

GINO.NET

Reference Card

version 6.5

BRADLY ASSOCIATES LTD
Manhattan House
140 High Street
Crowthorne
Berkshire RG45 7AY
England
Tel: +44 (1344) 779381
Fax: +44 (1344) 773168
support@gino-graphics.com
www.gino-graphics.com



INTRODUCTION

This document is a summary of the classes and methods available in the GinoGraphics .NET namespace. The complete namespace consists of a set of Dynamically Linked Libraries (DLL's) which compose three functionally separate method class libraries; GINO, GINOGRAF and GINOSURF and a set of common or shared enumerator and structure classes. The documentation is split into the three method class libraries with the relevant enumerator and structure classes preceding each one.

Each method is listed in alphabetical order (within the class) with its permitted arguments, some of which are optional due to overload alternatives being made available. A short description follows, describing each of the arguments making reference to the common enumerator and structure classes where they are used.

Syntax used in this Reference Card

len Underlined arguments return a value or structure and should be defined as 'out' or 'ref' in C# as appropriate.

[len] Arguments in square brackets are optional. Where multiple arguments are shown in brackets, all or none of the set may be supplied as required.

Information in this manual is subject to change without notice.

While Bradly Associates Ltd. makes every endeavour to ensure the accuracy of this document, it does not accept liability for any errors or omissions, or for any consequences arising from the use of the program or documentation.

GINO.NET Reference Card
© Copyright Bradly Associates Ltd. 2005
All rights reserved.

All trademarks where used are acknowledged.

GINO

Arguments which represent common values are provided in a set of enumerator classes within the GinoGraphics namespace. The following are used mainly in the GINO library but some are also used in GINOGRAPH and GINOSURF: Enumerators marked with an asterix can also be passed as an integer value.

Enumerator Classes:

Alignment	Character alignment:	
	GBOTTOM	Base of character string
	GCENTRE	Centre of character string
	GTOP	Top of character string
Axis	Axis identifier:	
	GXAXIS	X Axis
	GYAXIS	Y Axis
	GZAXIS	Z Axis
BrokenLineType	Broken line type: *	
	GSOLID	
	GSHORTDASHED	
	GSHORTDOTTED	
	GSHORTCHAINED	
	GLONGDASHED	
	GLONGDOTTED	
	GLONGCHAINED	
	GDOTTED	
CgmEncoding	Cgm Encoding type:	
	GCGMCHAR	Character encoding
	GCGMBINARY	Binary encoding
CgmErrorLevel	CGM error code level:	
	GNO	No error checking
	GFAST	Fast error checking
	GFULL	Full error checking
CharacterFitOption		
	GB2P	Adjust string angle only
	GSIZE	Adjust character size and string angle
CharacterMode	Character mode:	
	GHardware	Hardware
	GPSEUDO	Pseudo hardware
	GUNTRANSFORMABLE	Software untransformable.
	GTRANSFORMABLE	Software transformable
CharacterOrientation		
	GHORIZ	Horizontal (0 degrees)
	GVERT	Vertical (90 degrees)
ClippingMode		
	GNOCLIP	No hardware or software clipping
	GHARD	Use hardware clipping (default)
	GSOFT	Use software clipping
Colour	Drawing colour index: *	
	GBACKGROUND	
	GBLACK	
	GRED	
	GORANGE	
	GYELLOW	
	GGREEN	
	GCYAN	
	GBLUE	
	G MAGENTA	
	GBROWN	
	GWHITE	

GINO

CopyPos	Segment copy position GCURRENT GANCHOR	Copy to current position Copy at segment anchor position
CursorAction	Cursor action types: GDEFAULT GRUBBERBAND GRUBBERBOX GRUBBERSQUARE GRUBBERELLIPSE GRUBBERCIRCLE GPOLYLINE	
CursorType	Cursor types: * GDEFAULT GSMALLCROSS GLARGECROSS GX GPOINTER	
DebugLevel	Device driver debug level GSTANDARD GEXTRA	Lists all device driver entries As above plus extra monitoring
DepthTestMode	Facet depth test mode GEQUALTO GNOTEQUALTO GGREATERTHAN GGREATERTHANOEQUALTO GLESSTHAN (default) GGREATERTHANOEQUALTO GNEVER GALWAYS	
Direction	Direction of arcs, facet winding etc. GCLOCKWISE GANTICLOCKWISE	
EndCondition	Curve end condition GXPOINT GNONE GANGLE	Extra point supplied No end conditions Direction defined by angle
ErrorSwitch	Gino error handling level GALLOFF GERRORON GALLON	Errors and warnings off Errors on Errors and warnings on (default)
EventType	Event data type: GALL GNULL GKEYPRESS GSEGMENT GSEGMENTANDKEY GLOCATOR GSTRING GREALS GINTEGERS GMOVEMENT GKEYRELEASE GRESIZE GPOINTERLEAVING GPOINTERENTERING GMOUSEWHEEL	All data types - used for removal Null data type Key or function button Picture segment number Picture segment number and key Screen position and key Text string Array of real values Array of integer values Pointer, mouse or tablet movement Key or function button release Window resize Pointer leaving window Pointer entering window Mouse wheel movement
FacetFace	Facet face: GBACK GFRONT	

GINO

FacetFillStyle	Facet fill style: GSOLID GHOLLOW	Solid fill Boundary only
FacetOffsetMode	Facet offset mode: GOFF GBOUNDARYAWAY GINTERIORAWAY GINTERIORNEAR GBOUNDARYNEAR	Facet drawn without offset Boundary shifted away from viewpoint Interior shifted away from viewpoint Interior shifted towards viewpoint Boundary shifted away from viewpoint
FileAccessMode	File access mode: GREADONLY GUPDATE GWRITE GSCRATCH	Open for reading only Open for updating (read/write) Open for writing only Open scratch file, removed at close
FileAssociationAction	File association action flag: GOPEN GPRINT	Open file Print file
FillAreaFlag	Fill area flag: GAREA GINVERSE	Fill polygons Fill all but polygons
FillStyle	Fill style: * GNOFILL GHOLLOW GSOLID GFINEHORIZONTAL GFINEVERTICAL GFINELEFTDIAGONAL GFINERIGHTDIAGONAL GFINEHORIZONTALGRID GFINEDIAGONALGRID GFINEHORIZONTALMESH GFINEDIAGONALMESH GCOARSEHORIZONTAL GCOARSEVERTICAL GCOARSELEFTDIAGONAL GCOARSERIGHTDIAGONAL GCOARSEHORIZONTALGRID GCOARSEDIAGONALGRID GCOARSEHORIZONTALMESH GCOARSEDIAGONALMESH	
FogMode	Fog mode: GNONE GLINEAR GEXP1 GEXP2	No fog (default) Linear for depth-cueing Exponential for cloud and heavy fog Exponential for smoke and weather haze

GINO

FontFace	Font face: *	
	GDEFAULT	Default hardware/software font
	GRoman_Simplex	Software fonts from GINO font file
	GRoman_Duplex	
	GRoman_Complex	
	GRoman_Triplex	
	GItalic_Complex	
	GItalic_Triplex	
	GScript_Simplex	
	GScript_Complex	
	GGreek_Simplex	
	GGreek_Complex	
	GGothic_English	
	GGothic_German	
	GGothic_Italian	
	GCyrillic_Complex	
	GSwiss_Solid	
	GDutch_Solid	
	GWestern	
	GComputer	
	GDisplay	
	GLatin	
	GGreek_Font_1	
	GGreek_Font_2	
	GGreek_Font_3	
	GGreek_Font_4	
	GGreek_Font_5	
	GMaths_Symbols	Symbol fonts from GINO font file
	GHershey_Maths_Symbols	
	GHershey_Symbols_1	
	GHershey_Symbols_2	
	GSymbol1_normal	
	GSymbol1_thick	
	GSymbol1_filled	
	GSymbol2_normal	
GSymbol2_filled		
GINO_Dingbats		
Gcourier	Hardware/Software fonts	
GHelvetica		
GTimes		
GAvant_Garde		
GLublin_Graph		
GNew_Century_Schoolbook		
GSouvenir		
GPalatino		
GChancery		
FontFillStyle	Font fill style: *	
	GOUTLINE	Outline only
	GFILLED	Filled font
	GOUTFILL	Fill and draw outline
FontSpacing	Font Spacing:	
	GNORMAL	Normal font spacing
	GFIXEDPITCH	Force equal spacing
GenerationMode	Generation mode (thick lines, segments etc.):	
	GHARDWARE	Use hardware generation
	GMIXWARE	Use mixture of hardware/software
	GSOFTWARE	Use software generation
GenerationSwitch	Generation switch (arcs etc.):	
	GHARD	Use hardware generation
	GSOFT	Use software generation

GINO

GinoState	Gino state:		
	GGINOINITIALIZED	Gino is initialized	
	GDEVICENOMINATED	Device nominated	
	GDEVICINITIALIZED	Device initialized	
	GPICTURESTARTED	Picture started	
GSEGMENTOPEN	Segment open		
ImageType	Image file types:		
	GBMPFILE	Uncompressed Windows Bitmap	
	GXWDFILE	X Windows Dump format	
	GICOFILE	Windows Icon file	
	GJPEGFILE	JPEG image format	
GPNGFILE	Portable Network Graphics file		
InterpolationSwitch	GXDATA		
	GYDATA		
	GZDATA		
Justification	String justification:		
	GLEFT	Left justified	
	GCENTRE	Centre justified	
GRIGHT	Right justified		
LightType	Light type		
	GAMBIENT	Light with no source or direction	
	GDIRECTIONAL	Light source from infinite distance	
	GPOINTLIGHT	Light emanating from a specified point	
GSPOTLIGHT	Spot light with concentration and spread		
LineEnd	Line end type:		
	GNONE		
	GSQUARE		
GROUND			
LineMode	Chained line mode:		
	GSOLID	Solid line	
	GCONTDASH	Dashed line continued across line segs.	
	GDISCONTDASH	Dashed line restarted at new line seg.	
	GCONTCHAIN	Chained line continued across line segs.	
GDISCONTCHAIN	Chained line restarted at new line seg.		
LineStyle	Line style index: *		
	GBACKGROUND		
	GBLACK		
	GRED		
	GORANGE		
	GYELLOW		
	GGREEN		
	GCYAN		
	GBLUE		
	GMAGENTA		
	GBROWN		
	GWHITE		
	MappingMode	GINO restoration state mapping mode:	
GABSOLUTE		Restore without mapping	
GMAPPED	Map to current device limits		
Marked	Segment marked status:		
	GUNMARK	Unmarked (default)	
GMARK	Marked		

GINO

Marker	Marker symbol: *	
	GDOT	Dot
	GSPOT	Dot
	GUP	Upwards arrow
	GDOWN	Downwards arrow
	GPLUS	Plus sign
	GCROSS	Cross
	GBOX	Small square box
	GDIAMOND	Diamond
	GCIRCLE	Small circle
GSTAR	Star or asterisk	
MetafileMappingMode	GABSOLUTE	Same size
	GMAPED	Scaled to fit
	GTRANSFORMED	Subject to transformation
	GWHOLE	All 4 quadrants (gGetDrawing)
MetafilePaperSource	GPROGRAM	Use application paper size
	GMETAFILE	Use metafile paper size
MouseEnvironment	GSCREEN	Relative to screen or display
	GDRAWINGAREA	Relative to current drawing area
NullForm	Null character form (software)	
	GNO SLASH	Without / (default)
	GSLASH	With /
	GTICK	With tick
PenType	Pen type or drawing mode: *	
	GDEFAULT	Undefined (default)
	GERASER	Erase mode
	GNOT	NOT mode
	GAND	AND mode
	GOR	OR mode
PixelDisplayMode	Pixel display mode	
	G OFF	Don't draw pixel/cell array primitives
	G ON	Draw pixel/call arrays (default)
	G BOUNDARY	Draw boundary only
PixelDrawingDirection	GTOPLEFTHORIZONTAL	
	GTOPLEFTVERTICAL	
	GTOPRIGHTHORIZONTAL	
	GTOPRIGHTVERTICAL	
	GBOTTOMLEFTHORIZONTAL	
	GBOTTOMLEFTVERTICAL	
	GBOTTOMRIGHTHORIZONTAL	
PointStorageMode	G OFF	Point storage disabled
	GSPACE	Points stored in space coordinates
	GPICTURE	Pointes stored in picture coordinates
	G CLEAR	Clear point storage
Registry	GREGCLASSESROOT	Root classes registry
	GREGCURRENTUSER	Current User Registry
	GREGLOCALMACHINE	Local Machine Registry
	GREGUSERS	Users Registry
	GREGCURRENTCONFIG	Current Configuration Registry

GINO

RegistryValueType

GREGNONE	Null value
GREGSTR	String
GREGEXPANDSTR	Expanded string
GREGBINARY	Binary value
GREGVALUE	Real value
GREGMULTISTR	Multi-string

Sensitivity

Segment sensitivity status:	
GUNSENSITIVE	Unsenstive to hits (default)
GSENSITIVE	Sensitive

ShadingMode

Shading mode	
GNONE	No shading (default)
GFLAT	Flat shading
GGOURAUD	Smooth shading using vertex normals
GPHONG	Smooth shading (interpolated normals)

Switch

On/off switch	
GOFF	
GON	

TaskPriority

Appication task priority	
GREALTIME	Highest permitted priority
GHIGH	Higher than normal
GNORMAL	Normal priority
GLOW	Lower than normal
GIDLE	Idle state

TextureCoordinateGenerationMode

GOFF	No texture coordinate generation (def)
GOBJECT	Object coordinates used
GSPHERICAL	Spherical texture coordinates

TextureFilter

Texture mapping filter	
GNEAREST	Use nearest texel (default)
GLINEAR	Use wieghted avarage of 2x2 texels
GNEARESTNEAREST	Nearest mipmap using nearest texel
GNEARESTLINEAR	Nearest mipmap using linear texel
GLINEARNEAREST	Linear mipmap using nearest texel
GLINEARLINEAR	Linear mipmap and linear texel

TextureMappingMode

GOFF	No texture mapping (default)
GOVERLAY	Overlay texture on surface
GMODULATE	Modulate texture with surface colour
GBLEND	Blend texture with constant blend col.

