
(i) unsigned short *uspLine : line No. (0 to)
(i) unsigned short *uspRow : row No. (0 to)
(o) unsigned short *pusErrCode : error code
GERR_NOERR : normal
GERR_NCTABLE_RANGEOVER : cell No. outside range

Return value None

Details Gets the cell No. including pertinent coordinates.

File name gncwin.h

GCSNCTableSetLineTitleString

gncwin.h

Outline	Set line title
Syntax	void GCSNCTableSetLineTitleString(GBaseObject *pSelf, const GTCHAR *pString, unsigned short usLine, unsigned short *pusErrCode)
Argument	(i) GbaseObject *pSelf : object for operation (i) const GTCHAR *pString : set character string (i) unsigned short usLine : line No. (0 to) (o) unsigned short *pusErrCode : error code GERR_NOERR : normal GERR_NCTABLE_NOTITLEMODE : title non-display mode GERR_NCTABLE_RANGEOVER : cell No. outside range GERR_NCTABLE_MEMORYERROR : securing memory failed
Return value	None
Details	Sets the line title.
File name	gncwin.h

GCSNCTableGetLineTitleString

gncwin.h

Outline Get line title

Syntax

```
void GCSNCTableGetLineTitleString(
  GBaseObject     *pSelf,
  GTCHAR          *pString,
  long            lMaxCount,
  unsigned short  usLine,
  unsigned short  *pusErrCode
)
```

Argument

(i) GbaseObject *pSelf	: object for operation
(o) GTCHAR *pString	: character string to be acquired
(i) long lMaxCount	: number of characters to be acquired
(i) unsigned short usLine	: line No. (0 to)
(o) unsigned short *pusErrCode	: error code
GERR_NOERR	: normal
GERR_NCTABLE_NOTITLEMODE	: title non-display mode
GERR_NCTABLE_RANGEOVER	: cell No. outside range
GERR_NCTABLE_MEMORYERROR	: securing memory failed

Return value None

Details Gets the line title.

File name gncwin.h

GCSNCTableSetRowTitleString

gncwin.h

Outline	Set row title
Syntax	void GCSNCTableSetRowTitleString(GBaseObject **pSelf,* GTCHAR **pString,* unsigned short *usRow,* unsigned short **pusErrCode*)
Argument	(i) GbaseObject *pSelf : object for operation (i) GTCHAR *pString : set character string (i) unsigned short usRow : row No. (0 to) (o) unsigned short *pusErrCode : error code GERR_NOERR : normal GERR_NCTABLE_NOTITLEMODE : title non-display mode GERR_NCTABLE_RANGEOVER : cell No. outside range GERR_NCTABLE_MEMORYERROR : securing memory failed
Return value	None
Details	Sets the row title.
File name	gncwin.h

GCSNCTableGetRowTitleString

gncwin.h

Outline

Get row title

Syntax

```
void GCSNCTableGetRowTitleString(
  GBaseObject     *pSelf,
  GTCHAR          *pString,
  long            lMaxCount,
  unsigned short  usRow,
  unsigned short  *pusErrCode
)
```

Argument

(i) GbaseObject *pSelf	: object for operation	
(o) GTCHAR *pString	: character string to be acquired	
(i) long lMaxCount	: number of characters to be acquired	
(i) unsigned short usRow	: row No. (0 to)	
(o) unsigned short *pusErrCode	: error code	
GERR_NOERR		: normal
GERR_NCTABLE_NOTITLEMODE		: title non-display mode
GERR_NCTABLE_RANGEOVER		: cell No. outside range
GERR_NCTABLE_WRONGSTRINGGETSIZE		: character string size to be acquired illegal

Return value

None

Details

Gets the row title.

File name

gncwin.h

GCSNCTableSetLineTitleStringPosition

gncwin.h

Outline Set line title character string position

Syntax

```
void GCSNCTableSetLineTitleStringPosition(
  GBaseObject     *pSelf,
  unsigned char   ucHPosition,
  unsigned char   ucVPosition,
  unsigned short  usLine,
  unsigned short  *pusErrCode
)
```

Argument

(i) GbaseObject *pSelf : object for operation
(i) unsigned char ucHPosition : character display position horizontal
(i) unsigned char ucVPosition : character display position vertical
(i) unsigned short usLine : line No. (0 to)
(o) unsigned short *pusErrCode : error code

GERR_NOERR	: normal
GERR_NCTABLE_NOTITLEMODE	: title non-display mode
GERR_NCTABLE_RANGEOVER	: cell No. outside range
GERR_NCTABLE_ WRONGVALUE	: value outside setting range

Return value None

Details Sets the line title character string position.

File name gncwin.h

GCSNCTableSetRowTitleStringPosition

gncwin.h

Outline Set row title character string position

Syntax

```
void GCSNCTableSetRowTitleStringPosition(
  GBaseObject     *pSelf,
  unsigned char   ucHPosition,
  unsigned char   ucVPosition,
  unsigned short  usRow,
  unsigned short  *pusErrCode
)
```

Argument

(i) GbaseObject *pSelf : object for operation
(i) unsigned char ucHPosition : character display position horizontal
(i) unsigned char ucVPosition : character display position vertical
(i) unsigned short usRow : row No. (0 to)
(o) unsigned short *pusErrCode : error code

GERR_NOERR	: normal
GERR_NCTABLE_NOTITLEMODE	: title non-display mode
GERR_NCTABLE_RANGEOVER	: cell No. outside range
GERR_NCTABLE_ WRONGVALUE	: value outside setting range

Return value None

Details Sets the row title character string position.

File name gncwin.h

GCSNCTableSetCellStringPosition

gncwin.h

Outline

Set area part character string position

Syntax

```
void GCSNCTableSetCellStringPosition(
  GBaseObject     *pSelf,
  unsigned char   ucHPosition,
  unsigned char   ucVPosition,
  unsigned short  usLine,
  unsigned short  usRow,
  unsigned short  *pusErrCode
)
```

Argument

(i) GbaseObject *pSelf : object for operation
(i) unsigned char ucHPosition : character display position horizontal
(i) unsigned char ucVPosition : character display position vertical
(i) unsigned short usLine : line No. (0 to)
(i) unsigned short usRow : row No. (0 to)
(o) unsigned short *pusErrCode : error code

GERR_NOERR	: normal
GERR_NCTABLE_RANGEOVER	: cell No. outside range
GERR_NCTABLE_ WRONGVALUE	: value outside setting range

Return value

None

Details

Sets the area part character string position.

File name

gncwin.h

GCSNCTableSetBeforeSubCursorNum

gncwin.h

Outline

Set last sub-cursor No.

Syntax

```
void GCSNCTableSetBeforeSubCursorNum(
  GBaseObject     *pSelf,
  unsigned short  usLine,
  unsigned short  usRow,
  unsigned short  *pusErrCode
)
```

Argument

(i) GbaseObject *pSelf : object for operation
(i) unsigned short usLine : line No. (0 to)
(i) unsigned short usRow : row No. (0 to)
(o) unsigned short *pusErrCode : error code

GERR_NOERR	: normal
GERR_NCTABLE_DISABLESUBCURSOR	: sub-cursor invalid
GERR_NCTABLE_RANGEOVER	: cell No. outside range

Return value

None

Details

Sets the last sub-cursor No.

File name

gncwin.h

GCSNCTableGetBeforeSubCursorNum

gncwin.h

Outline	Get last sub-cursor No.
Syntax	void GCSNCTableGetBeforeSubCursorNum(GBaseObject **pSelf,* unsigned short **usLine,* unsigned short **usRow,* unsigned short **pusErrCode*)
Argument	(i) GbaseObject *pSelf : object for operation (i) unsigned short *usLine : line No. (0 to) (i) unsigned short *usRow : row No. (0 to) (o) unsigned short *pusErrCode : error code GERR_NOERR : normal GERR_NCTABLE_DISABLESUBCURSOR : sub-cursor invalid
Return value	None
Details	Gets the last sub-cursor No.
File name	gncwin.h

GCSNCTableSetLineTitleForeColor

gncwin.h

Outline

Set foreground color of line title

Syntax

```
void GCSNCTableSetLineTitleForeColor(
  GBaseObject     *pSelf,
  GColor          gcColor,
  unsigned short  usLine,
  unsigned short  *pusErrCode
)
```

Argument

(i) GbaseObject *pSelf : object for operation
(i) GColor gcColor : color
(i) unsigned short usLine : line No. (0 to)
(o) unsigned short *pusErrCode : error code

GERR_NOERR	: normal
GERR_NCTABLE_NOTITLEMODE	: title non-display mode
GERR_NCTABLE_RANGEOVER	: cell No. outside range

Return value

None

Details

Sets the foreground color of line title.

File name

gncwin.h

GCSNCTableSetLineTitleBackColor

gncwin.h

Outline

Set background color of line title

Syntax

```
void GCSNCTableSetLineTitleBackColor(
  GBaseObject     *pSelf,
  GColor          gcColor,
  unsigned short  usLine,
  unsigned short  *pusErrCode
)
```

Argument

(i) GbaseObject *pSelf : object for operation
(i) GColor gcColor : color
(i) unsigned short usLine : line No. (0 to)
(o) unsigned short *pusErrCode : error code

GERR_NOERR	: normal
GERR_NCTABLE_NOTITLEMODE	: title non-display mode
GERR_NCTABLE_RANGEOVER	: cell No. outside range

Return value

None

Details

Sets the background color of line title.

File name

gncwin.h

GCSNCTableSetRowTitleForeColor

gncwin.h

Outline

Set foreground color of row title

Syntax

```
void GCSNCTableSetRowTitleForeColor(
  GBaseObject     *pSelf,
  GColor          gcColor,
  unsigned short  usRow,
  unsigned short  *pusErrCode
)
```

Argument

(i) GbaseObject *pSelf : object for operation
(i) GColor gcColor : color
(i) unsigned short usRow : row No. (0 to)
(o) unsigned short *pusErrCode : error code

GERR_NOERR	: normal
GERR_NCTABLE_NOTITLEMODE	: title non-display mode
GERR_NCTABLE_RANGEOVER	: cell No. outside range

Return value

None

Details

Sets the foreground color of row title.