TextureWrappingMode

GREPEAT	Repeat texture map (default)
GCLAMP	Clamp texture to image limits

TracerSwitch

Routine tracer switch	
GOFF	No routine tracing (default)
GGINOCALLS	Trace user call to GINO library only
GLIBRARIES	Trace user calls to all libraies
GALLCALLS	Trace all calls to all libraries

TransformEditMode

GAPPLY	Apply matrix
GREPLACE	Replace current matrix

TransformationMode

GSPACE	World coordinates
GPICTURE	Drawing coordinates

GINO

TransformationSwitch	GRESET GOFF GON GINIT	Initialise model and view matrices Switch transforming off Switch transforming on Initialise model matrix and switch on
VectorType	Vector type GRELATIVE GABSOLUTE	Relative coordinates supplied Absolute coordinates supplied
ViewPlane	Plane about which view rotation is applied GYZPLANE GXZPLANE GXYPLANE	
ViewportMappingMode	GCENTRAL GBOTTOMLEFT GDEFORMED	Keep aspect ratio and centralize Keep aspect ratio, in bottom left Allow deformation of picture
ViewState	View State GNONE GPARALLEL GPERSPECTIVE	No view defined Parallel view defined Perspective view defined
Visibility	Line/segment visibility: GINVISIBLE GVISIBLE	Invisible Visible (default)
WindowingSwitch	GOFF GON GON2D GON3D	Switch windowing/clipping off Switch windowing/clipping on Set 2D window to current viewport Set 3D window to current viewport

GINO Structure Classes:

Each structure class is listed in alphabetical order followed by a list of its properties each with its value type or enumerator class following in brackets (). Only those classes with an asterisk (*) contain a parametrized constructor as in all other cases there is no need to pass the class to a GINO method.

GBRKSTY *	Broken Line Style mode (LineMode) repeat (double) dash (double) dot (double)	Broken line mode Repeat length Dash length Dot length
GCHASTY	Character Attributes type (CharacterMode) width (double) height (double) slant (double) angle (double)	Character mode Character width Character height Italic angle String angle
GDATE	Date components year (int) month (int) day (int)	
GDIM *	Paper/Device dimensions xpap (double) ypap (double)	Horizontal paper/device dimension Vertical paper/device dimension

GINO

GEVEREC	Event record key (int) impkey (int) impdat (int) nseg (int) pos (GPOINT) nargs (int) args (double[]) iargs (double[])	ASCII key code Implement generating key Implement generating data Picture segment number Screen position Number of data items Real data items Integer data items
GFNTFILSTY *	Font fill style type (FontFillStyle) ffil (FillStyle) fline (LineStyle) bfill (FillStyle) bline (LineStyle)	Font fill style Foreground filling style Foreground line style Background filling style Background line style
GFOGATT	Fog attributes mode (FogMode) colour (Colour) density (double) start (double) end (double)	Fog mode Fog colour(index or 24bit RGB) Exponential density [0.0025] Linear start depth [0.0] Linear end depth [1.0]
GHATSTY *	Hatch Style pitch (double) angle (double) xshift (double) yshift (double) xshear (double) xhatch (Switch)	Gap between lines Hatch angle Local X shift Local Y shift Shear angle Hatch direction switch
GHLSTY	HLS colour style hue (double) light (double) sat (double)	Hue Lightness Saturation
GHSVSTY	HSV colour style hue (double) sat (double) value (double)	Hue Saturation Value
GIMPLEMENTATION	Implementation attributes rmin (double) rmax (double) rsmall (double) rsig (double) imin (int) imax (int) nopr (int) nfmaz (int) nbits (int) nbmask (int) nfimpl (int) nflice (int) iso (int) dsep (string)	Minimum real value Maximum real value Minimum real value > 0.0 Smallest significant real value Minimum integer value Maximum integer value Number of integers per real word Number of bytes per integer word Number of bits per integer word Integer word with all bits set to 1 Implementation number Licence number ISO flag Directory separator
GLIBSTATE	GINO State gino (GinoState) graf (Switch) surf (Switch) menu (Switch)	GINO state GINOGRAF state GINOSURF state GINOMENU state
GLIMIT *	2D Rectangular limits xmin (double) xmax (double) ymin (double) ymax (double)	Lower X value Upper X value Lower Y value Upper Y value

GINO

GLIMIT3 *	3D Rectangular limits xmin (double) xmax (double) ymin (double) ymax (double) zmin (double) zmax (double)	Lower X value Upper X value Lower Y value Upper Y value Lower Z value Upper Z value
GLINSTY *	Line Style vis (Visibility) brk (BrokenLineType) col (Colour) width (double) type (PenType) end (LineEnd)	Visibility Broken line type Colour index Line width Pen/drawing mode Line end
GLITATT	Light Attributes state (Switch) type (LightType) col (Colour) dir (GPOINT3) pos (GPOINT3) att1 (double) att2 (double) conc (double) spang (double) spec (Colour)	Light state (GON or GOFF) Light type Light colour (index/24bit RGB) Light direction Light position Constant attenuation factor Linear attenuation factor Spot light concentration Spot light spread angle Specular colour component
GMATSTY *	Material Style ambient (double) diffuse (double) specular (double) shine (double) trans (double)	Ambient coefficient (0.0 -> 1.0) Diffuse coefficient (0.0 -> 1.0) Specular coefficient (0.0 -> 1.0) Shininess (%) Translucence (0.0 -> 1.0)
GPICATT	Picture segment attributes exist (int) vis (Visibility) sens (Sensitivity) mark (Marked) anchor (GPOINT3)	Existence Visibility Sensitivity Marked Anchor
GPIXEL	Integer pixel position (relative to top left corner) ix (int) iy (int)	X position Y position
GPOINT *	2D coordinate x (double) y (double)	X position Y position
GPOINT3 *	3D coordinate x (double) y (double) z (double)	X position Y position Z position
GPOLYGON	2D polygon component nvert (int) verts (GPOINT[])	Number of vertices Pointer to vertices
GPOLYGON3	3D polygon component nvert (int) verts (GPOINT3[])	Number of vertices Pointer to vertices
GRGBSTY *	RGB colour style red (double) green (double) blue (double)	Red component Green component Blue component

GINO

GSHADING	Shading attributes	
	mode (ShadingMode)	Shading mode
	culling (CullingMode)	Culling mode
	blending (Switch)	Blending
	winding (Direction)	Facet winding
GTEXATT	Texture mapping attributes:	
	mode (TextureMappingMode)	Texture mapping mode
	blendcol (Colour)	Blend colour
	wraps (TextureWrappingMode)	Wrapping in S
	wrapt (TextureWrappingMode)	Wrapping in T
	maxfil (TextureFilter)	Maximizing filter
minfil (TextureFilter)	Minimizing filter	
	bordercol (Colour)	Border colour
GTEXVEC	Texture coordinate transformation vector	
	Trans (TransformationMode)	Transformation mode
	xfactor (double)	
	yfactor (double)	
	zfactor (double)	
	wfactor (double)	
GTIME	Time components	
	hour (int)	
	min (int)	
	sec (int)	
	millsec (int)	
GVIEWSTATE	GINO Viewing State	
	mode (ViewState)	View state
	cflag (int)	View centre flag
	upflag (int)	View up vector flag
	dir (GPOINT3)	View direct. vector
	centre (GPOINT3)	View centre
	dist (double)	Perspective view distance
	shift (GPOINT)	View shift
	upvec (GPOINT3)	View up vector

GINO

Gino.AddEventType

(intype)

Adds data type to list of event data types. (**intype** Class=EventType)

Gino.ArchiveSegs

(unit)

Archives the complete contents of the Software Display File either from memory or file.

Gino.BuildMatrix2D

(xo, yo, dx, dy, angz, sx, sy, t2d)

Builds a 2D transformation matrix (**t2d**) containing translation, rotation and scaling factors about an origin (**xo, yo**).

Gino.BuildMatrix3D

(xo, yo, zo, dx, dy, dz, angx, angey, angz, sx, sy, sz, t3d)

Builds a 3D transformation matrix (**t3d**) containing transformation, rotation and scaling factors about an origin (**xo, yo, zo**).

Gino.CGMInterpreter

(code, unit, nseg, mode, errlev)

Interprets an encoding of a complete CGM metafile (**code** Class=CgmEncoding, **mode** Class=MetafileMappingMode, **errlev** Class=CgmErrorLevel)

Gino.ClearPolygonWorkspace

Reinitializes polygon workspace, deleting all existing polygon definitions.

Gino.ClearViewport

Clears the current viewport area by filling it with the background colour.

Gino.CloseAuxDrawingArea

(ident)

Closes an auxiliary drawing area opened by gOpenAuxDrawingArea().

Gino.CloseCGMFile

Closes the CGM metafile opened by gOpenCGMFile().

Gino.CloseDevice

Terminates output to device.

Gino.CloseGino

Terminates GINO.NET.

Gino.CloseSeg

Closes the currently opened picture segment.

Gino.CombineMatrix2D

(a2, xo, yo, dx, dy, angz, sx, sy, t2d)

Computes a transformation matrix that is the composition of the input matrix (**a2**) with rotation, translation and scaling factors.

Gino.CombineMatrix3D

(a3, xo, yo, zo, dx, dy, dz, angx, angey, angz, sx, sy, sz, t3d)

Computes a 3D transformation matrix that is the composition of the input matrix (**a3**) with rotation, transformation and scaling factors.

Gino.ConvertInteger

(number, nwidth, string)

Returns the integer value **number** as a decimal character string in **string**.

Gino.ConvertRealExponent

(value, nwidth, nplace, string)

Returns the real **value** as a decimal floating-point character in **string**.

GINO

Gino.ConvertRealFixed

(value, nwidth, nplace, string)

Returns the real **value** as a decimal fixed-point character in **string**.

Gino.ConvertRealFloat

(value, nwidth, string)

Returns the real **value** as a decimal floating-point character in **string**.

status=Gino.CopyFile

(filea, fileb)

Copies **filea** to **fileb**.

Gino.CopyPixelArea

(source, dest, ix, iy, width, height, idx, iyd)

Copies the pixel area from (**ix, iy**) of **width** and **height** on source drawing area (**source**) to (**idx, iyd**) on destination drawing area (**dest**).

Gino.CopySeg

(nseg, pos)

Makes copy of segment **nseg** (**pos** Class=CopyPos).

Gino.CreatePlanarShadowMatrix

(plane, light, a3)

Returns a viewing transformation matrix by which subsequent drawing will appear on the specified plane (e.g. a shadow).

stat=Gino.CreateRegistryKey

(registry, key, subkey)

Creates a new subkey in the Windows Registry.

stat=Gino.CreateRegistryValue

(registry, key, value, type, string, data)

Creates a new data value in the Windows Registry.

Gino.Debug

(unit, level)

Outputs trace of information flow between GINO and a device driver to **unit** (**level** Class=DebugLevel).

Gino.DefineBrokenLineStyle

(brk, rep)

Redefines broken line type where **brk** = 1 to 256 and **rep** is of type GBRKSTY

Gino.DefineFog

(mode, colour, [start, end, density])

Defines fog attributes in terms of **mode** (Class=FogMode) and **colour** (Colour) which may be colour index or 24bit RGB.

start, end= Start and end depth for linear fog

density=Fog density for exponential types [0.0025]

Gino.DefineGroupRange

(ngmin, ngmax)

Defines range of numbers available for segment groups.