File name

gncwin.h

GCSNCTableSetRowTitleBackColor

gncwin.h

Outline

Set background color of row title

Syntax

```
void GCSNCTableSetRowTitleBackColor(
  GBaseObject     *pSelf,
  GColor          gcColor,
  unsigned short  usRow,
  unsigned short  *pusErrCode
)
```

Argument

(i) GbaseObject *pSelf : object for operation
(i) GColor gcColor : color
(i) unsigned short usRow : row No. (0 to)
(o) unsigned short *pusErrCode : error code

GERR_NOERR	: normal
GERR_NCTABLE_NOTITLEMODE	: title non-display mode
GERR_NCTABLE_RANGEOVER	: cell No. outside range

Return value

None

Details

Sets the background color of row title.

File name

gncwin.h

GCSNCTableSetSubCursorForeColor

gncwin.h

Outline	Set foreground color of sub-cursor
Syntax	void GCSNCTableSetSubCursorForeColor(GBaseObject *pSelf, GColor gcColor, unsigned short *pusErrCode)
Argument	(i) GbaseObject *pSelf : object for operation (i) GColor gcColor : color (o) unsigned short *pusErrCode : error code GERR_NOERR : normal
Return value	None
Details	Sets the foreground color of sub-cursor.
File name	gncwin.h

GCSNCTableSetSubCursorBackColor

gncwin.h

Outline	Set background color of sub-cursor
Syntax	void GCSNCTableSetSubCursorBackColor(GBaseObject *pSelf, GColor gcColor, unsigned short *pusErrCode)
Argument	(i) GbaseObject *pSelf : object for operation (i) GColor gcColor : color (o) unsigned short *pusErrCode : error code GERR_NOERR : normal
Return value	None
Details	Sets the background color of sub-cursor.
File name	gncwin.h

6. GIO

This section describes the GIO function group.

6.1 List of GIP Functions

No.	Name	Type	File name	Outline
1	GIOOpen	HGIO	GIO.c	I/O open process
2	GIOClose	void	GIO.c	I/O close process
3	GIORead	unsigned long	GIO.c	I/O read
4	GIOWrite	unsigned long	GIO.c	I/O write
5	GIOSeek	long	GIO.c	File pointer movement

6.2 Outline of GIO Functions

GIOOpen

GIO.c

Outline	I/O open process
Syntax	HGIO GIOOpen(unsigned short *usDirType,* GTCHAR **pIOName,* GTCHAR **pOperation*)
Argument	(i) unsigned short usDirType : Standard directory (i) GTCHAR *pIOName : name of I/O being opened (i) GTCHAR *pOperation : opening operation
Return value	NULL : open failed Other than NULL : HGIO handle
Details	Carries out the process for opening file I/O.
File name	GIO.c

GIOClose

GIO.c

Outline	I/O close process
Syntax	void GIOClose(HGIO *hIO*)
Argument	(i) HGIO hIO : handle being closed
Return value	None
Details	Carries out the process for closing file I/O.
File name	GIO.c

GIORead

GIO.c

Outline	I/O read
Syntax	unsigned long GIORead(void *pBuffer, unsigned long ulSize, unsigned long ulCount, HGIO hIO)
Argument	(o) void *pBuffer : area for storing read results (i) unsigned long ulSize : read data size units (i) unsigned long ulCount : number of pieces of read data (i) HGIO hIO : file handle
Return value	Number of pieces of data read
Details	Carries out the reading of data from file I/O.
File name	GIO.c

GIOWrite

GIO.c

Outline	I/O write

Syntax

```
unsigned long GIOWrite(
  void          *pBuffer,
  unsigned long  ulSize,
  unsigned long  ulCount,
  HGIO           hIO
)
```

Argument	(i) void *pBuffer	: area for storing write results
	(i) unsigned long ulSize	: write data size units
	(i) unsigned long ulCount	: number of pieces of data written
	(i) HGIO hIO	: file handle

Return value	Number of pieces of data written
Details	Carries out the writing of data for file I/O.
File name	GIO.c

GIOSeek

GIO.c

Outline	File pointer movement
Syntax	long GIOSeek(HGIO *hIO*, long *lOffset*, long *lOrigin*)
Argument	(i) long lOffset : offset position from initial position (i) long lOrigin : initial position
Return value	Results from execution 0 : succeeded Other than 0 : failed
Details	Moves the file pointer to the specified position.
File name	GIO.c

7. GMMemory

This section describes the GMMemory function group.

7.1 List of GMMemory Functions

No.	Name	Type	File name	Outline
1	GMGMemCreate	GMGMEM	GMMemory.c	Create named shared memory area
2	GMGMemSelect	GMGMEM	GMMemory.c	Select memory area
3	GMGMemDelete	void	GMMemory.c	Release shared memory area
4	GMGMemGetShort	short	GMMemory.c	Get shared memory 16 bit integer
5	GMGMemSetShort	void	GMMemory.c	Set shared memory 16 bit integer
6	GMGMemGetChar	char	GMMemory.c	Get shared memory 8 bit integer
7	GMGMemSetChar	void	GMMemory.c	Set shared memory 8 bit integer
8	GMGMemGetLong	long	GMMemory.c	Get shared memory 32 bit integer
9	GMGMemSetLong	void	GMMemory.c	Set shared memory 32 bit integer
10	GMGMemGetDouble	double	GMMemory.c	Get shared memory 64 bit floating point value
11	GMGMemSetDouble	void	GMMemory.c	Set shared memory 64 bit floating point value
12	GMGMemGetString	GTCHAR *	GMMemory.c	Get shared memory character string
13	GMGMemSetString	void	GMMemory.c	Set shared memory character string

7.2 Outline of GMMemory Functions

GMGMemCreate

GMMemory.c

Outline	Create named shared memory area
Syntax	GMGMEM GMGMemCreate(GTCHAR *pszName, unsigned long ulSize)
Argument	(i) GTCHAR *pszName : shared memory name (i) unsigned long ulSize : allocation size
Return value	Allocated region
Details	Shared memory with the name specified in pszName and the size specified in ulSize is allocated.
File name	GMMemory.c

GMGMemSelect

GMMemory.c

Outline	Select memory area
Syntax	GMGMEM GMGMemSelect(GTCHAR *pszLangStr)
Argument	(i) GTCHAR *pszLangStr : shared memory name
Return value	Shared memory area
Details	Selects the region for the specified name from the shared memory areas previously created.
File name	GMMemory.c

GMGMemDelete

GMMemory.c

Outline	Release shared memory area
Syntax	void GMGMemDelete(GMGMEM *gmem*)
Argument	(i) GMGMEM gmem : shared memory area
Return value	None
Details	Releases the specified shared memory area.
File name	GMMemory.c

GMGMemGetShort

GMMemory.c

Outline	Get shared memory 16 bit integer
Syntax	short GMGMemGetShort(GMGMEM *gmem,* unsigned long *ulOffset*)
Argument	(i) GMGMEM gmem : shared memory area (i) unsigned long ulOffset : offset position
Return value	16 bit integer value
Details	Gets the ulOffset byte data from the beginning of the specified shared memory as a 16 bit integer value.
File name	GMMemory.c

GMGMemSetShort

GMMemory.c

Outline	Set shared memory 16 bit integer
Syntax	void GMGMemSetShort(GMGMEM *gmem,* unsigned long *ulOffset,* short *data*)
Argument	(i) GMGMEM gmem : shared memory area (i) unsigned long ulOffset : offset position (i) short data : setting value
Return value	None
Details	Sets a 16 bit integer value in the ulOffset byte data from the beginning of the specified shared memory.
File name	GMMemory.c

GMGMemGetChar

GMMemory.c

Outline	Get shared memory 8 bit integer
Syntax	char GMGMemGetChar(GMGMEM *gmem,* unsigned long *ulOffset*)
Argument	(i) GMGMEM gmem : shared memory area (i) unsigned long ulOffset : offset position
Return value	8 bit integer value
Details	Gets the ulOffset byte data from the beginning of the specified shared memory as an 8 bit integer value.
File name	GMMemory.c

GMGMemSetChar

GMMemory.c

Outline	Set shared memory 8 bit integer
Syntax	void GMGMemSetChar(GMGMEM *gmem,* unsigned long *ulOffset,* char *data*)
Argument	(i) GMGMEM gmem : shared memory area (i) unsigned long ulOffset : offset position (i) char data : setting value
Return value	None
Details	Sets an 8 bit integer value in the ulOffset byte data from the beginning of the specified shared memory.
File name	GMMemory.c

GMGMemGetLong

GMMemory.c

Outline	Get shared memory 32 bit integer
Syntax	long GMGMemGetLong(GMGMEM *gmem,* unsigned long *ulOffset*)
Argument	(i) GMGMEM gmem : shared memory area (i) unsigned long ulOffset : offset position
Return value	32 bit integer value
Details	Gets the ulOffset byte data from the beginning of the specified shared memory as a 32 bit integer value.
File name	GMMemory.c

GMGMemSetLong

GMMemory.c

Outline	Set shared memory 32 bit integer
Syntax	void GMGMemSetLong(GMGMEM *gmem,* unsigned long *ulOffset,* long *data*)
Argument	(i) GMGMEM gmem : shared memory area (i) unsigned long ulOffset : offset position (i) long data : setting value
Return value	None
Details	Sets a 32 bit integer value in the ulOffset byte data from the beginning of the specified shared memory.
File name	GMMemory.c

GMGMemGetDouble

GMMemory.c

Outline	Get shared memory 64 bit floating point value
Syntax	double GMGMemGetDouble(GMGMEM *gmem,* unsigned long *ulOffset*)
Argument	(i) GMGMEM gmem : shared memory area (i) unsigned long ulOffset : offset position
Return value	64 bit floating point value
Details	Gets the ulOffset byte memory from the beginning of the specified shared memory as a 64 bit floating point value.
File name	GMMemory.c