Gino.DefineHatchStyle

(fill, rep)

Redefines hatch style index where **fill** is 1 to 256 and **rep** is of type GHATSTY.

Gino.DefineHLS

(col, hue, light, sat)

Redefines colour and intensity of colour index in terms of Hue, Lightness and Saturation.

Gino.DefineHSV

(col, hue, sat, value)

Redefines colour and intensity of colour index in terms of Hue, Saturation and Value.

GINO

Gino.DefineLightSource

(light, colour)

Defines ambient light source.

(light, colour, dir)

Defines directional light source from **dir** (GPOINT3)

(light, colour, pos, atten1, atten2)

Defines point light source at **pos** (GPOINT3)

(light, colour, dir, atten1, atten2, pos, conc, spread, spec)

Defines spot light

light=Light number (1-8)

colour=Light colour (index or 24bit RGB)

dir=Direction Vector (GPOINT3)

atten1=Point light const. attenuation factor

atten2=Point light linear attenuation factor

pos=Light position (GPOINT3)

conc=Spot light concentration %

spread=Spot light spread angle (0.0 to 360.0)

spec=Specular component (index or 24bit RGB)

Gino.DefineLineStyle

(line, rep)

Redefines attribute values for line style where **line** is GCURRENT or 1 to 256 and **rep** is of type GLINSTY.

Gino.DefineMaterial

(mat, rep)

Redefines attributes for material type where **mat** is 1 to 256 and **rep** is of type GMATSTY.

Gino.DefineNullChar

(nul)

Sets the required representation of the 0 (zero) character for the default software font. (**null** Class=NULLForm)

nul=GNOSLASH

nul=GSLASH

nul=GTICK

Gino.DefineParallelView

(dx, dy, dz, xcen, ycen, zcen)

Defines parallel view along direction (**dx, dy, dz**) from view centre (**xcen, ycen, zcen**).

Gino.DefinePerspView

(xe, ye, ze, dx, dy, dz, dist)

Defines perspective view along line specified by (**dx, dy, dz**) from viewpoint (**xe, ye, ze**) at perspective distance (**dist**).

Gino.DefinePictureUnits

(umils)

Sets current units to new unit of measure equal to **umils** millimetres.

Gino.DefinePixelPacking

(nbp, nrb, npw, ndir, dir)

Defines the users pixel data characteristics as used by `gDrawPixelArea()` and `gGetPixelArea()`.

Gino.DefinePointWorkspace

(nw)

Sets up area in global workspace for points storage.

Gino.DefinePolygonWorkspace

(nw)

Sets up area in global workspace for storage of polygon vertices.

GINO

Gino.DefineRGB

(col, red, green, blue)

Redefines colour and intensity of colour index in terms of Red, Green and Blue.

Gino.DefineSegGroup

(ngrp, ngmin, ngmax)

Defines range of segment numbers in group.

Gino.DefineSegWorkspace

(nw)

Sets up area in global workspace for segment storage and switches storage to that area.

Gino.DefineSphericalView

(xcen, ycen, zcen, rad, dx, dy, dz, dist)

Defines perspective view enclosed by sphere centre (**xcen**, **ycen**, **zcen**) of radius **rad** from direction (**dx**, **dy**, **dz**) at perspective distance (**dist**).

Gino.DefineTexture

(level, xgrid, ygrid, border, nbyte, pixbuf)

Defines texture map image.

level=0

Primary image

level>0

Mipmap reduction

xgrid, ygrid

Width & height

border (Class=Switch)

Border switch

nbyte

No. of bytes in data

pixbuf

Texture

Gino.DeleteEventQueue

Deletes the event queue.

stat=Gino.DeleteRegistryKey

(registry, key)

Deletes a key from the Windows Registry.

stat=Gino.DeleteRegistryValue

(registry, key, value)

Deletes a data value from the Windows Registry.

Gino.DeleteSeg

(nseg)

Deletes picture segment or segments.

Gino.DisplayAsciiChar

(asc)

Outputs single character with ASCII code = **asc**.

Gino.DisplayInteger

(number, nwidth)

Outputs integer **number** as decimal character string.

Gino.DisplayRealExponent

(value, nwidth, nplace)

Outputs real **value** as mantissa and exponent. Mantissa has **nplace** places after the decimal point.

Gino.DisplayRealFixed

(value, nwidth, nplace)

Outputs real **value** with **nplace** places after the decimal point.

Gino.DisplayRealFloat

(value, nwidth)

Outputs real **value** as mantissa and exponent. Mantissa has (**nwidth** - 6) places after the decimal point, similar to the FORTRAN E FORMAT.

Gino.DisplayStr

(string)

Outputs character **string**.

GINO

Gino.DisplayStrPolylineBy2D

(npts, points, string)

Displays a character **string** along the polyline represented by the relative points in the array **points** (GPOINT).

Gino.DisplayStrPolylineTo2D

(npts, points, string)

Displays a character **string** along the polyline represented by the absolute points in array **points** (GPOINT).

Gino.DragSeg

(nseg)

Drags picture segment or group with the cursor until key pressed.

Gino.DrawAkimaBy2D

(npts, points, beg, fin)

Draws a 2-D Akima curve through a number of relative **points** (GPOINT). (**beg,fin** Class=EndCondition)

Gino.DrawAkimaTo2D

(npts, points, beg, fin)

Draws a 2-D Akima curve through a number of absolute **points** (GPOINT). (**beg,fin** Class=EndCondition)

Gino.DrawArcBy2D

(dxc, dyc, dxe, dye, sense)

Draws 2-D arc, centre (**dxc, dyc**) to a point that lies on the line from (**dxc, dyc**) to (**dxe, dye**). All coordinates are relative to the current drawing position.

sense (Class=direction)=GCLOCKWISE

Clockwise

sense (Class=direction)=GANTICLOCKWISE

Anti-clockwise

Gino.DrawArcBy3D

(dxc, dyc, dzc, dxe, dye, dze, dxt, dyt, dzt)

Draws 3-D arc, centre (**dxc, dyc, dzc**) to a point that lies on the line from (**dxc, dyc, dzc**) to (**dxe, dye, dze**) using (**dxt, dyt, dzt**) as the start direction vector. All coordinates are relative to the current drawing position.

Gino.DrawArcTo2D

(xc, yc, xe, ye, sense)

Draws 2-D arc, centre (**xc, yc**) to a point that lies on the line from (**xc, yc**) to (**xe, ye**). All coordinates are absolute.

sense (Class=direction)=GCLOCKWISE

Clockwise

sense (Class=direction)=GANTICLOCKWISE

Anti-clockwise

Gino.DrawArcTo3D

(xc, yc, zc, xe, ye, ze, dxt, dyt, dzt)

Draws 3-D arc, centre (**xc, yc, zc**) to a point that lies on the line from (**xc, yc, zc**) to (**xe, ye, ze**) using (**dxt, dyt, dzt**) as the start direction vector. All coordinates are absolute.

Gino.DrawBezierBy2D

(npts, points2)

Draws a 2D bezier curve through the relative **points2** (GPOINT).

Gino.DrawBezierBy3D

(npts, points3)

Draws a 3D bezier curve through the relative **points3** (GPOINT3).

Gino.DrawBezierSphere

(xp, yp, zp, radius, [urot, vrot, wrot], [ucomp, vcomp])

Draws a sphere centred about **xp, yp, zp**, constructed from eight Bezier surfaces.

GINO

Gino.DrawBezierSurface

(nx, ny, mesh, [ucomp, vcomp])

Draws a Bezier surface from a 2D array of 3D control points held in the array **mesh** (GPOINT3).

Gino.DrawBezierTo2D

(npts, points2)

Draws a 2D bezier curve through the absolute **points2** (GPOINT).

Gino.DrawBezierTo3D

(npts, points3)

Draws a 3D bezier curve through the absolute **points3** (GPOINT3).

Gino.DrawBezierVolume

(xp, yp, zp, npts, points, [ucomp, vcomp],[urot, vrot, wrot])

Draws four Bezier surfaces from a single 2D Bezier curve defined through control **points** (GPOINT) rotated around a central axis starting at **xp, yp, zp**.

Gino.DrawBox

(xp, yp, zp, [udim, vdim, wdim], [urot, vrot, wrot], [uvec, vvec, wvec, abs], [ucomp, vcomp, wcomp])

Draws a box using dimensions or vectors where **xp, yp, zp**=bottom left corner. (**abs** Class=VectorType)

Gino.DrawCellArray

(x1, y1, x2, y2, npixx, npixy, isx, isy, idx, idy, pixbuf)

Outputs rectangular cell array of dimension (**npixx** by **npixy**). Corners are in current units.

Gino.DrawCone

(xp, yp, zp, rad1, rad2, [height], [vvec, abs], [urot, vrot, wrot], [ucomp, vcomp])

Draws a cone in 3-D space with specified base and radii. Height is specified using **gHeight** or **gVVec**. If **rad2=0.0**, then a pyramid is generated. (**abs** Class=VectorType)

Gino.DrawCube

(xp, yp, zp, dim, [urot, vrot, wrot], [ucomp, vcomp, wcomp])

Draws a cube of size **dim** centred at the specified position.

Gino.DrawCurveBy2D

(npts, points, beg, fin)

Draws smooth curve (with optionally specified end conditions **beg** and **fin**) through **npts** relative increments in array **points** (GPOINT) starting from the current drawing position. (**beg,fin** Class=EndCondition)

Gino.DrawCurveTo2D

(npts, points, beg, fin)

Draws smooth curve with optionally specified end conditions **beg** and **fin** through **npts** points specified in array **points** (GPOINT). (**beg,fin** Class=EndCondition)

Gino.DrawCylinder

(xp, yp, zp, radius, [height], [vvec, abs], [urot, vrot, wrot], [ucomp, vcomp])

Draws a cylinder in 3-D space with specified base position and **radius**. Height is specified using **gHeight** or **gVVec**. (**abs** Class=VectorType)

GINO

Gino.DrawFacet

(npts, points, [normals], [textcoords], [colours])
Display facet containing **npts** (3 recommended) in **points** (GPOINT3) array.
normals (GPOINT3) Optional normals
textcoords (GPOINT3) Optional Texture coordinates
colours (Colour/int) Optional Vertex colours

Gino.DrawLineBy2D

(dx, dy)
Draws straight line from current drawing position to relative point (**dx**, **dy**).

Gino.DrawLineBy3D

(dx, dy, dz)
Draws straight line from current drawing position to relative point (**dx**, **dy**, **dz**).

Gino.DrawLineTo2D

(x, y)
Draws straight line from current drawing position to absolute point (**x**, **y**).

Gino.DrawLineTo3D

(x, y, z)
Draws straight line from current drawing position to absolute point (**x**, **y**, **z**).

Gino.DrawMarker

(nsym)
Draws either a software or hardware symbol centred about the current drawing position. (**nsym** Class=Marker)

Gino.DrawPixel

(ix, iy, pix)
Draws a pixel at (**ix**, **iy**) of colour **pix**.

Gino.DrawPixelArea

(ix, iy, npixx, npixy, isx, isy, idx, idy, pixbuf)
Draws a rectangular pixel array from colour values stored in **pixbuf**.

Gino.DrawPolygonBound

(line)
Draws boundaries of all currently selected polygons using line style index **line**.

Gino.DrawPolylineBy2D

(npts, points2)
Draws straight lines joining **npts** points specified as relative coordinate increments in array **points** (GPOINT) starting from the current drawing position.

Gino.DrawPolylineBy3D

(npts, points3)
Draws straight lines joining **npts** points specified as relative coordinate increments in array **points** (GPOINT3) starting from the current drawing position.

Gino.DrawPolylineSet2D

(npol, polylines2)
Draws 2D polyline set from array of GPOLYGON structures.

Gino.DrawPolylineSet3D

(npol, polylines3)
Draws 3D polyline set from array of GPOLYGON3 structures.

Gino.DrawPolylineTo2D

(npts, points2)
Draws straight lines joining **npts** points specified in array **points** (GPOINT) starting from the current drawing position.

GINO

Gino.DrawPolylineTo3D

(npts, points3)

Draws straight lines joining **npts** points specified in array **points** (GPOINT3) starting from the current drawing position.

Gino.DrawPolymarkerBy2D

(npts, points2, nsym)

Draws symbol at **npts** points specified as relative coordinate increments in array **points** (GPOINT) starting from the current drawing position. (**nsym** Class=Marker)

Gino.DrawPolymarkerBy3D

(npts, points3, nsym)

Draws symbol at **npts** points specified as relative coordinate increments in array **points** (GPOINT3) starting from the current drawing position. (**nsym** Class=Marker)

Gino.DrawPolymarkerTo2D

(npts, points2, nsym)

Draws symbol at **npts** points specified as absolute coordinates in array **points** (GPOINT). (**nsym** Class=Marker)

Gino.DrawPolymarkerTo3D

(npts, points3, nsym)

Draws symbol at **npts** points specified as absolute coordinates in array **points** (GPOINT3). (**nsym** Class=Marker)

Gino.DrawRect3D

(xmin, xmax, ymin, ymax, zmin, zmax)

Draws a rectangular parallelepiped with sides perpendicular to current axes.

Gino.DrawRuledBezierSurface

(np1, points1, np2, points2, [ucomp, vcomp])

Draws a ruled surface between two 3D Bezier curves (GPOINT3). If the two curves have a different number of points, the smaller is elevated to match the larger.

Gino.DrawSeg

(nseg)

Redraws an existing segment from the segment display file.

Gino.DrawShadedPolylineTo3D

(npts, points, normals, [textcoords])

Draws a 3D polyline through the absolute **points** (GPOINT3) and shaded/textured according to the **normals** (GPOINT3).

Gino.DrawSphere

(xp, yp, zp, radius, [urot, vrot, wrot], [ucomp, vcomp])

Draws a sphere in 3-D space with specified centre and **radius**.

Gino.DrawSplineBy2D

(npts, points2, beg, fin)

Draws a 2-D spline curve through a number of relative **points** (GPOINT) starting at current position. (**beg,fin** Class=EndCondition)

Gino.DrawSplineBy3D

(npts, points3, beg, fin)

Draws a 3-D spline curve through a number of relative **points** (GPOINT3) starting at current position. (**beg,fin** Class=EndCondition)

Gino.DrawSplineSurface

(nx, ny, mesh, [ucomp, vcomp])

Draws a spline surface from a 2D array of 3D points held in **mesh**.

Gino.DrawSplineTo2D

(npts, points2, beg, fin)

Draws a 2-D spline curve through a number of absolute **points** (GPOINT). (**beg,fin** Class=EndCondition)

GINO

Gino.DrawSplineTo3D

(npts, points3, beg, fin)

Draws a 3-D spline curve through a number of absolute **points** (GPOINT3). (**beg,fin** Class=EndCondition)

Gino.DrawSweptBezierSurface

(np1, points1, np2, points2, [ucomp, vcomp])

Draws a Bezier surface by sweeping one Bezier curve (GPOINT3) along a second Bezier curve.

Gino.DrawTabulatedBezierSurface

(np, points, vector, [ucomp, vcomp])

Draws a tabulated surface by extruding a 3D Bezier curve (GPOINT3) along a 3D **vector** (GPOINT3).

Gino.DrawVolume

(xp, yp, zp, npts, points, [vvec, abs], [urot, vrot, wrot], [ucomp])

Draw a volume of rotation at the specified position according to an edge definition held in the **points** array (GPOINT3). Orientation is set by vector **vvec** or rotations. (**abs** Class=VectorType)

Gino.DrawWedge

(xp, yp, zp, [udim, vdim, wdim], [uvec, vvec, wvec, abs], [urot, vrot, wrot])

Displays wedge using dimensions, or vectors where **xp, yp, zp**=bottom left corner (**abs** Class=VectorType)

Gino.Dummy

Nominates notional device. Does not produce graphical output.

Gino.EditSegTo2D

(nseg, tag, t2, sw)

Replaces the modelling transformation matrix that immediately follows the identifier **tag** in segment **nseg**, with 2D matrix **t2**. Matrix application is set by **sw** (Class=TransformEditMode).

Gino.EditSegTo3D

(nseg, tag, t3, sw)

Replaces the modelling transformation matrix that immediately follows the identifier **tag** in segment **nseg**, with 3D matrix **t3**. Matrix application is set by **sw** (Class=TransformEditMode).

Gino.ElevateBezier2D

(npts, points2)

Returns an increased (by 1) set of control points that equates to the bezier curve given by **points2** (GPOINT).

Gino.ElevateBezier3D

(npts, points3)

Returns an increased (by 1) set of control points that equates to the bezier curve given by **points3** (GPOINT3).

Gino.EndBatchUpdate

Ends batch of modifications to a buffered display file, and outputs to display.

Gino.EndPolygon

Closes polygon, after which no more vertices can be added to it.

Gino.EnqArcState

(sw, nincs, tol)

Returns arc settings; sw (Class=GenerationSwitch), number of increments and tolerance.

Gino.EnqBrokenLine

(brk)

Returns current broken line type.

Gino.EnqBrokenLineStyle

(brk, rep)

Returns attributes of stored broken line type **brk** in **rep** (GBRKSTY).

GINO

Gino.EnqCharAttribs

(rep)

Returns character settings in structure of type GCHASTY.

Gino.EnqCharTransform

(dx, dy, point)

Transforms a coordinate position according to the current character angle and italic settings.

Gino.EnqClippingMode

(sw)

Enquires hardware clipping mode (Class = ClippingMode).

Gino.EnqColourInfo

(ndc, ndt)

Returns device's number of colours and colour display type.

ndc =0	Monochrome
ndc =1	Monochrome + erase
ndc >1	Colour/greyscale
ndt =0	No colours
ndt ±1	Fixed colour
ndt ±2	Static colour palette
ndt ±3	Dynamic colour
ndt ±4	Direct/True colour

lcfg=Gino.EnqConfigStatus

([cfgdir])

Returns status of GINO Configuration file with optional location directory.

Gino.EnqCursorAction

(action, lverts, points)

Returns current settings for the cursor action type as set by Gino.SetCursorAction(). (**action** Class=CursorAction)

Gino.EnqCursorType

(type, forcol, bakcol)

Returns the current settings for the cursor type as set by Gino.SetCursorType(). (**type** Class=CursorType, **forcol**,**bakcol** Class=Colour)

Gino.EnqCurveAttribs2D

(dxbeg, dybeg, dxfin, dyfin, begp2, finp2)

Returns 2D curve end conditions. Points **begp2** and **finp2** are of type GPOINT.

Gino.EnqCurveAttribs3D

(dxbeg, dybeg, dzbeg, dxfin, dyfin, dzfin, begp3, finp3)

Returns 3D curve end conditions. Points **begp3** and **finp3** are of type GPOINT3.

Gino.EnqDepthMode

(mode, dinit)

Enquires depth buffer operation settings. (**mode** Class=DepthTestMode)

Gino.EnqDeviceState

(devstate)

Returns information relating to the currently nominated device in structure of type GDEVSTATE.

Gino.EnqDrawingLimits

(dim, type)

Returns drawing limits (GDIM) in current units and paper type.

Gino.EnqEscapeChar

(cha)

Returns string escape character.

GINO

Gino.EnqFacetFillStyle

(fill)

Returns current facet fill style (Class=FacetFillStyle).

Gino.EnqFacetMaterialProps

(face, amb, diff, spec, emit, shine, trans)

Enquires facet material properties directly (not by material table) for **face** (Class=FacetFace).

Gino.EnqFacetOffsetMode

(mode)

Enquires current facet offset mode. (Class=FacetOffsetMode)

Gino.EnqFog

(attribs)

Returns fog mode and attributes in structure of type GFOGATT.

Gino.EnqFontStyle

(font, style, weight, space, rep)

Returns the current settings of the font attributes where style is of type GFNTFILSTY.

Gino.EnqGinoState

(gstate)

Returns current operating state (GLIBSTATE) of GINO and its associated libraries.

Gino.EnqHardFontList

(list, n, count)

Returns a **list** of hardware font numbers that the current device has available.

Gino.EnqHatchStyle

(fill, rep)

Returns attributes of hatch style index (**fill**) from table in structure of type GHATSTY.

Gino.EnqHLS

(col, hls)

Returns Hue, Lightness and Saturation of colour index in structure of type GHLSSTY.

Gino.EnqHSV

(col, hsv)

Returns Hue, Saturation and Value of colour index in structure of type GHSVSTY.

Gino.EnqImageFile

(file, type, ixgrid, iygrid, nbbp, ncols)

Enquires the size and colour information of an external image file; its **type** (Class=ImageType), dimensions, and colour information.

type=0

Unknown

Gino.EnqImpAttribs

(implem)

Enquires various attributes of the current GINO implementation placing them in a structure of type GIMPLEMENTATION.

state=Gino.EnqKeyState

(key)

Returns the state of keyboard or mouse **key**.

state=0

Key up

state=1

Key down

Gino.EnqLastErrors

(list, n, count)

Returns **n** most recent error and warning numbers in array **list** and the current **count**.

GINO

Gino.EnqLightAttribs

(light, att)

Enquires lighting attributes of **light** (1-8) in structure of type GLITATT.

Gino.EnqLineColour

(col)

Returns currently requested line colour.

Gino.EnqLineEnd

(end)

Returns currently requested line end type. (Class=LineEnd)

Gino.EnqLineStyle

(line, rep)

Returns attributes of line style index in structure of type GLINSTY.

Gino.EnqLineVis

(vis)

Returns line visibility setting. (Class=Visibility)

Gino.EnqLineWidth

(width)

Returns currently requested line **width** in current units.

Gino.EnqLineWidthMode

(sw)

Returns the thick line generation setting (Class=GenerationMode).

Gino.EnqLineWidthScaling

(scale)

Returns the line width scale factor set by Gino.SetLineWidthScaling().

Gino.EnqMaskState

(sw, bounds)

Returns the current masking state and **bounds** (GLIMIT).

sw=0

Masking off

sw=1

Rectangle mask

sw=2

Polygonal mask

Gino.EnqMaterial

(mat, rep)

Returns material type in structure of type GMATSTY.

Gino.EnqMaterialAttribs

(fcol, bcol, fmat, bmat)

Enquire material attributes.

fcol

Front facing colour

bcol

Back facing colour

fmat

Front facing material

bmat

Back facing material

Gino.EnqMaxDrawingLimits

(dim)

Returns the maximum paper or window limits for the currently nominated device (GDIM).

Gino.EnqMousePos

(env, point)

Return current mouse position in **point** (GPIXEL) relative to **env** (Class=MouseEnvironment):

Gino.EnqNumberOfErrors

(count)

Returns number of errors and warnings generated since last call to Gino.SetErrorTrap(Switch.GON). Returns -1 if error trapping disabled.

GINO

Gino.EnqOpenSeg

(nseg)

Returns currently open segment. **nseg**=0 if none open.

Gino.EnqPenType

(type)

Returns requested pen type.

Gino.EnqPicturePos

(point)

Returns current drawing position in picture coordinates (GPOINT3).

Gino.EnqPixelAttribs

(ori, xzca, ysca, xrep, yrep)

Returns current pixel orientation, scaling and replication settings.

Gino.EnqPixelPacking

(nbp, nrb, npw, ndir, dir)

Returns the users pixel data characteristics as set by Gino.SetPixelPacking().

Gino.EnqPixelPos

(xsc, ysc, pix)

Returns the pixel coordinates (GPIXEL) of a screen position (**xsc**, **ysc**) with respect to picture axes.

Gino.EnqPixelResolution

(nxpix, nypix)

Returns the pixel resolution for the current device.

Gino.EnqPointMode

(switch)

Enquires point storage mode (Class=PointStorageMode).

Gino.EnqPolygonList

(list, n, count)

Returns a list of polygon identifiers that are currently selected for polygon filling with the gFillSelectedPolygons() routine.

Gino.EnqPolygonMaskList

(list, n, count)

Returns a list of polygon identifiers that are currently selected for polygon masking.

Gino.EnqPolygonWindowList

(list, n, count)

Returns a list of polygon identifiers that are currently selected for polygon windowing.

Gino.EnqPolygonWorkspace

(npoly, nvert, nfree, ident)

Returns information about polygon storage.

Gino.EnqPosOfPixel

(ix, iy, point)

Returns the screen coordinates (GPOINT3) of a pixel position (**ix**, **iy**).

Gino.EnqQueueLength

(len)

Returns number of queue events waiting.

stat=Gino.EnqRegistryKeyInfo

(registry, key, nsub, maxsub, nval, maxval, maxdat)

Returns information pertaining to a Windows Registry key, namely the number of subkeys, their maximum length, the number of data values, their maximum length and maximum data setting (in bytes).

GINO

Gino.EnqRegistryValue

(registry, key, value, type, string, data)

Returns the **type** and **string** or **data** setting of a Windows Registry data value.

Gino.EnqRGB

(col, rgb)

Returns Red, Green and Blue components of colour index in structure of type GRGBSTY.