GMGMemSetDouble

GMMemory.c

Outline	Set shared memory 64 bit floating point value
Syntax	void GMGMemSetDouble(GMGMEM *gmem,* unsigned long *ulOffset,* double *data*)
Argument	(i) GMGMEM gmem : shared memory area (i) unsigned long ulOffset : offset position (i) double data : setting value
Return value	None
Details	Sets a 64 bit floating point value in the ulOffset byte data from the beginning of the specified shared memory.
File name	GMMemory.c

GMGMemGetString

GMMemory.c

Outline	Get shared memory area
Syntax	GTCHAR *GMGMemGetString(GMGMEM *gmem,* unsigned long *ulOffset*)
Argument	(i) GMGMEM gmem : shared memory area (i) unsigned long ulOffset : offset position (o) GTCHAR *data : character string acquisition region
Return value	None
Details	Gets the ulOffset byte data from the beginning of the specified shared memory as a character string.
File name	GMMemory.c

GMGMemSetString

GMMemory.c

Outline	Set shared memory character string
Syntax	void GMGMemSetString(GMGMEM *gmem,* unsigned long *ulOffset,* GTCHAR **data*)
Argument	(i) GMGMEM gmem : shared memory area (i) unsigned long ulOffset : offset position (i) GTCHAR *data : setting value
Return value	None
Details	Sets a character string in the ulOffset byte data from the beginning of the specified shared memory.
File name	GMMemory.c

Appendix 1. Data Type Definitions

The definitions for various structure data types are given in the following.

```
#define   GColor          long                /* color variable                    */
#define   HGFONT          GFontHandle*        /* 2 font handle                     */
#define   HGDRAW          GDraw*              /* drawing handle                    */

/*****************************************************************************
 * Rectangle structure
 *****************************************************************************/
typedef struct _GRect{
        short             nXmin;              /* upper left X coordinate           */
        short             nYmin;              /* upper left Y coordinate           */
        short             nXmax;              /* lower right X coordinate          */
        short             nYmax;              /* lower right Y coordinate          */
}GRect;

/*****************************************************************************
 * Point structure
 *****************************************************************************/
typedef struct _GPoint{
        short             nX;                 /* X coordinate                      */
        short             nY;                 /* Y coordinate                      */
}GPoint;

/*****************************************************************************
 * Line structure
 *****************************************************************************/
typedef struct _GLine{
        short             nX1;                /* X coordinate for starting point   */
        short             nY1;                /* Y coordinate for starting point   */
        short             nX2;                /* X coordinate for ending point     */
        short             nY2;                /* Y coordinate for ending point     */
}GLine;
```

```
/*****************************************************************************
 * Polygon structure
 *****************************************************************************/
typedef struct _GPoly{
        short           nNumPoints;     /* number of vertices                   */
        GPoint          *pgptPoints;    /* pointer to vertex data               */
        GRect           grBounds;       /* outline rectangle                    */
}GPoly;

/*****************************************************************************
 * Brush structure
 *****************************************************************************/
typedef struct _GBrush{
        short           nFillPattern;   /* fill pattern                         */
        GColor          gcForeColor;    /* fill foreground color                */
        GColor          gcBackColor;    /* fill background color                */
}GBrush;

/*****************************************************************************
 * Border structure
 *****************************************************************************/
typedef struct _GBorder{
        unsigned char   fBorder;        /* 3D border present or absent          */
        GColor          gcULColor;      /* upper left border color              */
        GColor          gcLRColor;      /* lower right border color             */
        GColor          gcLineColor;    /* line color                           */
        short           nSize;          /* 3D border size                       */
}GBorder;
```

```
/*****************************************************************************
 * Caption structure
 *****************************************************************************/
typedef struct _GCaption{
        GColor          gcColor;            /* caption character color          */
        unsigned char   ucHPosition;        /* horizontal display position      */
        unsigned char   ucVPosition;        /* vertical display position        */
        short           nLeftMargin;        /* left margin                      */
        short           nRightMargin;       /* right margin                     */
        short           nTopMargin;         /* top margin                       */
        short           nBottomMargin;      /* bottom margin                    */
}GCaption;

/*****************************************************************************
 * Cursor structure
 *****************************************************************************/
typedef struct _GCursor{
        unsigned char   ucType;             /* cursor type                      */
        GColor          gcColor;            /* cursor color                     */
}GCursor;

/*****************************************************************************
 * Font structure
 *****************************************************************************/
typedef struct _GFontSize{
        short           nAscent;            /* height from baseline to top line */
        short           nDscent;            /* height from baseline to bottom line */
        unsigned short  usWidth;            /* basic character width            */
        unsigned short  usHeight;           /* character height                 */
        unsigned short  usMaxWidth;         /* maximum character width          */
}GFontSize;
```

```
/*****************************************************************************
 * Simple font structure
 *****************************************************************************/
typedef struct _GSimpleFont{
        GTCHAR          szFontName[MAX_FONTNAME_LEN+1];   /* font name          */
        unsigned char   ucSize;                 /* font size                    */
        unsigned char   gptXScale;              /* horizontal scale             */
        unsigned char   gptYScale;              /* vertical scale               */
        unsigned char   ucWeight;               /* thickness                    */
        unsigned char   ucStyle;                /* shape                        */
}GSimpleFont;

typedef struct _GFont{
        short           nID;                    /* font ID                      */
        GTCHAR          szFontName[MAX_FONTNAME_LEN+1];   /* font name          */
        unsigned short  usCharacterSet;         /* character code               */
        short           nFontSizeCount;         /* number of font sizes         */
        GFontSize       *gfsSize;               /* pointer to font size array   */
        short           nFixedWidth;            /* fixed width font information */
        long            lFontSupport;         /* forms supported by the font (italics, bold, etc.) */
}GFont;

typedef struct _GFontAttribute{
        unsigned short  usWidth;                /* character width              */
        unsigned short  usHeight;               /* character height             */
        unsigned short  usWeight;               /* character thickness          */
        unsigned short  usItalic;               /* character italics            */
        unsigned short  usOutline;              /* character border             */
        void            *pData;                 /* additional information       */
        unsigned char   ucXScale;               /* horizontal scale             */
        unsigned char   ucYScale;               /* vertical scale               */
}GFontAttribute;
```

```
typedef struct _GFontPattern{
        GTCHAR          *pcChar;            /* pointer to character data            */
        GTCHAR          *pcNextChar;        /* pointer to next character data       */
        short           nWidth;             /* width                                */
        short           nHeight;            /* height                               */
        short           nBpp;               /* number of dots per pixel             */
        char            *pcPattern;         /* pointer to pattern                   */
        short           nGetSize;           /* get size flag                        */
}GFontPattern;

typedef struct _GFontHandle{
        short           nID;                /* font ID                              */
        short           nAttributeType;     /* use either pgfaAttribute or          */
                                            /* pnAttributeArray as attribute        */
        GFontAttribute  *pgfaAttribute;     /* font attribute                       */
        short           *pnAttributeArray;  /* font attribute (array)               */
}GFontHandle;
```

```
/*****************************************************************************
 * Image data structure
 *****************************************************************************/
typedef struct _GRFHeader{
        short           nType;          /* image type                       */
        short           nWidth;         /* width                            */
        short           nHeight;        /* height                           */
        short           nBpp;           /* number of bits per pixel         */
        long            lSize;          /* data size                        */
        unsigned char   *pData;         /* pointer to real data             */
}GRFHeader;
typedef struct _GRFHeaderDIB{
        short           nType;          /* image type                       */
        short           nWidth;         /* width                            */
        short           nHeight;        /* height                           */
        short           nBpp;           /* number of bits per pixel         */
        long            lSize;          /* data size                        */
        unsigned char   *pData;         /* pointer to real data             */
        unsigned char   *pPalette;      /* pointer to palette data          */
}GRFHeaderDIB;

/* image structure */
typedef struct _GImage{
        GRFHeader       *pImage;        /* pointer to image data structure  */
}GImage;


/*****************************************************************************
 * System time structure
 *****************************************************************************/
typedef struct _GSystemTime{
        unsigned long   ulLTime;        /* lower-side 32 bits for system time */
        unsigned long   ulUTime;        /* upper-side 32 bits for system time */
}GSystemTime;
```

```
/*****************************************************************************
 * Drawing environment structure
 *****************************************************************************/
/* GDraw structure */
typedef struct _GDraw{
        GRect               grLocalRect;            /* physical coordinate area                          */
        GPoint              gptLocalOrigin;         /* position of origin on physical coordinates        */
        GRect               grVirtualRect;          /* virtual coordinate area                           */
        GRect               grClipRect;             /* clipping rectangle                                */
        GPoint              gptPenPosition;         /* current position                                  */
        GColor              gcPenColor;             /* line color                                        */
        GPoint              gptPenSize;             /* line thickness                                    */
        short               nPenCap;                /* line end shape                                    */
        short               nPenJoin;               /* line contact shape                                */
        short               nPenDash;               /* line type No.                                     */
        short               nDashOffset;            /* line pattern offset                               */
        short               nTextMode;              /* text mode                                         */
        GColor              gcForeColor;            /* fill foreground color                             */
        GColor              gcBackColor;            /* fill background color                             */
        short               nFillPattern;           /* fill pattern No.                                  */
        short               nDrawCondition;         /* drawing condition                                 */
        short               nDrawingMode;           /* raster operation                                  */
        HGVRAM              hSystemVram;            /* display destination VRAM                          */
        HGVRAM              hDrawVram;              /* drawing destination VRAM                          */
        HGFONT              hFont;                  /* font                                              */
        void                *vgdftDraw;             /* function table for drawing                        */
        char                cDashPatterns[NUM_DASH_PATS][DASH_PAT_SIZE];
                                                    /* line pattern                                      */
        char                cFillPatterns[NUM_FILL_PATS][FILL_PAT_SIZE];
                                                    /* fill pattern                                      */
        GDrawParamPoly      gpPoly;                 /* polygon drawing parameter                         */
        GDrawParamWideLine  gpwWideLine;            /* wide line drawing parameter                       */
        GDrawParamOval      gpwOval;                /* circle, arc and sector drawing parameter          */
        unsigned short      usErrorCode;            /* previous error code                               */
        unsigned char       ucGradationType;        /* gradation type                                    */
                                                    /* (0: up to down, 1: left to right)                 */
        GColor              gcGradationColor1;      /* color1                                            */
        GColor              gcGradationColor2;      /* color2                                            */
        unsigned short      usVertexPos;            /* gradation vertex position (0 to 100)              */
        unsigned short      usGradationLevel;       /* gradation level (0 to 256)                        */
        unsigned char       ucColorMode;            /* actual VRAM color
                                                       environment information                        */
        GColor      gcRedMask;                      /* direct color R value mask                         */
        GColor      gcGreenMask;                    /* direct color G value mask                         */
        GColor      gcBlueMask;                     /* direct color B value mask                         */
        char        cRedShift; /                    * direct color R value shift value                  */
        char        cGreenShift;                    /* direct color G value shift value                  */
        char        cBlueShift;                     /* direct color B value shift value                  */
        GColor      gcForeColorOrg;                 /* foreground color (original)                       */
        GColor      gcBackColorOrg;                 /* background color (original)                       */
        GColor      gcPenColorOrg;                  /* background color (original)                       */

        unsigned long ulExParam;                    /* extension parameter                               */
        short       nExParam;                       /* extension parameter                               */
}GDraw;
```

```
/*****************************************************************************
 * Memory management structure
 *****************************************************************************/
typedef struct _GMemory{
        unsigned char       fUseSpace;       /* shows use (0)/not use (1) status for memory space */
        unsigned char       cReserve[3];    /* reserved space (for 4 byte environment adjustment) */
        unsigned long       ulSize;                 /* memory space size                          */
        struct _GMemory     *pvPrevMemorySpace;
                                            /* pointer to GMemory in previous memory space */
        struct _GMemory     *pvSmallMemorySpace; /* pointer to space GMemory in small          */
                                            /* memory space                                   */
        struct _GMemory     *pvLargeMemorySpace;/* pointer to space GMemory in large          */
                                            /* memory space                                   */
}GMemory;

typedef struct _GMemorySpaceInformation{
        unsigned char       ucType;                 /* memory space type                      */
        unsigned char       ucPlane;                /* memory space plane No.                 */
        unsigned char       cReserve[2];    /* reserved space (for 4 byte environment adjustment) */
        char                *pvMemorySpace;         /* pointer to memory space allocated by user */
        unsigned long       ulMemorySpaceSize;
                                    /* size of memory space allocated by user (multiples of 32) */
        GMemory             *pgmNoUseMemoryTree;
                                        /* pointer to unused two-branch memory management */
}GMemorySpaceInformation;
```

```
/*****************************************************************************
 * Control related structures
 *****************************************************************************/
/* Design structure */
typedef struct GDesign{
        GBrush          gbBrush;        /* fill brush                   */
        unsigned short  usImageID;      /* image resource ID            */
}GDesign;

/* Focus movement structure */
typedef struct GFocusObject{
        unsigned short  usKeyCode;      /* virtual key code             */
        unsigned short  usType;         /* focus movement method        */
        unsigned short  usID;           /* ID of object being moved     */
}GFocusObject;

/* Focus movement structure */
typedef struct GFocusInformation{
        unsigned short  usCount;        /* number of focus settings     */
        GFocusObject    *pFocusArray;   /* focus setting (array)        */
}GFocusInformation;

/* Value structure */
typedef union GValue{
        short           nValue;         /* short value                  */
        unsigned short  usValue;        /* unsigned short value         */
        long            lValue;         /* long value                   */
        unsigned long   ulValue;        /* unsigned long value          */
        float           fValue;         /* float value                  */
}GValue;
```

```
/* GBaseWindow Export/Import structure */
typedef struct GBaseWindowProperty{
        unsigned short      usType;          /* object type                */
        unsigned short      usID;            /* object ID                  */
        short               nX;              /* X coordinate               */
        short               nY;              /* Y coordinate               */
        short               nWidth;          /* width                      */
        short               nHeight;         /* height                     */
        unsigned long       ulStyle;         /* object shape               */
        GFocusInformation   *pFocusInfo;     /* focus object               */
}GBaseWindowProperty;
```

Index

GC

GCSA

GCSAddBlinkControl 198
GCSAddChangePageLog..... 232
GCSAddChild 189
GCSAddDirtyRect..... 172
GCSAddOperationLog..... 231
GCSAddPageLog 228
GCSAddRefreshRect..... 155
GCSAddScrollControl..... 199
GCSAddTimerObject..... 218
GCSAddWindow..... 208
GCSAttachPanel 200

GCSB

GCSBeginModal..... 185
GCSButtonGetAction..... 260
GCSButtonGetBorderID 263
GCSButtonGetCaption 270
GCSButtonGetDisableDesign..... 267
GCSButtonGetDisplay..... 262
GCSButtonGetFocusDesign..... 266
GCSButtonGetFocusEffect..... 271
GCSButtonGetFontID..... 268
GCSButtonGetOffDesign..... 265
GCSButtonGetOnDesign..... 264
GCSButtonGetStatus..... 272
GCSButtonGetStringID..... 269
GCSButtonSetAction 259
GCSButtonSetBorderID..... 262
GCSButtonSetCaption..... 269
GCSButtonSetDisableDesign..... 266
GCSButtonSetDisplay 261
GCSButtonSetFocusDesign 265
GCSButtonSetFocusEffect 270
GCSButtonSetFontID 267
GCSButtonSetOffDesign 264
GCSButtonSetOnDesign 263
GCSButtonSetStatus.....271
GCSButtonSetStringID.....268
GCSBWLAdd 394
GCSBWLClear 395
GCSBWLCount 400
GCSBWLDelete 402
GCSBWLFindFirstItem.....398
GCSBWLFindLastItem 399
GCSBWLFindListItem 398
GCSBWLFindNextItem 399
GCSBWLFindPrevItem 400
GCSBWLGetFirst 397
GCSBWLGetItemPosition 401
GCSBWLGetLast 397
GCSBWLInit 394
GCSBWLMoveFirst 396
GCSBWLMoveLast 396
GCSBWLNew.....402
GCSBWLRemove 395
GCSBWLSetItemPosition.....401

GCSC

GCSCalcClientRect.....140
GCSCallBackMessage.....249
GCSChangeActiveFocus 417
GCSChangeFocus 208
GCSChar.....167
GCSCheckboxGetBorderID.....273
GCSCheckboxGetBoxColor 281
GCSCheckboxGetBoxSize.....280
GCSCheckboxGetCaption 276
GCSCheckboxGetDisableBoxColor 284
GCSCheckboxGetDisableCaptionColor 283
GCSCheckboxGetFocusColor.....282
GCSCheckboxGetFocusEffect.....277
GCSCheckboxGetFontID 274
GCSCheckboxGetStatus.....279
GCSCheckboxGetStringID.....275
GCSCheckboxSetBorderID 272
GCSCheckboxSetBoxColor.....280

GCSCheckboxSetBoxSize 279
GCSCheckboxSetCaption 275
GCSCheckboxSetDisableBoxColor 283
GCSCheckboxSetDisableCaptionColor 282
GCSCheckboxSetFocusColor 281
GCSCheckboxSetFocusEffect 276
GCSCheckboxSetFontID 273
GCSCheckboxSetStatus 278
GCSCheckboxSetStringID 274
GCSClearDrawStatus 171
GCSClearOperationLog 231
GCSClearPageLog 227
GCSClientToScreen 141
GCSClose 167
GCSCreate 151
GCSCreateChildren 191
GCSCreateGWindow 226
GCSCreateInstance 221
GCSCreateVram 152

GCSD

GCSDelete 152
GCSDeleteAllChildren 192
GCSDeleteChild 168
GCSDispatchMessage 209
GCSDoModal 215
GCSDrawBackGround 225
GCSDrawChildren 200
GCSDrawFrame 222
GCSDrawPanel 223

GCSE

GCSEditAddLine 291
GCSEditAddString 286
GCSEditDeleteLine 292
GCSEditDeleteString 290
GCSEditFind 301
GCSEditGetBackColor 289
GCSEditGetBorderID 285
GCSEditGetBrush 306
GCSEditGetCursor 294
GCSEditGetFillPattern 296
GCSEditGetFontID 286
GCSEditGetForeColor 295
GCSEditGetInsertMode 297
GCSEditGetLength 300
GCSEditGetLineBrush 298
GCSEditGetLineCount 305
GCSEditGetLineFeedCode 293
GCSEditGetLineString 292
GCSEditGetLineTextColor 299
GCSEditGetString 287
GCSEditGetTextColor 288
GCSEditInsertLine 291
GCSEditInsertString 305
GCSEditRemoveString 306
GCSEditReplace 303
GCSEditReplaceAll 304
GCSEditReverseFind 302
GCSEditSetBackColor 289
GCSEditSetBorderID 284
GCSEditSetBrush 290
GCSEditSetCursor 294
GCSEditSetFillPattern 296
GCSEditSetFontID 285
GCSEditSetForeColor 295
GCSEditSetInsertMode 297
GCSEditSetLineBrush 298
GCSEditSetLineFeedCode 293
GCSEditSetLineTextColor 299
GCSEditSetString 287
GCSEditSetTextColor 288
GCSEndModal 186
GCSExportProperty 151

GCSF

GCSFreeLoadData 183

GCSG

GCSGetActiveChild 195
GCSGetActiveControl 195
GCSGetActiveWindow 220
GCSGetAngle 253
GCSGetBackGroundDesign 191
GCSGetBlinkEnableStatus 237
GCSGetBlinkOffInterval 196