Gino.EnqSavdraDimension

(unit, type, dim)

Returns the **type** and paper limits (GLIMIT) of a SAVDRA metafile without interpreting the whole file.

Gino.EnqSavdraSegAttribs

(unit, nseg, att)

Returns segment attributes (GPICATT) of segment in SAVDRA file.

Gino.EnqSavdraSegList

(unit, list, n, count)

Returns **list** of segment numbers in a SAVDRA file.

Gino.EnqSegAttribs

(nseg, att)

Returns the segment attributes of the segment nseg in structure of type GPICATT.

Gino.EnqSegGroup

(ngrp, segmin, segmax)

Returns range of segment numbers in group.

Gino.EnqSegHit

(nseg, x, y, radius)

Searches for picture segment within hit-area of **radius** centred on (x, y). Returns -1 if no segment in hit area.

Gino.EnqSegTransform

(nseg, xscs, yscs, ang, xpos, ypos)

Returns the current segment transformation for segment **nseg**.

Gino.EnqSegTransform2D

(nseg, a)

Returns the 2D segment transformation matrix of segment **nseg**.

Gino.EnqSegWorkspace

(nw, nfree)

Returns the number of real words that have been allocated in the global workspace for use by the Software Display File and the amount of free space in that area.

Gino.EnqSelectedPen

(col, width, type)

Returns physical pen attributes currently implemented by the output device.

Gino.EnqShadingMode

(att)

Returns shading mode attributes in structure of type GSHADING.

Gino.EnqSpacePos

(point)

Returns current drawing position in space coordinates (GPOINT3).

Gino.EnqSplineTension

(tension)

Enquires spline curve tension.

GINO

Gino.EnqViewport3D

(piclim3, viewlim)

Enquires 3D user space cube **piclim** (GLIMIT3) and mapping to 2D viewport (GLIMIT).

Gino.EnqViewportMode

(sw)

Returns the current setting of the viewports aspect ratio/position, as set by Gino.SetViewportMode(). (Class=ViewportMappingMode)

Gino.EnqViewportState

(sw, clp, limit)

Returns current viewport switches and actual viewport **limits** (GLIMIT) in paper coordinates. (**sw** Class=ViewportMappingMode, **clp** Class=Switch)

Gino.EnqViewTransformMode

(mode)

Returns transformation mode. (Class=GenerationSwitch)

mode=GHARD

Hardware

mode=GSOFT

Software

Gino.EnqWindowState

(sw, bounds)

Returns current windowing state (**sw**), and current window limits in **bounds** (GLIMIT3).

sw=0

Windowing off

sw=1

User window

sw=2

Viewport limits

sw=3

Polygonal window

Gino.EnqWorkingDir

(directory)

Returns current working directory.

Gino.EnqWorkspaceLimit

(nl)

Returns size of global workspace area defined by Gino.SetWorkspaceLimit().

stat=Gino.ExecuteSysCommand

(command, [gShow, gSuspend, gHandle])

Executes system command within GINO application. Returns zero or error code.

Gino.ExtendSeg

(nseg)

Reopens picture segment for additional drawing.

stat=Gino.Fclose

(unit)

Closes file opened with gFopen().

Gino.FillPolygonBy2D

(fill, line, inv, npts, points2)

Fills an area defined by the relative boundary points in array **points2** (GPOINT). (**fill** Class=FillStyle, **line** Class=LineStyle, **inv** Class=FillAreaFlag)

Gino.FillPolygonBy3D

(fill, line, inv, npts, points3)

Fills an area defined by the relative boundary points in array **points3** (GPOINT3). (**fill** Class=FillStyle, **line** Class=LineStyle, **inv** Class=FillAreaFlag)

Gino.FillPolygonSet2D

(fill, line, inv, npol, polygons2)

Fills a set of **npol** 2D polygons defined in array **polygons2** (GPOLYGON). (**fill** Class=FillStyle, **line** Class=LineStyle, **inv** Class=FillAreaFlag)

GINO

Gino.GetEventRecord

(intype, everec)

Returns **event** data (GEVEREC) of event type **intype** (Class=EventType) after Gino.WaitForEvent() or Gino.DragSeg() are called.

Gino.GetFileAssociation

(ext, action, application)

Returns the registered **application** pathname (and switches) associated with a file extension (**ext**) and **action** (Class=FileAssociationAction))

Gino.GetFullDirList

(pattern, n, files, types, dates, sizes)

Returns list of **files** matching file search pattern (**pattern**) together with their associated attributes.

Gino.GetImageFile

(type, file, coldef, coloff, collim, ixgrid, iygrid, psiz, pix)

Reads an external image **file** of **type** (Class=ImageType) into a pixel array (**pix**), mapping the colours as requested.

coldef=0

Skip colour table

coldef=1

Update GINO colours

coldef=2

Map to GINO colours

Gino.GetPicture

(unit, nseg)

Interprets segment or segments from device independent SAVDRA or SAVPIC metafile on file **unit**.

Gino.GetPixel

(ix, iy, pix)

Reads a single pixel value from device.

Gino.GetPixelArea

(ix, iy, npixx, npixy, isc, isr, idx, idy, pixbuf)

Reads a rectangular pixel array from a devices pixel display and stores it in the array **pixbuf**.

Gino.GetRand

(rand)

Returns non-repeating random number in the range 0.0 to 1.0.

Gino.GetTransform2D

(a2)

Stores copy of current 2D transformation in (3x2)-element array **a2**.

Gino.GetTransform3D

(a3)

Stores copy of current 3D transformation in (4x4)-element array **a3**.

Gino.GetViewParams

(vdata)

Returns viewing parameters into 15-element array **vdata**.

Gino.GetViewState

(vstate)

Returns viewing parameters into structure of type GVIEWSTATE.

Gino.InitView

Resets viewing parameters to initial defaults.

Gino.InsertSegRef

(nseg)

Inserts a reference to segment **nseg** into the currently opened segment.

Gino.InsertSegTag

(tag)

Sets a user-supplied identifier in a picture segment to facilitate later editing.

GINO

Gino.InterpolateData2D

(nopt, ptint, npts, points2, nptout, ptout1)

Returns array of interpolated values from supplied value **ptint** which is of type **nopt** (Class=InterpolationSwitch). Input data is supplied in **points2** (GPOINT) and returned in **nptout** and array of size **nptout**.

Gino.InterpolateData3D

(nopt, ptint, npts, points3, nptout, ptout1, ptout2)

Returns array of interpolated values from supplied value **ptint** which is of type **nopt** (Class=InterpolationSwitch). Input data is supplied in **points2** (GPOINT3) and returned in **nptout** and array of size **nptout**.

Gino.InterpretCGMElement

(element)

Interprets the next **element** identifier found by gGetCGMElement() when interpreting a CGM metafile element by element.

status=Gino.MakeDir

(path)

Creates a directory specified by **path**.

Gino.MarkSeg

(nseg, mark)

Switches marking or highlighting of picture segment or segments according to **mark** (Class=Marked).

Gino.ModifyTransform2D

(a2)

Modifies current transformation by that saved in (3x2)-element array **a2**.

Gino.ModifyTransform3D

(a3)

Modifies current transformation by that saved in (4x4)-element array **a3**.

Gino.ModifyView

(a3)

Modifies the current viewing matrix (e.g. supplied from Gino.CreatePlanarShadowMatrix()).

Gino.MoveBy2D

(dx, dy)

Moves by (**dx**, **dy**) from current drawing position.

Gino.MoveBy3D

(dx, dy, dz)

Moves by (**dx**, **dy**, **dz**) from current drawing position.

Gino.MoveSegBy2D

(nseg, dx, dy)

Repositions picture segment anchor by relative increment (**dx**, **dy**).

Gino.MoveSegTo2D

(nseg, x, y)

Repositions anchor of picture segment at (**x**, **y**).

Gino.MoveTo2D

(x, y)

Moves from current position to (**x**, **y**).

Gino.MoveTo3D

(x, y, z)

Moves from current position to (**x**, **y**, **z**).

Gino.MoveToNextLine

Moves the current position to the beginning of the next line in a text block.

GINO

Gino.MoveViewCentre

(dist)

Moves view centre **dist** along line of sight in direction of viewing.

Gino.NewDrawing

Clears the drawing area (without removing segments from the display file).

Gino.OpenAuxDrawingArea

(ident, title, xp, yp, width, height)

Opens an auxiliary drawing area at specified position (**xp, yp**), with **width** and **height**. Areas may be visible (even numbered **ident**) or invisible (odd numbered **ident**).

Gino.OpenCGMFile

(code, unit, mode, errlev)

Opens a CGM metafile on **unit** for interpretation element by element. (**code** Class=CgmEncoding, **mode** Class=MetafileMappingMode, **errlev** Class=CgmErrorLevel)

Gino.OpenGino

Initializes GINO.NET.

Gino.OpenSeg

(nseg)

Starts new picture segment.

Gino.PlaySound

(freq, time)

Generates output on system speaker of specified frequency and time. Negative values of **freq** access standard Windows sounds.

Gino.PolygonHit

(**ident**, x, y, radius)

Searches polygon workspace for polygons that overlap the hit-area of **radius** centred on (**x, y**). Returns -1 if no polygons found.

Gino.PopTransform

Resets current transformation to that saved by last call to gPushTransform().

Gino.PosViewCentre

(xp, yp)

Maps view centre onto point (**xp, yp**).

Gino.PushTransform

Stores copy of current transformation (up to ten copies).

Gino.ReduceBezier2D

(npts, points2)

Returns a reduced (by 1) set of control points that approximates to the curve given by **points2** (GPOINT).

Gino.ReduceBezier3D

(npts, points3)

Returns a reduced (by 1) set of control points that approximates to the curve given by **points3** (GPOINT3).

status=Gino.RemoveDir

(path)

Removes the directory specified by **path**.

Gino.RemoveEventType

(intype)

Deletes data type (Class=EventType) from the list set up by Gino.AddEventType().

status=Gino.RemoveFile

(file)

Removes the file from the specified path.

GINO

Gino.RemoveSegGroup

(ngrp)

Removes segment group or groups formed by Gino.DefineSegGroup().

status=Gino.RenameFile

(filea, fileb)

Renames **filea** to **fileb**.

Gino.RenameSeg

(nseg, newseg)

Renames picture segment to **newseg**.

Gino.RestoreGinoState

(map)

Restores previously stored GINO state table and updates device to match using **map** (Class=MappingMode).

Gino.RestoreTransform

Restores state of transforming saved by last call to gSaveTransform().

Gino.RetrieveSegs

(unit)

Restores the complete contents of an archived Software Display File into memory or file.

Gino.ReturnDirDate

(pdate, date, time)

Unpacks directory date stamp returned by gGetFullDirList().

n=Gino.ReturnInternalPts2D

(nn, points2, np, polyline2, npts, npol)

Returns **n** polylines in **points** (GPOINT) and **polyline2** (GPOLYGON) from internal point storage.

nn, np

Supplied arrays sizes

npts

No. of points stored

npol

No. of polylines stored

n=Gino.ReturnInternalPts3D

(nn, points3, np, polyline3, npts, npol)

Returns **n** polylines in **points** (GPOINT3) and **polyline3** (GPOLYGON3) from internal point storage.

nn, np

Supplied arrays sizes

npts

No. of points stored

npol

No. of polylines stored

Gino.ReturnPlanarNormal

(npts, points, normal)

Calculates planar **normal** (GPOINT3) from **npts points**.

Gino.ReturnRegistryKeys

(registry, key, nkey, subkeys)

Returns the list of (up to **nkey**) **subkeys** under the Windows Registry **key**.

Gino.ReturnRegistryValues

(registry, key, nval, values, types, strings, data)

Returns the list of (up to **nval**) **values** and their **types** and settings under the Windows Registry **key**.

Gino.ReturnStrInfo

(string, rln, nnl, tch, sch, nesc)

Returns information concerning the characteristics of **string**.

rln

Maximum length

nnl

No. of lines

tch

Max char height

sch

Height above base line

nesc

No of non escape chars

GINO

Gino.Rotate2D

(angle)

Rotates space axes about origin by **angle** degrees. Positive degrees = anticlockwise.

Gino.Rotate3D

(axis, angle)

Rotates space axes about **axis** (Class=Axis) by **angle** degrees. Rotation is right-handed in sense with respect to **axis**.

Gino.SaveGinoState

Saves GINO state table on stack.

Gino.SaveLineStyle

(line)

Stores current line attributes as line style index **line**.

Gino.SaveTransform

Stores information about the state of transforming for subsequent recall by gRestoreTransform().

Gino.Scale2D

(f1, f2)

Scales X and Y coordinates by separate scaling factors, **sx** and **sy** respectively.

Gino.Scale3D

(f1, f2, f3)

Scales X, Y and Z coordinates by separate scaling factors, **sx**, **sy** and **sz** respectively.