GCSGetBlinkOnInterval 197
GCSGetBlinkStatus 175
GCSGetBlinkType 236
GCSGetBounds 140
GCSGetBrush 254
GCSGetCaptionScrollDelayTime 243
GCSGetCaptionScrollEnable 239
GCSGetCaptionScrollMovementValue 245
GCSGetCaptionScrollPosition 247
GCSGetCaptionScrollRefreshTime 244
GCSGetCaptionScrollStartPosition 246
GCSGetCaptionScrollStatus 242
GCSGetCaptionStatus 146
GCSGetChild 192
GCSGetChildAll 193
GCSGetChildList 190
GCSGetClientRect 141
GCSGetCloseButtonStatus 188
GCSGetConsumableStatus 145
GCSGetDrawForceStatus 170
GCSGetEdgeStatus 189
GCSGetEnableStatus 144
GCSGetEndPoint 256
GCSGetExitStatus 213
GCSGetFocus 209
GCSGetFocusInformation 166
GCSGetFocusStatus 147
GCSGetFrame 207
GCSGetGBaseObject 183
GCSGetHeight 138
GCSGetHWDC 210
GCSGetID 135
GCSGetMemorySpace 169
GCSGetModalStatus 184
GCSGetNextBlinkControl 178
GCSGetNextScrollControl 180
GCSGetNextTimerObject 174
GCSGetOperationLog 230
GCSGetPageNumber 232
GCSGetPageOffset 230
GCSGetPaletteColors 218
GCSGetPaletteStatus 216
GCSGetPanel 201
GCSGetPanelPlane 211
GCSGetPanelStatus 148
GCSGetParent 150
GCSGetParentPanel 176
GCSGetPenColor 250
GCSGetPenDash 251
GCSGetPenSize 252
GCSGetPoly 257
GCSGetPosition 137
GCSGetPrevBlinkControl 177
GCSGetPrevScrollControl 179
GCSGetPrevTimerObject 173
GCSGetRedrawStatus 171
GCSGetRetValue 185
GCSGetScreen 182
GCSGetSize 139
GCSGetStartPoint 255
GCSGetStyle 142
GCSGetStyleStatus 149
GCSGetSystemActiveColor 206
GCSGetSystemBackColor 205
GCSGetSystemBorderID 203
GCSGetSystemFontID 202
GCSGetSystemForeColor 204
GCSGetSystemInactiveColor 207
GCSGetSystemPaletteID 217
GCSGetSystemStatus 214
GCSGetTitleBarStatus 187
GCSGetType 135
GCSGetVisibleStatus 143
GCSGetWDC 210
GCSGetWidth 138
GCSGetWindow 219
GCSGetWindowPlane 212
GCSGetXPosition 136
GCSGetYPosition 136

GCSH

GCSHtmlbrowserGetBackColor 311
GCSHtmlbrowserGetBorderID 307
GCSHtmlbrowserGetFontID 308
GCSHtmlbrowserGetHtmlFileName 313
GCSHtmlbrowserGetLinkColor 312
GCSHtmlbrowserGetScrollBarWidth 309
GCSHtmlbrowserGetTextColor 310
GCSHtmlbrowserSetBackColor 311
GCSHtmlbrowserSetBorderID 307
GCSHtmlbrowserSetFontID 308
GCSHtmlbrowserSetHtmlFileName 313

GCSHtmlbrowserSetLinkColor 312
GCSHtmlbrowserSetScrollBarWidth 309
GCSHtmlbrowserSetTextColor 310

GCSI

GCSImportProperty 150
GCSIsCaptionOutOfBounds 242
GCSIsCaptionScrollFinish 248

GCSK

GCSKeyPress 162
GCSKeyRelease 163
GCSKillFocus 164

GCSL

GCSLabelGetCaption 316
GCSLabelGetFontID 314
GCSLabelGetStringID 315
GCSLabelSetCaption 316
GCSLabelSetFontID 314
GCSLabelSetStringID 315
GCSLButtonPress 156
GCSLButtonRelease 159
GCSListAddString 328
GCSListGetBorderID 317
GCSListGetCaption 319
GCSListGetCurrentSelect 330
GCSListGetDisableColor 324
GCSListGetEnableScrollBarStatus 332
GCSListGetFocusColor 320
GCSListGetFontID 318
GCSListGetLineCount 334
GCSListGetListString 328
GCSListGetMaxListLines 327
GCSListGetNormalColor 323
GCSListGetScrollBarColor 322
GCSListGetScrollBarWidth 325
GCSListGetScrollButtonColor 321
GCSListGetSelectBarColor 326
GCSListGetTopLine 333
GCSListGetVisibleScrollBarStatus 331
GCSListInsertString 333
GCSListRemoveAllStrings 329
GCSListRemoveString 334
GCSListSetBorderID 317
GCSListSetCaption 319
GCSListSetCurrentSelect 329
GCSListSetDisableColor 324
GCSListSetEnableScrollBarStatus 331
GCSListSetFocusColor 320
GCSListSetFontID 318
GCSListSetMaxListLines 327
GCSListSetNormalColor 323
GCSListSetScrollBarColor 322
GCSListSetScrollBarWidth 325
GCSListSetScrollButtonColor 321
GCSListSetSelectBarColor 326
GCSListSetTopLine 332
GCSListSetVisibleScrollBarStatus 330
GCSLoadData 181
GCSLoadGWindow 234
GCSLoadProjectFile 233
GCSLoadScreenFile 249

GCSM

GCSModalGWindow 234
GCSMouseMove 160

GCSN

GCSNCPLCButtonGetAction 418
GCSNCPLCButtonGetBorderID 421
GCSNCPLCButtonGetCaption 433
GCSNCPLCButtonGetDevice 437
GCSNCPLCButtonGetDeviceType 436
GCSNCPLCButtonGetDeviceVal 438
GCSNCPLCButtonGetDisableDesign 427
GCSNCPLCButtonGetDisplay 420
GCSNCPLCButtonGetFocusDesign 426
GCSNCPLCButtonGetFocusEffect 434
GCSNCPLCButtonGetFontID 428
GCSNCPLCButtonGetOffDesign 423
GCSNCPLCButtonGetOffStringID 430
GCSNCPLCButtonGetOnDesign 422
GCSNCPLCButtonGetOnOffDesign 425

GCSNCPLCButtonGetOnOffStringID 432
GCSNCPLCButtonGetOnOnDesign 424
GCSNCPLCButtonGetOnOnStringID 431
GCSNCPLCButtonGetOnStringID 429
GCSNCPLCButtonGetStatus 435
GCSNCPLCButtonSetAction 417
GCSNCPLCButtonSetBorderID 420
GCSNCPLCButtonSetCaption 432
GCSNCPLCButtonSetDevice 436
GCSNCPLCButtonSetDeviceType 435
GCSNCPLCButtonSetDeviceVal 437
GCSNCPLCButtonSetDisableDesign 426
GCSNCPLCButtonSetDisplay 419
GCSNCPLCButtonSetFocusDesign 425
GCSNCPLCButtonSetFocusEffect 433
GCSNCPLCButtonSetFontID 427
GCSNCPLCButtonSetOffDesign 422
GCSNCPLCButtonSetOffStringID 429
GCSNCPLCButtonSetOnDesign 421
GCSNCPLCButtonSetOnOffDesign 424
GCSNCPLCButtonSetOnOffStringID 431
GCSNCPLCButtonSetOnOnDesign 423
GCSNCPLCButtonSetOnOnStringID 430
GCSNCPLCButtonSetOnStringID 428
GCSNCPLCButtonSetStatus 434
GCSNCPLCTextboxGetBackGroundPattern 458
GCSNCPLCTextboxGetBorderID 439
GCSNCPLCTextboxGetCaption 441
GCSNCPLCTextboxGetCheckMaxStatus 449
GCSNCPLCTextboxGetCheckMinStatus 450
GCSNCPLCTextboxGetCommaStatus 448
GCSNCPLCTextboxGetDecimalWidth 462
GCSNCPLCTextboxGetDevice 465
GCSNCPLCTextboxGetDisableColor 445
GCSNCPLCTextboxGetFocusColor 443
GCSNCPLCTextboxGetFocusEffect 442
GCSNCPLCTextboxGetFontID 440
GCSNCPLCTextboxGetGValue 455
GCSNCPLCTextboxGetIntegerWidth 461
GCSNCPLCTextboxGetMaxGValue 451
GCSNCPLCTextboxGetMinGValue 452
GCSNCPLCTextboxGetNormalColor 444
GCSNCPLCTextboxGetOffset 464
GCSNCPLCTextboxGetPasswordStatus 447
GCSNCPLCTextboxGetScale 463
GCSNCPLCTextboxGetSelectStringEndPos 460
GCSNCPLCTextboxGetSelectStringStartPos 459
GCSNCPLCTextboxGetString 453
GCSNCPLCTextboxGetStringBuffer 457
GCSNCPLCTextboxGetTextSize 466
GCSNCPLCTextboxGetTextType 446
GCSNCPLCTextboxGetZeroSuppressStatus 467
GCSNCPLCTextboxSetBackGroundPattern 457
GCSNCPLCTextboxSetBorderID 438
GCSNCPLCTextboxSetCaption 440
GCSNCPLCTextboxSetCheckMaxStatus 448
GCSNCPLCTextboxSetCheckMinStatus 449
GCSNCPLCTextboxSetCommaStatus 447
GCSNCPLCTextboxSetDecimalWidth 461
GCSNCPLCTextboxSetDevice 464
GCSNCPLCTextboxSetDisableColor 444
GCSNCPLCTextboxSetFocusColor 442
GCSNCPLCTextboxSetFocusEffect 441
GCSNCPLCTextboxSetFontID 439
GCSNCPLCTextboxSetGValue 454
GCSNCPLCTextboxSetIntegerWidth 460
GCSNCPLCTextboxSetMaxGValue 450
GCSNCPLCTextboxSetMinGValue 451
GCSNCPLCTextboxSetNormalColor 443
GCSNCPLCTextboxSetOffset 463
GCSNCPLCTextboxSetPasswordStatus 446
GCSNCPLCTextboxSetScale 462
GCSNCPLCTextboxSetSelectStringEndPos 459
GCSNCPLCTextboxSetSelectStringStartPos 458
GCSNCPLCTextboxSetString 452
GCSNCPLCTextboxSetStringBuffer 456
GCSNCPLCTextboxSetTextSize 465
GCSNCPLCTextboxSetTextType 445
GCSNCPLCTextboxSetZeroSuppressStatus 466
GCSNCTableGetBeforeSubCursorNum 485
GCSNCTableGetCellNumFromPoint 476
GCSNCTableGetCellString 471
GCSNCTableGetLineTitleString 478
GCSNCTableGetRowTitleString 480
GCSNCTableGetSubCursorNum 473
GCSNCTableSetBeforeSubCursorNum 484
GCSNCTableSetCellBackColor 469
GCSNCTableSetCellForeColor 468
GCSNCTableSetCellHight 475
GCSNCTableSetCellString 470
GCSNCTableSetCellStringPosition 483
GCSNCTableSetCellWidth 474
GCSNCTableSetLineTitleBackColor 487
GCSNCTableSetLineTitleForeColor 486

GCSNCTableSetLineTitleString 477
GCSNCTableSetLineTitleStringPosition..... 481
GCSNCTableSetRowTitleBackColor..... 489
GCSNCTableSetRowTitleForeColor..... 488
GCSNCTableSetRowTitleString 479
GCSNCTableSetRowTitleStringPosition 482
GCSNCTableSetSubCursorBackColor..... 491
GCSNCTableSetSubCursorForeColor 490
GCSNCTableSetSubCursorNum..... 472
GCSNextPage 229

GCSO

GCSOnDraw..... 153
GCSOnDrawWindow..... 224
GCSOnLButtonPress..... 157
GCSOnLButtonRelease..... 159
GCSOnMouseMove..... 161
GCSOnNcDraw 154
GCSOnNcLButtonPress 158
GCSOnNcLButtonRelease 160
GCSOnNcMouseMove 161

GCSP

GCSPageLogOperation..... 222
GCSPaint..... 153
GCSPauseCaptionScroll 240
GCSPictureGetBorderID..... 410
GCSPictureGetCaption..... 413
GCSPictureGetDisplay 409
GCSPictureGetFontID 411
GCSPictureGetMaxStatus 415
GCSPictureGetStatus..... 414
GCSPictureGetStatusDesign..... 416
GCSPictureGetStringID 412
GCSPictureSetBorderID 410
GCSPictureSetCaption 413
GCSPictureSetDisplay..... 408
GCSPictureSetFontID..... 411
GCSPictureSetMaxStatus 415
GCSPictureSetStatus 414
GCSPictureSetStatusDesign 416
GCSPictureSetStringID 412
GCSPrevPage 228

GCSR

GCSRadiobuttonGetBorderID 341
GCSRadiobuttonGetBoxColor.....348
GCSRadiobuttonGetBoxSize 347
GCSRadiobuttonGetCaption 344
GCSRadiobuttonGetDisableBoxColor.....351
GCSRadiobuttonGetDisableCaptionColor.....350
GCSRadiobuttonGetFocusColor 349
GCSRadiobuttonGetFocusEffect.....345
GCSRadiobuttonGetFontID.....342
GCSRadiobuttonGetRadioGroup 352
GCSRadiobuttonGetStatus 346
GCSRadiobuttonGetStringID.....343
GCSRadiobuttonSetBorderID.....340
GCSRadiobuttonSetBoxColor 347
GCSRadiobuttonSetBoxSize.....346
GCSRadiobuttonSetCaption.....343
GCSRadiobuttonSetDisableBoxColor 350
GCSRadiobuttonSetDisableCaptionColor 349
GCSRadiobuttonSetFocusColor.....348
GCSRadiobuttonSetFocusEffect 344
GCSRadiobuttonSetFontID 341
GCSRadiobuttonSetRadioGroup.....351
GCSRadiobuttonSetStatus.....345
GCSRadiobuttonSetStringID 342
GCSRemoveBlinkControl.....198
GCSRemoveScrollControl.....199
GCSRemoveTimerObject.....219
GCSRestartCaptionScroll.....241

GCSS

GCSScrollbarexGetBackGroundColor 364

GCSScrollbarexGetBarEnableStatus 368
GCSScrollbarexGetDirection 355
GCSScrollbarexGetDisplay 353
GCSScrollbarexGetNextBtnOffImgID 363
GCSScrollbarexGetNextBtnOnImgID 362
GCSScrollbarexGetOutLineBorderID 365
GCSScrollbarexGetPageSize 356
GCSScrollbarexGetPinchColor 367
GCSScrollbarexGetPinchWidth 366
GCSScrollbarexGetPrevBtnOffImgID 361
GCSScrollbarexGetPrevBtnOnImgID 360
GCSScrollbarexGetRange 357
GCSScrollbarexGetScrollArrowColor 358
GCSScrollbarexGetScrollPosition 359
GCSScrollbarexSetBackGroundColor 363
GCSScrollbarexSetBarEnableStatus 367
GCSScrollbarexSetDirection 354
GCSScrollbarexSetDisplay 352
GCSScrollbarexSetNextBtnOffImgID 362
GCSScrollbarexSetNextBtnOnImgID 361
GCSScrollbarexSetOutLineBorderID 364
GCSScrollbarexSetPageSize 355
GCSScrollbarexSetPinchColor 366
GCSScrollbarexSetPinchWidth 365
GCSScrollbarexSetPrevBtnOffImgID 360
GCSScrollbarexSetPrevBtnOnImgID 359
GCSScrollbarexSetRange 356
GCSScrollbarexSetScrollArrowColor 357
GCSScrollbarexSetScrollPosition 358
GCSSetActiveChild 194
GCSSetAngle 253
GCSSetBackGroundDesign 190
GCSSetBlinkEnableStatus 236
GCSSetBlinkOffInterval 196
GCSSetBlinkOnInterval 197
GCSSetBlinkStatus 175
GCSSetBlinkType 235
GCSSetBounds 139
GCSSetBrush 254
GCSSetCaptionScrollDelayTime 243
GCSSetCaptionScrollEnable 238
GCSSetCaptionScrollMovementValue 245
GCSSetCaptionScrollPosition 247
GCSSetCaptionScrollRefreshTime 244
GCSSetCaptionScrollStartPosition 246
GCSSetCaptionScrollStatus 241
GCSSetCaptionStatus 145
GCSSetCloseButtonStatus 187
GCSSetConsumableStatus 144
GCSSetDrawAttribute 258
GCSSetDrawForceStatus 169
GCSSetEdgeStatus 188
GCSSetEnableStatus 143
GCSSetEndPoint 256
GCSSetExitStatus 213
GCSSetFocus 164
GCSSetFocusInformation 166
GCSSetFocusStatus 146
GCSSetGBaseObject 182
GCSSetHWDCEnvironment 181
GCSSetMemorySpace 168
GCSSetModalStatus 184
GCSSetModuleID 227
GCSSetNextBlinkControl 177
GCSSetNextScrollControl 179
GCSSetNextTimerObject 174
GCSSetPageOffset 229
GCSSetPaletteColors 217
GCSSetPaletteStatus 215
GCSSetPanelPlane 211
GCSSetPanelStatus 147
GCSSetParent 149
GCSSetPenColor 250
GCSSetPenDash 251
GCSSetPenSize 252
GCSSetPoly 257
GCSSetPosition 137
GCSSetPrevBlinkControl 176
GCSSetPrevScrollControl 178
GCSSetPrevTimerObject 173
GCSSetRedrawStatus 170
GCSSetScreenManager 226
GCSSetStartPoint 255
GCSSetStyleStatus 148
GCSSetSystemActiveColor 205
GCSSetSystemBackColor 204
GCSSetSystemBorderID 202
GCSSetSystemFontID 201
GCSSetSystemForeColor 203
GCSSetSystemInactiveColor 206
GCSSetSystemPalette 225
GCSSetSystemPaletteID 216
GCSSetSystemStatus 214
GCSSetTitleBarStatus 186

GCSSetVisibleStatus 142
GCSSetWindowPlane 212
GCSShowPanel 220
GCSStartCaptionBlink 237
GCSStartCaptionScroll 239
GCSStopCaptionBlink 238
GCSStopCaptionScroll 240

GCST

GCSTextboxGetBackGroundPattern 393
GCSTextboxGetBorderID 369
GCSTextboxGetCaption 371
GCSTextboxGetCheckMaxStatus 385
GCSTextboxGetCheckMinStatus 386
GCSTextboxGetCommaStatus 379
GCSTextboxGetDisableColor 375
GCSTextboxGetFocusColor 373
GCSTextboxGetFocusEffect 372
GCSTextboxGetFontID 370
GCSTextboxGetFormatID 377
GCSTextboxGetGValue 391
GCSTextboxGetMaxGValue 387
GCSTextboxGetMinGValue 388
GCSTextboxGetNormalColor 374
GCSTextboxGetPasswordStatus 378
GCSTextboxGetRefuseInputCapitalLetterStatus 382
GCSTextboxGetRefuseInputFullLetterStatus 384
GCSTextboxGetRefuseInputNumberStatus 380
GCSTextboxGetRefuseInputSmallLetterStatus 381
GCSTextboxGetRefuseInputSymbolLetterStatus 383
GCSTextboxGetString 389
GCSTextboxGetStringBuffer 392
GCSTextboxGetTextType 376
GCSTextboxSetBackGroundPattern 393
GCSTextboxSetBorderID 368
GCSTextboxSetCaption 370
GCSTextboxSetCheckMaxStatus 384
GCSTextboxSetCheckMinStatus 385
GCSTextboxSetCommaStatus 378
GCSTextboxSetDisableColor 374
GCSTextboxSetFocusColor 372
GCSTextboxSetFocusEffect 371
GCSTextboxSetFontID 369
GCSTextboxSetFormatID 376
GCSTextboxSetGValue 390
GCSTextboxSetMaxGValue 386
GCSTextboxSetMinGValue 387
GCSTextboxSetNormalColor 373
GCSTextboxSetPasswordStatus 377
GCSTextboxSetRefuseInputCapitalLetterStatus 381
GCSTextboxSetRefuseInputFullLetterStatus 383
GCSTextboxSetRefuseInputNumberStatus 379
GCSTextboxSetRefuseInputSmallLetterStatus 380
GCSTextboxSetRefuseInputSymbolLetterStatus 382
GCSTextboxSetString 388
GCSTextboxSetStringBuffer 392
GCSTextboxSetTextType 375
GCSTimer 165