Gino.SelectDrawingArea

(ident)

Selects auxiliary drawing area for subsequent GINO output. **ident**=0 (even) for screen, 1(odd) for backing store etc.

Gino.SelectPolygons

(list, n)

Specifies current list of polygon identifiers in array **list** of **n** items. **n** = 0 deletes existing list.

Gino.SetAlphaMode

Switches on-line device into character mode and empties graphics buffer.

Gino.SetArcIncrement

(nincs)

Sets number of increments to draw full circle (software arcs).

Gino.SetArcMode

(sw)

Switches hardware arcs on/off (Class=GenerationSwitch).

Gino.SetArcTolerance

(tol)

Sets tolerance for software arcs.

Gino.SetBrokenLine

(brk)

Sets broken line type (Class=BrokenLineType).

Gino.SetBrokenLineMode

(sw)

Switches hardware broken lines on/off (Class=GenerationSwitch).

Gino.SetCharFont

(font)

Sets the current character **font** (FontFace) to be used by all subsequent character and string output routines.

GINO

Gino.SetCharSize

(width, height)
Sets character size in current units.

Gino.SetCharSizePoint

(points)
Sets character size in points (1/72nd inch).

Gino.SetCharTransformMode

(sw)
Switches software transformable characters (Class=Switch).

Gino.SetClippingMode

(sw)
Sets clipping mode (Class=ClippingMode).

Gino.SetColourInfo

(ndc, ndt)
Sets device colour capabilities..

ndc=0	Monochrome
ndc=1	Monochrome + erase
ndc>1	Colour/greyscale
ndt=0	No colours
ndt±1	Fixed colour
ndt±2	Static colour palette
ndt±3	Dynamic colour
ndt±4	Direct/True colour

Gino.SetCursorAction

(action, lverts, points)
Sets the **action** (Class=CursorAction) of the CURSOR or pointer when the Gino.GetCursorEvent() routine is used. Where **action** = GPOLYLINE, **lverts** coordinates in **points** (GPOINT) are used.

Gino.SetCursorPos

(x, y)
Defines start position for Gino.GetCursorEvent().

Gino.SetCursorType

(type, forcol, bakcol)
Sets the **type** (Class=CursorType) of the CURSOR or pointer for use within the Gino.GetCursorEvent() or Gino.WaitForEvent() routines.

Gino.SetCurveAttribs2D

(dxbeg, dybeg, dxfin, dyfin, xbeg, ybeg, xfin, yfin)
Sets 2D curve end conditions.

dxbeg, dybeg	Start slope/angle
dxfin, dyfin	End slope/angle
xbeg, ybeg	Extra start point
xfin, yfin	Extra end point

Gino.SetCurveAttribs3D

(dxbeg, dybeg, dzbeg, dxfin, dyfin, dzfin, xbeg, ybeg, zbeg, xfin, yfin, zfin)
Sets 3D curve end conditions.

dxbeg, dybeg, dzbeg	Start slope/angle
dxfin, dyfin, dzfin	End slope/angle
xbeg, ybeg, zbeg	Extra start point
xfin, yfin, zfin	Extra end point

Gino.SetDebugSwitch

(sw)
Switches controlling the output from Gino.Debug() (Class=Switch).

Gino.SetDepthMode

(mode, dinit)
Sets depth buffer **mode** (Class=DepthTestMode) and initial value.
dinit Initial depth value [1.0]

GINO

Gino.SetDeviceFilename

(filename, ntype)

Specifies output file name and format.

ntype=0

Unformatted

ntype=-1

Formatted

Gino.SetDeviceTitle

(title)

Defines device title banner string.

Gino.SetDialogueVis

(vis)

Switches dialogue area visibility (Class=Visibility):

Gino.SetDrawingLimits

(dim, type)

Specifies plotter paper size (GDIM) in current units and paper type.

Gino.SetErrorFile

(unit)

Sets the output **unit** for subsequent error and tracer messages.

Gino.SetErrorMode

(sw)

Switches error and/or warning messages on/off (Class=ErrorSwitch).

Gino.SetErrorTrap

(sw)

Switches error trapping on/off (Class=Switch).

Gino.SetEscapeChar

(cha)

Sets string escape character.

Gino.SetFacetFillStyle

(fill)

Sets current facet fill style (Class=FacetFillStyle).

Gino.SetFacetMaterialProps

(face, amb, diff, spec, emit, shine, trans)

Sets facet material properties directly for each **face**

(Class=FacetFace) (rather than by material table) using index or 24bit

RGB. Colour black (Colour.GBLACK) should be used if any of the

material colour properties is not required.

amb

Ambient colour

diff

Diffuse colour

spec

Specular colour

emit

Emission colour

shine

Shininess %

trans

Translucence 0.0 - 1.0

Gino.SetFacetOffsetMode

(mode)

Sets current facet offset **mode** (Class=FacetOffsetMode) to allow for overlays.

Gino.SetFillMode

(sw)

Switches hardware area-fill on/off (Class=GenerationSwitch).

Gino.SetFontFillStyle

(style)

Sets the style (GFNTFILSTY) for hardware and polygon fonts.

Gino.SetFontForm

(rep)

Represents a software font by displaying the character box only thus speeding up output.

GINO

- Gino.SetFontSpacing**
(space)
Sets character spacing for proportional fonts (Class=FontSpacing).
- Gino.SetFontWeight**
(weight)
Sets the font weight for subsequent character output.
- Gino.SetGraphicsVis**
(vis)
Sets the visibility of graphics drawing area (Class=Visibility).
- Gino.SetHardChars**
Switches on hardware character mode.
- Gino.SetHardCharSize**
(nsize, nhv)
Sets hardware character size index **nsize** and orientation **nhv** (Class=CharacterOrientation).
- Gino.SetInterlineSpace**
(drpfac)
Sets the current inter-line spacing factor for text blocks.
- Gino.SetItalicAngle**
(slant)
Sets italic character angle from the vertical.
- Gino.SetLightSwitch**
(light, sw)
Switch light on/off (Class=Switch).
- Gino.SetLineColour**
(col)
Selects colour index (Class=Colour).
- Gino.SetLineEnd**
(end)
Specifies line end type (Class=LineEnd).
- Gino.SetLineStyle**
(line)
Sets attributes of current line to those of line style index **line**.
- Gino.SetLineVis**
(vis)
Switches line visibility on/off (Class=Visibility).
- Gino.SetLineWidth**
(width)
Specifies line width.
- Gino.SetLineWidthMode**
(sw)
Sets thick line generation mode (Class=GenerationMode).
- Gino.SetLineWidthScaling**
(scale)
Sets a **scale** factor to be used for future settings of line width.
- Gino.SetMask2D**
(limit)
Defines the limits (GLIMIT) of a 2D rectangular mask using non-transformable picture coordinates.
- Gino.SetMaskMode**
(sw)
Switches the state of masking (Class=Switch).

GINO

Gino.SetMaterialColour

(fcol, bcol)

Sets material colours (index or 24bit RGB) for front and back surfaces.

Gino.SetMaterialIndex

(fmat, bmat)

Sets current material index from the table for front (**fmat**) and back (**bmat**) faces.

fmat, bmat=GOFF

colour=line colour

fmat<0

No change

fmat>0

Set front face colour

bmat<0

No change

bmat>0

Set back face colour
and 2-sided lighting

Gino.SetMaxErrorLimit

(nerrs)

Sets maximum number of errors and warnings. **nerrs**=-1 for no limit.

Gino.SetMixedChars

Switches on mixed hardware/software character mode depending on the capabilities of the device.

Gino.SetMousePos

(env, xpos, ypos)

Set current mouse position in pixels relative to **env**

(Class=MouseEnvironment):

Gino.SetPenType

(type)

Specifies pen type or drawing mode for subsequent graphical output (Class=PenType).

Gino.SetPixelDisplayMode

(mode)

Sets pixel area display mode (Class=PixelDisplayMode), enabling a quick display facility for pixels.

Gino.SetPixelReplication

(xrep, yrep)

Specifies the direction and number of pixels to be replicated from the pixel array.

Gino.SetPixelTransform

(ori, xsca, ysca)

Specifies the orientation and scaling transformation of subsequent pixel rectangles.

ori=0

No rotation

ori=1

90 degrees

ori=2

180 degrees

ori=3

270 degrees

Gino.SetPointMode

(sw)

Sets internal point storage mode (Class=PointStorageMode).

Gino.SetPolygonIdent

(ident)

Defines the current polygon identifier - the number assigned to the polygon when it is closed.

Gino.SetPolygonMask

(list, n)

Selects a list of polygon identifiers to define a polygon mask.

Gino.SetPolygonMode

(sw)

Polygon vertex storage mode (Class=Switch).

GINO

Gino.SetPolygonWindow

(list, n)

Selects a list of polygon identifiers to define a polygon window.

Gino.SetRandSeed

(seed)

Sets integer **seed** value for pseudo random number generator gGetRand().

Gino.SetSegHit

(nseg, sens)

Switches sensitivity (Class=Sensitivity) of picture segment **nseg**.

Gino.SetSegMarkColour

(col)

Sets the colour index which is used by the software emulation of gMarkSeg().

Gino.SetSegMode

(sw)

Sets or changes the segmentation mode (Class=GenerationMode).

Gino.SetSegTransform

(nseg, xsca, ysca, ang, xpos, ypos)

Sets the elements of a 2D segment transformation matrix to be applied to segment **nseg**.

Gino.SetSegTransform2D

(nseg, a)

Sets the 2D segment transformation matrix of segment **nseg**.

Gino.SetSegVis

(nseg, vis)

Switches visibility (Class=Visibility) of picture segment nseg.

Gino.SetShadingMode

(mode, [[[culling], blending], winding])

Defines shading **mode** (Class=ShadingMode) and optional culling (Class=CullingMode), blending (Class=Switch) and winding (Class=Direction).

Gino.SetSoftChars

Switches on software character mode.

Gino.SetSplineTension

(tension)

Sets 2D and 3D spline curve tension in range -2.0 to 10.0. Default=0.0.

Gino.SetStrAngle

(angle)

Rotates subsequent character strings by **angle** degrees.

Gino.SetStrExponent

(relcw, relch, posexp, posind)

Sets the relative size and position of string exponents and indices.

relcw, relch	Relative width/height
posexp	Height above base
posind	Height below base

Gino.SetStrJustify

(jus)

Sets the current string justification (Class=Justification).

Gino.SetStrUnderscore

(und)

Switches underscoring of strings (Class=Switch).

Gino.SetSysPriority

(pri)

Set current task priority (Class=TaskPriority).

GINO

Gino.SetTextureCoordGeneration

(mode, [[svec], tvec])

Sets texture coordinate generation mode

(Class=TextureCoordinateGenerationMode) where **svec** and **tvec** are optional transformation vectors (GTEXVEC).

Gino.SetTextureMappingMode

(mode, [blendcol, wraps, wrapt, maxfil, minfil, bordercol])

Set texture mapping **mode** (Class=TextureMappingMode) with optional attributes (see GTEXATT):

blendcol (Class=Colour)

wraps and **wrapt** (Class=TextureWrappingMode)

maxfil and **minfil** (Class=TextureFilter)

bordercol (Class=Colour)

Gino.SetTracerMode

(sw)

Sets subroutine trace facility mode (Class=TracerSwitch).

Gino.SetTransform

(sw)

Sets transforming status (Class=TransformationSwitch).

Gino.SetTransform2D

(a2)

Restores copy of current 2D transformation from (3x2)-element array **a2**.

Gino.SetTransform3D

(a3)

Restores copy of current 3D transformation from (4x4)-element array **a3**.

Gino.SetTransformMode

(sw)

Sets transforming mode (Class=TransformationMode).

Gino.SetViewAxis

(nh, nv)

Specifies which axes are to be horizontal (**nh**) and vertical (**nv**) (Class=Axis).

Gino.SetViewEyeDistance

(dist)

Calculates new eye position at perspective distance **dist**.

Gino.SetViewParams

(vdata)

Resets viewing parameters from stored 15-element array **vdata**.

Gino.SetViewPlaneDistance

(dist)

Defines new view plane position at perspective distance **dist**.

Gino.SetViewport2D

(piclim2, viewlim)

Sets up a 2D viewport mapping from user picture limits (GLIMIT) to viewport in device or paper coordinates (GLIMIT).

Gino.SetViewport3D

(piclim3, viewlim)

Sets up a 3D viewport mapping from user picture cube (GLIMIT3) to viewport in device or paper coordinates (GLIMIT).

Gino.SetViewportClipSwitch

(sw)

Sets viewport clipping switch (Class=Switch).