GCSU

GCSUpdateScroll 248
GCSUser 165

GCSW

GCSWDCBeginDraw 405
GCSWDCDelete 407
GCSWDCEndDraw 406
GCSWDCGetClipRect 404
GCSWDCGetHGDRAW 403
GCSWDCInceptClipRect 405
GCSWDCNew 406
GCSWDCOffsetOrigin 403
GCSWDCSetClipRect 404
GCSWDCSetOrigin 407

GD

GDB

GDBeginDraw 8
GDBitBlt 9

GDC

GDCalcBorderRect 10
GDCloseDrawHandle 10

GDCreateVram 88

GDD

GDDefineFillPattern 11
GDDefinePenDash 12
GDDeleteVram 88
GDDrawArc 13
GDDrawBezier 14
GDDrawCaptionCursorString 15
GDDrawCaptionString 16
GDDrawCaptionStringResource 17
GDDrawGrf 18
GDDrawImage 19
GDDrawImageResource 20
GDDrawOval 21
GDDrawOvalBorder 22
GDDrawOvalBorderResource 23
GDDrawPie 24
GDDrawPoly 25
GDDrawRect 25
GDDrawRectBorder 26
GDDrawRectBorderResource 27
GDDrawRRect 28
GDDrawString 29
GDDrawStringResource 29

GDE

GDEndDraw 30

GDF

GDFlushScreen 30
GDFlushScreenForce 89

GDG

GDGetCharHeight 31
GDGetCharWidth 31
GDGetGradationInfo 90
GDGetLocalOrigin 90
GDGetNextChar 32
GDGetPalette 33
GDGetPixel 34
GDGetPrevChar 35
GDGetStringLength 35
GDGetStringWidth 36
GDGetVram 91

GDI

GDInceptClipRect 37
GDInitPlatform 37

GDL

GDLineTo 38

GDM

GDMoveTo 38

GDO

GDOffsetLocalOrigin 39
GDOpenDrawHandle 39

GDQ

GDQueryError 40
GDQuitPlatform 40

GDR

GDRemoveDrawableRegion 91

GDS

GDSetBackColor 41
GDSetBrush 41
GDSetClipRect 42
GDSetDisplay 42
GDSetDrawableRegion 92

GDSetDrawingMode 43
GDSetFillAttribute 44
GDSetFillColor 45
GDSetFillPattern 46
GDSetFont 47
GDSetForeColor 47
GDSetGradationInfo 92
GDSetLocalMode 48
GDSetLocalOrigin 93
GDSetPenColor 48
GDSetPenDash 49
GDSetPenJoin 50
GDSetPenSize 51
GDSetPixel 51
GDSetSimpleFont 52
GDSetSimpleFontResource 52
GDSetVirtualMode 53
GDSetVirtualRect 53
GDSetVram 54

GE

GEC

GECheckTimer 111
GEClearTargetEvent 113
GEClearTimerEvent 109
GECompareSystemTime 112
GECreateEventMessage 107

GEG

GEGetEvent 109
GEGetSystemTime 112

GEI

GEInitEvent 106

GEK

GEKillTimer 110

GEQ

GEQuitEvent 106

GES

GESetEvent 108
GESetSystemTime 111
GESetTimer 110

GF

GFG

GFGetFontAttribute 54
GFGetFontCount 55
GFGetFontInformation 55
GFGetFontUserAttribute 56
GFGetSize 56

GFL

GFLoadFont 57

GFR

GFReleaseFont 57

GFS

GFSetFontAttribute 58
GFSetFontUserAttribute 58
GFSetSize 59

GI

GIO

GIOClose 493
GIOOpen 493
GIORead 494

GIOSeek 496
GIOWrite 495

GL

GLD

GLDuplicateLine 70

GLG

GLGetLine 71

GLM

GLMapLine 72

GLO

GLOffsetLine 73

GLP

GLPhysicalToVirtualLine 73

GLS

GLSetLine 74

GLV

GLVirtualToPhysicalLine 75

GM

GMD

GMDeleteMemorySpace 85

GMG

GMGMemCreate 498
GMGMemDelete 499
GMGMemGetChar 500
GMGMemGetDouble 502
GMGMemGetLong 501
GMGMemGetShort 499
GMGMemGetString 503
GMGMemSelect 498
GMGMemSetChar 501
GMGMemSetDouble 503
GMGMemSetLong 502
GMGMemSetShort 500
GMGMemSetString 504

GMM

GMMemoryAlloc 86
GMMemoryFree 87

GMS

GMSetMemorySpace 84

GP

GPG

GPGetPoly 80

GPM

GPMapPoly 81

GPO

GPOffsetPoly 82

GPP

GPPhysicalToVirtualPoly 82

GPS

GPSetPoly 83

GPT

GPTDuplicatePoint 75
GPTGetPoint 76
GPTMapPoint 76
GPTMapToScreenPoint 77
GPTMapToVirtualPoint 77
GPTOffsetPoint 78
GPTPhysicalToVirtualPoint 78
GPTSetPoint 79
GPTVirtualToPhysicalPoint 79

GPV

GPVirtualToPhysicalPoly 83

GR

GRC

GRCGetLocale 99
GRCGetResourceHandle 102
GRCLoadBorder 95
GRCLoadFillPattern 99
GRCLoadFont 95
GRCLoadHtmlFileName 100
GRCLoadImage 96
GRCLoadLanguageStr 100
GRCLoadLinePattern 101
GRCLoadLocaleFont 97
GRCLoadLocaleString 97
GRCLoadNowLanguageStr 101
GRCLoadPalette 98
GRCLoadString 96
GRCMultiByteToUnicode 104
GRCSetLocale 98
GRCSetResourceHandle 102
GRCUnicodeToMultiByte 103

GRD

GRDuplicateRect 59

GRE

GREqualRect 60

GRG

GRGetRect 61

GRI

GRInceptRect 62

GRM

GRMapRect 63
GRMapToScreenRect 63
GRMapToVirtualRect 64

GRO

GROffsetRect 64

GRP

GRPhysicalToVirtualRect 65
GRPointInRect 65

GRR

GRResizeRect 66

GRS

GRSetRect 67
GRShiftRect..... 68

GRU

GRUnionRect 69

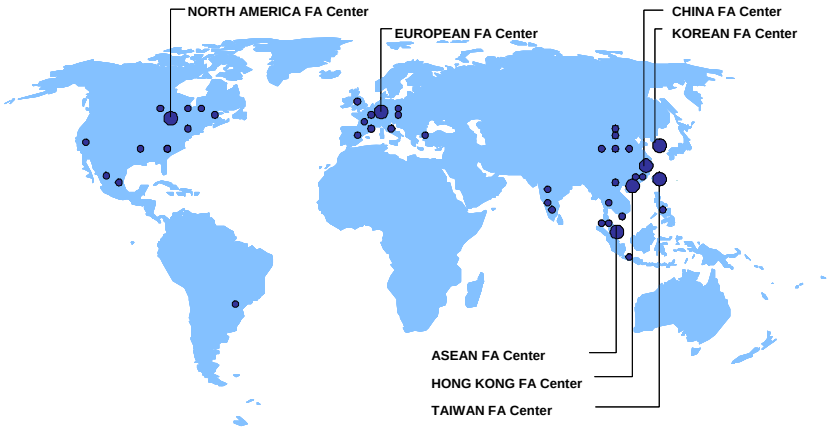
GRV

GRVirtualToPhysicalRect 69

Revision History

Date of revision	Manual No.	Revision details
Nov. 2004	IB(NA)1500109-A	First edition created.
Nov. 2004	IB(NA)1500109-B	The following functions were added to section 5. 4 functions about GCSScrollbar, such as GCSScrollbarexSetRange. 16 functions about GCSPicture, such as GCSPictureSetDisplay. Mistakes were corrected.
Feb. 2007	IB(NA)1500109-C	GCS functions were added. Mistakes were corrected.

Global service network



North America FA Center (MITSUBISHI ELECTRIC AUTOMATION INC.)

Illinois CNC Service Center
500 CORPORATE WOODS PARKWAY, VERNON HILLS, IL. 60061, U.S.A.
TEL: +1-847-478-2500 (Se FAX: +1-847-478-2650 (Se
California CNC Service Center
5665 PLAZA DRIVE, CYPRESS, CA. 90630, U.S.A.
TEL: +1-714-220-4796 FAX: +1-714-229-3818
Georgia CNC Service Center
2810 PREMIERE PARKWAY SUITE 400, DULUTH, GA., 30097, U.S.A.
TEL: +1-678-258-4500 FAX: +1-678-258-4519
New Jersey CNC Service Center
200 COTTONTAIL LANE SOMERSET, NJ. 08873, U.S.A.
TEL: +1-732-560-4500 FAX: +1-732-560-4531
Michigan CNC Service Satellite
2545 38TH STREET, ALLEGAN, MI., 49010, U.S.A.
TEL: +1-847-478-2500 FAX: +1-269-673-4092
Ohio CNC Service Satellite
62 W. 500 S., ANDERSON, IN., 46013, U.S.A.
TEL: +1-847-478-2608 FAX: +1-847-478-2690
Texas CNC Service Satellite
1000, NOLEN DRIVE SUITE 200, GRAPEVINE, TX. 76051, U.S.A.
TEL: +1-817-251-7468 FAX: +1-817-416-1439
Canada CNC Service Center
4299 14TH AVENUE MARKHAM, ON. L3R OJ2, CANADA
TEL: +1-905-475-7728 FAX: +1-905-475-7935
Mexico CNC Service Center
MARIANO ESCOBEDO 69 TLALNEPANTLA, 54030 EDO. DE MEXICO
TEL: +52-55-9171-7662 FAX: +52-55-9171-7698
Monterrey CNC Service Satellite
ARGENTINA 3900, FRACC. LAS TORRES, MONTERREY, N.L., 64720, MEXICO
TEL: +52-81-8365-4171 FAX: +52-81-8365-4171
Brazil MITSUBISHI CNC Agent Service Center
(AUTOMOTION IND. COM. IMP. E EXP. LTDA.)
ACESSO JOSE SARTORELLI, KM 2.1 18550-000 BOITUVA – SP, BRAZIL
TEL: +55-15-3363-9900 FAX: +55-15-3363-9911

European FA Center (MITSUBISHI ELECTRIC EUROPE B.V.)

Germany CNC Service Center
GOTHAER STRASSE 8, 40880 RATINGEN, GERMANY
TEL: +49-2102-486-0 FAX:+49-2102486-591
South Germany CNC Service Center
KURZE STRASSE. 40, 70794 FILDERSTADT-BONLANDEN, GERMANY
TEL: +49-711-3270-010 FAX: +49-711-3270-0141
France CNC Service Center
25, BOULEVARD DES BOUVETS, 92741 NANTERRE CEDEX FRANCE
TEL: +33-1-41-02-83-13 FAX: +33-1-49-01-07-25
Lyon CNC Service Satellite

U.K CNC Service Center
TRAVELLERS LANE, HATFIELD, HERTFORDSHIRE, AL10 8XB, U.K.
TEL: +44-1707-282-846 FAX:-44-1707-278-992
Italy CNC Service Center
ZONA INDUSTRIALE VIA ARCHIMEDE 35 20041 AGRATE BRIANZA, MILANO ITALY
TEL: +39-039-60531-342 FAX: +39-039-6053-206
Spain CNC Service Satellite
CTRA. DE RUBI, 76-80 -APDO.420 08190 SAINT CUGAT DEL VALLES, BARCELONA SPAIN
TEL: +34-935-65-2236 FAX:
Turkey MITSUBISHI CNC Agent Service Center
(GENEL TEKNIK SISTEMLER LTD. STI.)
DARULACEZE CAD. FAMAS IS MERKEZI A BLOCK NO.43 KAT2 80270 OKMEYDANI ISTANBUL, TURKEY
TEL: +90-212-320-1640 FAX: +90-212-320-1649
Poland MITSUBISHI CNC Agent Service Center (MPL Technology Sp. z. o. o)
UL SLICZNA 34, 31-444 KRAKOW, POLAND
TEL: +48-12-632-28-85 FAX:
Wroclaw MITSUBISHI CNC Agent Service Satellite (MPL Technology Sp. z. o. o)
UL KOBIERZYCKA 23, 52-315 WROCLAW, POLAND
TEL: +48-71-333-77-53 FAX: +48-71-333-77-53
Czech MITSUBISHI CNC Agent Service Center
(AUTOCONT CONTROL SYSTEM S.R.O.)
NEMOCNICNI 12, 702 00 OSTRAVA 2 CZECH REPUBLIC
TEL: +420-596-152-426 FAX: +420-596-152-112

ASEAN FA Center (MITSUBISHI ELECTRIC ASIA PTE. LTD.)

Singapore CNC Service Center
307 ALEXANDRA ROAD #05-01/02 MITSUBISHI ELECTRIC BUILDING SINGAPORE 159943
TEL: +65-6473-2308 FAX: +65-6476-7439
Thailand MITSUBISHI CNC Agent Service Center (F. A. TECH CO., LTD)
898/19,20,21,22 S.V. CITY BUILDING OFFICE TOWER 1 FLOOR 12,14 RAMA III RD BANGPONGPANG, YANNAWA, BANGKOK 10120. THAILAND
TEL: +66-2-682-6522 FAX: +66-2-682-6020
Malaysia MITSUBISHI CNC Agent Service Center
(FLEXIBLE AUTOMATION SYSTEM SDN. BHD.)
60, JALAN USJ 10/1B 47620 UEP SUBANG JAYA SELANGOR DARUL EHSAN MALAYSIA
TEL: +60-3-5631-7605 FAX: +60-3-5631-7636
JOHOR MITSUBISHI CNC Agent Service Satellite
(FLEXIBLE AUTOMATION SYSTEM SDN. BHD.)
NO. 16, JALAN SHAHBANDAR 1, TAMAN UNGKU TUN AMINAH, 81300 SKUDAI, JOHOR MALAYSIA
TEL: +60-7-557-8218 FAX: +60-7-557-3404
Indonesia MITSUBISHI CNC Agent Service Center
(PT. AUTOTEKNINDO SUMBER MAKMUR)
WISMA NUSANTARA 14TH FLOOR JL. M.H. THAMRIN 59, JAKARTA 10350 INDONESIA
TEL: +62-21-3917-144 FAX: +62-21-3917-164
India MITSUBISHI CNC Agent Service Center (MESSUNG SALES & SERVICES PVT. LTD.)
B-36FF, PAVANA INDUSTRIAL PREMISES M.I.D.C., BHOASRI PUNE 411026, INDIA
TEL: +91-20-2711-9484 FAX: +91-20-2712-8115
BANGALORE MITSUBISHI CNC Agent Service Satellite
(MESSUNG SALES & SERVICES PVT. LTD.)
S 615, 6TH FLOOR, MANIPAL CENTER, BANGALORE 560001, INDIA
TEL: +91-80-509-2119 FAX: +91-80-532-0480
Delhi MITSUBISHI CNC Agent Parts Center (MESSUNG SALES & SERVICES PVT. LTD.)
1197, SECTOR 15 PART-2, OFF DELHI-JAIPUR HIGHWAY BEHIND 32ND MILESTONE GURGAON 122001, INDIA
TEL: +91-98-1024-8895 FAX:
Philippines MITSUBISHI CNC Agent Service Center
(FLEXIBLE AUTOMATION SYSTEM CORPORATION)
UNIT No.411, ALABAMG CORPORATE CENTER KM 25. WEST SERVICE ROAD SOUTH SUPERHIGHWAY, ALABAMG MUNTINLUPA METRO MANILA, PHILIPPINES 1771
TEL: +63-2-807-2416 FAX: +63-2-807-2417
Vietnam MITSUBISHI CNC Agent Service Center (SA GIANG TECHNO CO., LTD)
47-49 HOANG SA ST. DAKAO WARD, DIST.1 HO CHI MINH CITY, VIETNAM
TEL: +84-8-910-4763 FAX: +84-8-910-2593

China FA Center (MITSUBISHI ELECTRIC AUTOMATION (SHANGHAI) LTD.)

China CNC Service Center
2/F., BLOCK 5 BLDG.AUTOMATION INSTRUMENTATION PLAZA, 103 CAOBAO RD. SHANGHAI 200233, CHINA
TEL: +86-21-6120-0808 FAX: +86-21-6494-0178
Shenyang CNC Service Center
TEL: +86-24-2397-0184 FAX: +86-24-2397-0185
Beijing CNC Service Satellite
9/F, OFFICE TOWER1, HENDERSON CENTER, 18 JIANGUOMENNEI DAJIE, DONGCHENG DISTRICT, BEIJING 100005, CHINA
TEL: +86-10-6518-8830 FAX: +86-10-6518-8030
China MITSUBISHI CNC Agent Service Center
(BEIJING JIAYOU HIGHTECH TECHNOLOGY DEVELOPMENT CO.)
RM 709, HIGH TECHNOLOGY BUILDING NO.229 NORTH SI HUAN ZHONG ROAD, HAIDIAN DISTRICT , BEIJING 100083, CHINA
TEL: +86-10-8288-3030 FAX: +86-10-6518-8030
Tianjin CNC Service Satellite
RM909, TAIHONG TOWER, NO220 SHIZILIN STREET, HEBEI DISTRICT, TIANJIN, CHINA 300143
TEL: -86-22-2653-9090 FAX: +86-22-2635-9050
Shenzhen CNC Service Satellite
RM02, UNIT A, 13/F, TIANAN NATIONAL TOWER, RENMING SOUTH ROAD, SHENZHEN, CHINA 518005
TEL: +86-755-2515-6691 FAX: +86-755-8218-4776
Changchun Service Satellite
TEL: +86-431-50214546 FAX: +86-431-5021690
Hong Kong CNC Service Center
UNIT A, 25/F RYODEN INDUSTRIAL CENTRE, 26-38 TA CHUEN PING STREET, KWAI CHUNG, NEW TERRITORIES, HONG KONG
TEL: +852-2619-8588 FAX: +852-2784-1323

Taiwan FA Center (MITSUBISHI ELECTRIC TAIWAN CO., LTD.)

Taichung CNC Service Center
NO.8-1, GONG YEH 16TH RD., TAICHUNG INDUSTIAL PARK TAICHUNG CITY, TAIWAN R.O.C.
TEL: +886-4-2359-0688 FAX: +886-4-2359-0689
Taipei CNC Service Satellite
TEL: +886-4-2359-0688 FAX: +886-4-2359-0689
Tainan CNC Service Satellite
TEL: +886-4-2359-0688 FAX: +886-4-2359-0689

Korean FA Center (MITSUBISHI ELECTRIC AUTOMATION KOREA CO., LTD.)

Korea CNC Service Center
1480-6, GAYANG-DONG, GANGSEO-GU SEOUL 157-200, KOREA
TEL: +82-2-3660-9631 FAX: +82-2-3664-8668

Notice

Every effort has been made to keep up with software and hardware revisions in the contents described in this manual. However, please understand that in some unavoidable cases simultaneous revision is not possible.

Please contact your Mitsubishi Electric dealer with any questions or comments regarding the use of this product.

Duplication Prohibited

This manual may not be reproduced in any form, in part or in whole, without written permission from Mitsubishi Electric Corporation.

© 2004-2007 MITSUBISHI ELECTRIC CORPORATION
ALL RIGHTS RESERVED.

MITSUBISHI CNC

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE : TOKYO BUILDING,2-7-3 MARUNOUCHI,CHIYODA-KU,TOKYO 100-8310,JAPAN

MODEL	NC Designer
MODEL CODE	008—341
Manual No.	IB-1500109 (ENG)

IB(NA)1500109(ENG)-C(0702)MEE

Specifications subject to change without notice.
Printed in Japan on recycled paper.