GINO

Gino.SetViewportMode

(sw)

Switches aspect ratio and position of viewport transformation (Class=ViewportMappingMode).

Gino.SetViewState

(vstate)

Sets viewing parameters from structure GVIEWSTATE.

Gino.SetViewTransformMode

(mode)

Sets hardware/software transformation mode (Class=GenerationSwitch).

Gino.SetViewUpDirection

(dx, dy, dz)

Defines vector (**dx**, **dy**, **dz**) parallel to picture Y axis.

Gino.SetWindow2D

(window2)

Defines new 2D window limits (GLIMIT) in picture coordinates.

Gino.SetWindow3D

(window3)

Defines new 3D window limits (GLIMIT3) in picture coordinates.

Gino.SetWindowMode

(sw)

Sets the windowing status (Class=WindowingSwitch).

ier=Gino.SetWorkingDir

(directory)

Sets current working **directory**.

Gino.SetWorkspaceLimit

(n)

Defines contiguous area of memory used for area-fill and software segment workspace.

Gino.Shear2D

(dep, a)

Shears axis (**dep**) (Class=Axis) by shear factor (**a**). **a** = arctan (shear angle)

Gino.Shear3D

(dir, dep, a)

Shears axis (**dep**)(Class=Axis) in direction (**dir**) (Class=Axis) by shear factor (**a**). **a** = arctan (shear angle).

Gino.Shift2D

(dx, dy)

Shifts the origin by displacements. (**dx**, **dy**).

Gino.Shift3D

(dx, dy, dz)

Shifts origin by displacements. (**dx**, **dy**, **dz**).

Gino.SkipCGMElement

(element)

Can be used to skip a selected **element** when Gino.GetCGMElement() is used. The file must be opened using Gino.OpenCGMFile().

Gino.StartBatchUpdate

Begins batch of modifications to a buffered display file.

Gino.StartPolygon

Starts storage of new polygon vertices, making an implicit call to Gino.SetPolygonMode(Switch.GON) which activates polygon storage.

GINO

Gino.StartTextBlock

(xbeg, ybeg)

Moves the current drawing position to the specified coordinates and indicates the start of a text block.

Gino.SuspendDevice

Suspends output to the currently nominated device i.e. Leave the device in graphics mode.

Gino.SwitchBrokenLineStyle

(switch)

Switch broken line styles (Class=LineMode).

Gino.TimeDelay

(wait)

Suspend GINO for **wait** milliseconds.

Gino.TransformHomogPoint3D

(xs, ys, zs, point, wh)

Transforms the 3D space point (**xs, ys, zs**) into homogeneous coordinates (point, wh) using the current transformation.

Gino.TransformPoint2D

(xs, ys, point2)

Transforms the 2D space point (**xs, ys**) into picture coordinates (GPOINT) using the current transformation.

Gino.TransformPoint3D

(xs, ys, zs, point3)

Transforms the 3-D space point (**xs, ys, zs**) into picture coordinates (GPOINT3) using the current transformation.

col=Gino.TrueCol

(red, green, blue)

Returns 24bit RGB triplet from **red, green** and **blue** colour components.

len=Gino.TrueLen

(string)

Returns length of **string** (without trailing spaces).

Gino.UntransformHomogPoint3D

(xh, yh, zh, wh, point)

Transforms the 3D homogeneous point (**xh, yh, zh, wh**) into space coordinates (GPOINT3) using the inverse of the current transformation.

Gino.UntransformPoint2D

(xp, yp, point2)

Transforms the 2D picture point (**xp, yp**) into space coordinates (GPOINT) using the inverse of the current transformation.

Gino.UntransformPoint3D

(xp, yp, zp, point3)

Transforms the 3D picture point (**xp, yp, zp**) into space coordinates (GPOINT3) using the inverse of the current transformation.

Gino.UpdateView

Updates the viewing transformation matrix from data specified by the viewing routines.

Gino.ViewRotate

(plane, angle, dist)

Rotates line of sight by **angle** about **plane** (Class=ViewPlane) at **dist** from eye.

Gino.ViewShift

(dx, dy, dz)

Shifts line of sight by (**dx, dy, dz**).

GINO

Gino.ViewTurn

(*xr*, *yr*, *zr*, *dx*, *dy*, *dz*, *angle*)

Rotates current view direction in direction (***dx***, ***dy***, ***dz***) from view centre origin (***xr***, ***yr***, ***zr***) about axis by ***angle*** degrees.

Gino.WaitForEvent

(*intype*)

Reads and passes back next discrete event of type (Class=EventType) in queue.

GINOGRAF

Arguments which represent common values are provided in a set of enumerator classes within the GinoGraphics namespace. The following are used in the GINOGRAF library.

Enumerator Classes:

AnnotationAlignment

GTOP	Top justified
GMIDDLE	Middle justified
GBOTTOM	Bottom justified
GDEFAULTPOSITION	Default for specified axis

AnnotationFlag

Grid annotation flag	
GNOGRID	Suppress axis and grid lines
GNOANNOTATION	Suppress annotation
GANNOTATION	Include annotation

AnnotationJustification

GLEFT	Left justified
GCENTRE	Centre justified
GRIGHT	Right justified
GDEFAULTPOSITION	Default for specified axis

AnnotationPosition

GONAXIS	Annotation on axis
GOFFSET	Annotation offset by factor

AnnotationReduction

GNOREDUCE	No reduction
GREDUCE	Reduce size to prevent overlap

AreaChartValue

Area chart display value flag	
GSTART	Display area start values
GFINISH	Display area finish values
GLOWER	Display area lower values
GUPPER	Display area upper values
GWIDTH	Display area widths
GHEIGHT	Display area heights
GAREA	Display area as values

ArrowHead

Type of arrow head	
GSOLID	Filled
GOPEN	Open
GCLOSED	Closed (as triangle)

AxisPosition

Point representing supplied axis position	
GDATAORIGIN	Data origin
GAXISSTART	Start of axis

AxisScaleFormat

Axis scale format	
GNOSCALE	Suppress scale factor
GSCALEPOWEROF10	Displayed as *10 ⁿ
GSCALEZEROES	Displayed as '000 or 0.0'
GSCALEWORD	Displayed in words
GSCALEPREFIX	Displayed in engineering units

AxisScaling

Axis scaling types	
GLINEARTYPE1	Approx limits and intervals
GLINEARTYPE2	Use increment and then round
GLINEARTYPE3	Use supplied increment and end value
GLOG10	Log ₁₀ - approx limits
GDISCRETE	Discrete-exact intervals

AxisScalingLinear

Linear axis/date scaling types	
GLINEARTYPE1	Approx limits and intervals
GLINEARTYPE2	Use increment and then round
GLINEARTYPE3	Use supplied increment and end values

GINOGRAF

AxisStyle	Axis or frame style GNONE GCARDINAL GINTERMEDIATE	Plane frame Ticks marks at cardinal points only Tick marks at intermediate points
BarChartScaling	Bar chart axis scaling types GLINEAR GLOG10 GDISCRETE	
BarChartValue	Bar chart display value flag GSTART GFINISH GLENGTH	Display bar start values Display bar finish values Display bar lengths
DataSwitch	Inclusion of data value in piechart annotation GNODATA GDATA	
DateFormat	Date display format GBRITISH GAMERICAN GLOGICAL	dd/mm/yy mm/dd/yy yy/mm/dd
DateInterval	Date axis interval GDECADE GYEAR GMONTH GWEEK GDAY	
DropStyle	Controls step edges in step charts; GDROPTYPE0 GDROPTYPE1 GDROPTYPE2 GDROPTYPE3	Draws step heights only Links adjacent steps Links steps and draws edges Draws all step edges to base
ErrorBarFlag	Error bar joining line flag GNONE GDRAWLINE	No line between symbols Draw line between symbols
ErrorBarSymbol	Error bar symbol end type GNONE GBAR GARROWSIN GTRIANGLESIN GSOLIDTRIANGLESIN GARROWSOUT GTRIANGLESOUT GSOLIDTRIANGLESOUT	No end Horizontal bar Arrows pointing inwards Triangles point inwards Filled triangles pointing inwards Arrows pointing outwards Triangles point outwards Filled triangles pointing outwards
FitMethod	Line fitting option GLEASTSQUARE	Least squares straight line fit
FitResult	Line fitting result flag GSUCCESS GFAIL	Fit succeeded Fit failed
GraphAxes	Full graph frame axis type GFRAME GAXES	Box frame 2 axes drawn (through data origin)
GraphAxis	GXAXIS GYAXIS	
GraphCharMode	Character generation switch for annotation, labels etc. GGINOMODE GSOFTWARE	Use current GINO mode Use software transformable chars (def.)

GINOGRAF

GraphCoordMode	Graph coordinates system	
	GSPACE	GINO world coordinates
	GGRAPH	Graf coordinates
GraphFrameSource		
	GGINOMODE	Frame defined by GINO
	GGINOGRAF	Frame defined by GRAF routines
GraphScalingMode		
	GDEFAULT	Default
	GEQUALLIMITS	Equal limits for both axes
	GEQUALRANGES	Equal ranges for both axes
	GEQUALGRAPHINTERVALS	Equal sized interval in graph coords
GEQUALSPACEINTERVALS	Equal sized interval in space coords	
GraphType	Data representation type	
	GSYMBOLS	Symbols only
	GSTRAIGHT	Straight lines
	GSTRAIGHTANDSYMBOLS	
	GCUBIC	Cubic spline curve
	GCUBICANDSYMBOLS	
	GSPLINE	Spline curve
	GSPLINEANDSYMBOLS	
	GAKIMA	Akima curve
GAKIMAANDSYMBOLS		
GridStyle	Grid style	
	GNONE	Plane frame
	GTICKS	Frame and tick marks
	GTICKSANDCROSSES	Frame, ticks and grid markers
	GGRIDLINES	Frame and grid lines
	GGRIDLINES2	Frame and intermediate grid lines
HeaderSwitch	Text chart column header switch	
	GNOHEAD	Omit header cell
	GHEAD	Include header cell
LineAngle	Line column angle flag	
	GHORIZONTAL	Horizontal
	GVERTICAL	Vertical
	GLEFTDIAGONAL	Bottom left to top right
	GRIGHTDIAGONAL	Top left to bottom right
MissingValueMode		
	GOFF	
	GEQUALTO	= val1 or val2
	GGREATERTHAN	> val1
	GGREATERTHANOEQUALTO	>= val1
	GLESSTHAN	< val1
	GLESSTHANOEQUALTO	<= val1
	GOUTSIDERANGE	< val1 and > val2
GINSIDERANGE	>= val1 and <= val2	
MultiHistogramType		
	GSTACKED	Data sets stacked in single column
	GCLUSTERED	Data sets displayed as multiple cols.
PercentSwitch	Inclusion of percentage value in piechart annotation	
	GNOPERCENT	
	GPERCENT	
PieChartAnnotation		
	GRADIAL	Display radially
	GINTERNAL	Display within pie segment
	GEXTERNAL	Display externally

GINOGRAF

PieChartBoxType	Piechart internal annotation box type
	GNONE No box
	GFILLED Box filled/masked
	GBOXED Box outline drawn
	GFILLEDANDBOXED Box filled/masked and outline drawn
PlotScaling	Full graph, histogram step, or area chart axis scaling types
	GLINEAR
	GLOG10
PolarAxis	Polar chart axis
	GRAXIS Radial axis
	GTHETAAXIS Theta axis
PolarStyle	Polar plot style
	GTICKS Frame and tick marks
	GTICKSANDRADII Frame and polar radii
	GTICKSANDCIRCLES Fraem, radii and polar circles
ReferenceJustification	Position of string around reference line
	GFARLEFT Beyond lower limit
	GLEFT Left justified at lower limit
	GCENTRE Central
	GRIGHT Right justified
	GFARRIGHT Beyond upper limit
StepChartValue	Step chart display value flag
	GSTART Display step start values
	GFINISH Display step finish values
	GWIDTH Display step widths
	GHEIGHT Display step heights
	GHEIGHTABOVEBASE Display height above base values
SquareWavePosition	Square wave change position
	GCURRENT At current point
	GHALFWAY Mid way between current and next
	GNEXT At next point
TextSwitch	Inclusion of text string in piechart annotation
	GNOTEXT
	GTEXT
ValueDirection	Position of axis annotation
	GCLOCKWISE Clockwise side of axis
	GANTICLOCKWISE Anticlockwise side of axis
	GNOVAL Suppress annotation
ValuePositionHorizontal	
	GOUTSIDELEFT
	GINSIDELEFT
	GCENTRE
	GINSIDERIGHT
	GOUTSIDERIGHT
	GSPECIFIED
ValuePositionVertical	
	GOUTSIDEBOTTOM
	GINSIDEBOTTOM
	GMIDDLE
	GINSIDETOP
	GOUTSIDETOP
	GSPECIFIED
VectorPosition	Vector chart vector position flag
	GTAIL Arrow tail
	GMIDDLE Middle of arrow
	GHEAD Head of arrow

GINOGRAF

GINOGRAF Structure Classes:

Each structure class is listed in alphabetical order followed by a list of its properties each with its value type or enumerator class following in brackets (). All the classes listed below contain a parametrized constructor.

GAREACHART *	Area chart data	
	s (double)	Lower X value
	f (double)	Upper X value
	h1 (double)	Lower Y value
	h2 (double)	Upper Y value
GARCHART *	Bar chart data	
	s (double)	Lower value
	f (double)	Upper value
GERROR *	Error graph data	
	lower (double)	Lower value
	upper (double)	Upper value
GSTEPCHART *	Step chart data	
	s (double)	Lower X value
	f (double)	Upper X value
	h (double)	Height
GVECTOR *	Vector chart data	
	direc (double)	Direction
	stren (double)	Strength
	col (Colour)	Colour

GINOGRAF

Graf.AddAkimaCurve

(npts, points)

Plots a graph using Akima smooth curve to join the **points** (GPOINT), with respect currently defined axes.

Graf.AddAreaChartOutline

(nareas, areas, xory)

Draws a set of data **areas** (GAREACHART) as rectangles on the last defined set of **xory** (Class=GraphAxis) axes.

Graf.AddAreaChartValues

(nareas, areas, sfl, xory)

Annotate **areas** (GAREACHART) with value **sfl** (Class=AreaChartValue) previously drawn by Graf.PlotAreaChart() or Graf.AddAreaChartOutline(). Widths are defined according to **xory** (Class=GraphAxis).

Graf.AddBarChartOutline

(nbars, bars, frac)

Draw **nbars bars** (GBARCHART) using currently defined axes.

Graf.AddBarChartValues

(nbars, bars, frac, sfl)

Annotates a bar chart data set with start, finish or length values as defined by **sfl** (Class=BarChartValue).

Graf.AddErrorBars

(npts, points, errors, type, line, xory)

Draws an error bar at each of the points held in array **points** (GPOINT) using currently defined axes. The bar end points are indicated by the relative distances above and below each point as store in array **errors** (GERROR). The arguments **type** (Class=ErrorBarSymbol) and **line** (Class=ErrorBarFlag) determine the style of the error bars which are drawn perpendicular to the **xory** (Class=GraphAxis) axis.

Graf.AddGraphCurve

(npts, points)

Draws smooth curve using axes already defined, through array **points** (GPOINT).

Graf.AddGraphLine

(x, y)

Draws straight line from current drawing position to point (x, y) in graphical axis coordinates.

Graf.AddGraphMarkers

(npts, points, sym, nspace)

Draws symbol graph using axes already defined, with symbol number **sym** (Class=Marker) at first and every subsequent (**nspace+1**)th point specified in array **points** (GPOINT).

nspace=0

Marks all points

Graf.AddGraphPolyline

(npts, points)

Draws straight-line graph using axes already defined, joining points specified in array **points** (GPOINT).

Graf.AddGraphSpline

(npts, points)

Fits a cubic spline through the points defined in array **points** (GPOINT) with respect to the current axes.

Graf.AddGraphValues

(npts, points, xory)

Annotates a graph with either of the values held in the **points** array (GPOINT) according to **xory** (Class=GraphAxis)

GINOGRAF

Graf.AddGrid

(style1, style2, anx, any)

Draws frame and grid according to following arguments:

style1 (Class=AxisStyle) Style of frame axes
style2 (Class=GridStyle) Style of grid
anx, any (Class=AnnotationFlag) Axis specific annotation/grid

Graf.AddHistogramOutline

(ncols, yarray, frac)

Draws a histogram with heights defined in real array **yarray** with respect to the current axes.

Graf.AddHistogramValues

(ncols, yarray, frac)

Annotates a histogram with height values defined in real array **yarray** with respect to the current axes.

Graf.AddPieChartSegment

(angfro, angto, value, string, fill, line)

Draws a single annotated, filled pie chart segment anticlockwise between the angles **angfro** and **angto** using **fill** (Class=FillStyle) and **line**(Class=LineStyle).

Graf.AddPopulationGraph

(dx, y, npts, points, popmax)

Draws a double line population graph on pre-defined axes. Data is supplied in **points** array (GPOINT) for population **y**.

Graf.AddReferenceLine

(string, xyval, labjus, labclock, hv, xory)

Draws a reference line positioned at **xyval** across a graph and labels it with **string**. The line is draw perpendicular to **xory** (Class=GraphAxis) with its annotation parallel to **hv** (Class=GraphAxis).

labjus (Class=ReferenceJustification)Position of string on line

labclock (Class=Direction) String side of reference line

Graf.AddSquareWave

(npts, points, pos, xory)

Fits a square wave to the points held in array **points** (GPOINT). The square wave interpolation is set by **pos** (Class=SquareWavePosition) and is drawn perpendicular to the **xory** axis (Class=GraphAxis).

Graf.AddStepChartOutline

(nsteps, steps, base, drop, xory)

Draws steps defined by the ranges held in array **steps** (GSTEPCHART) with respect to the current axes system. Step edges are drawn according to **drop** (Class=DropStyle) on the **xory** axis (Class=GraphAxis).

Graf.AddStepChartValues

(nsteps, steps, base, sfl, xory)

Annotates a step chart defined by the array **steps** (GSTEPCHART). The annotation value is determined by **sfl** (Class=StepChartValue) and the axis is **xory** (Class=GraphAxis).

Graf.AddVectors

(nx, ny, vectors, head)

Draws a vector chart using strength, direction and colour held in **nx** by **ny** **vectors** array (GVECTOR). Arrow head form is determined by head (Class=ArrowHead).

Graf.BlockFillAreaChart

(nareas, areas, xory, line)

Block fills and outlines rectangles as defined by data ranges in array **areas** (GAREACHART) using **line** (Class=LineStyle) setting the colour for each area and drawn on the **xory** axis (Class=GraphAxis).

GINOGRAF

Graf.BlockFillBarChart

(nbars, bars, frac, line)

Block fills and outlines **bars** (GBARCHART) of a Bar Chart using **line** (Class=LineStyle) setting the colour for each column.

Graf.BlockFillHistogram

(ncols, yarray, frac, line)

Block fills and outline the columns of a histogram using **line** (Class=LineStyle) setting the colour for each column.

Graf.BlockFillMultiHistogram

(type, rdata, ndim1, ncols, ndata, frac, gap, line, is1, is2)

Block fills and outlines a multi-component histogram of **type** (Class=MultiHistogramType). Data is supplied in two dimensional array **rdata** of dimension **ndim1** by **ndata**. Columns are displayed using data starting at col **is1**, row **is2** using solid colour **line** (Class=LineStyle) with **frac** determining the fraction and **gap** (for clustered form).

Graf.BlockFillStepChart

(nsteps, steps, base, xory, line)

Block fills and outlines rectangles defined by the data ranges held in the array **steps** (GSTEPCHART) using **line** (Class=LineStyle) setting the colour for each area and drawn on the **xory** axis (Class=GraphAxis).

Graf.ConvertDates

(ndates, dates, data)

Converts any character array of **dates** into array of day numbers.

Graf.ConvertDateToGraph

(date, data)

Converts single **date** string into real axis value.

Graf.ConvertGraphToDate

(value, date)

Converts a date real **value** into a character string.

Graf.DefineMissingValues

(mode, val1, val2, xory)

Defines one, two or a range of missing values to be omitted from line, symbol and value graphs according to **mode** (Class=MissingValueMode) on **xory** (Class=GraphAxis).

Graf.DisplayFillColumn

(x, y, nfill, fill, line, header)

Outputs a column of rectangles with an optional header with the column top left positioned at (x, y) using **fill** (Class=FillStyle) and **line** (Class=LineStyle) and **header** label

Graf.DisplayGeneratedColumn

(x, y, nval, vbeg, vend, header)

Outputs a column of values within the range **vbeg** to **vend** with an optional **header** label and positioned at (x, y).

Graf.DisplayLineColumn

(x, y, nline, line, ang, header)

Outputs a column of lines positioned at (x, y) from **line** array (Class=LineStyle) each of style **ang** (Class=LineAngle) with an optional **header** label.

Graf.DisplayMarkerColumn

(x, y, nsym, sym, line, header)

Outputs a column of symbols positioned at (x, y) from arrays **sym** (Class=Marker) and **line** (Class=LineStyle) with an optional **header** label.

GINOGRAF

Graf.DisplayPercentageColumn

(x, y, nval, values, header)

Calculates a set of percentage values and outputs them in a column with an optional **header** label, positioned at (x, y).

Graf.DisplayStringColumn

(x, y, nstr, string, header)

Outputs a column of character strings with an optional **header** label, positioned at (x, y).

Graf.DisplayValueColumn

(x, y, nval, values, header)

Outputs a column of values with an optional **header** label, positioned at (x, y).

Graf.DrawArrow

(xhead, yhead, head, mode)

Draws arrow from current drawing position to point (**xhead**, **yhead**) in **mode** (Class=GraphCoordMode) coordinates with **head** (Class=ArrowHead) style.

Graf.DrawAxes

(tick, tickside, val, xory)

Draws axis **xory** (Class=GraphAxis) with following style:

tick (Class=AxisStyle)

Style of tick marks

tickside (Class=Direction)

Side on which tick marks drawn

val (Class=ValueDirection)

Annotation setting

Graf.DrawAxesLabels

(nstr, string, val, xory)

Displays the labels in **string** array on **val** (Class=Direction) side of **xory** (Class=GraphAxis).

Graf.DrawAxesTitle

(string, yorx, xory, pos1, pos2)

Writes title at **yorx** on axis **xory** (Class=GraphAxis) using characters in **string**. String is justified according to **pos1** (Class=Alignment) relative to **yorx** and **pos2** (Class=Justification) to the axis.

Graf.DrawGraphTitle

(string, xpos, ypos)

Outputs a text **string** as a title withing graph drawing area according to horizontal and vertical justification xpos (Class=Justification) and ypos(Class=Alignment).

Graf.DrawPolarAxes

(tick1, tick2, val, nintp, vendp, rorth)

Draws the **rorth** axis (Class=PolarAxis) for a polar plot (defined by Graf.SetPolarChartAttribs()) with the following style:

tick1 (Class=AxisStyle)

Style of tick marks

tick2 (Class=PolarStyle)

Style of polar chart

val (Class=AnnotationFlag)

Annotation flag

nintp

Number of intervals

vendp

End of axis data range

Graf.EnqAxesAnnotation

(nrpd, npower, nrfigs, asty, xory)

Returns the current numerical annotation settings as defined by the most recent call to Graf.SetAxesAnnotation().

Graf.EnqAxesAttribs

(switch, xy, nstr1, nsk, aoffs, angstr, jstmb, jsler, reduc, xory)

Returns the current axes annotation settings as defined by the most recent call to Graf.SetAxesAttribs().

Graf.EnqAxesPos

(or, origin, axlen, xory)

Returns the position and length of either axis defined by the most recent call to Graf.SetAxesPos() (or the default setting).

GINOGRAF

Graf.EnqAxesScaling

(scale, nints, vbeg, vend, xory)

Returns the scaling parameters, range of values and number of intervals of the X or Y axis prior to and after drawing.

Graf.EnqBlockChartAttribs

(coloff, azim, elev, depth, top, side)

Returns the block fill attribute settings as set by Graf.SetBlockChartAttribs().

offset=Graf.EnqChartSegments

()

Returns current chart segment id. offset as set by Graf.SetChartSegments().

Graf.EnqDateAxesAnnotation

(fdow, fday, fmon, fyear, xory)

Enquires current date axes annotation format as set by Graf.SetDateAxesScaling().

Graf.EnqDateAxesScaling

(scale, dincr, dbeg, dend, xory)

Enquires date scaling set up by Graf.SetDateAxesScaling().

Graf.EnqDateFormat

(inform, insep, ouform, ousep)

Enquires date ordering format set by Graf.SetDateFormat().

Graf.EnqGridSymbol

(sym)

Enquires the grid intersection symbol.

Graf.EnqPieChartAnnotation

(type, txt, per, val, tol)

Returns the current pie chart annotation settings as set by Graf.SetPieChartAnnotation().

Graf.EnqPieChartSettings

(radius, origin, angle)

Returns the current pie chart **radius**, origin (GPOINT) and start **angle**.

Graf.EnqPlotFrame

(flg, limits)

Returns the current **limits** (GLIMIT) of the graph drawing area with their source **flg** (GraphFrameSource).

Graf.EnqTextChartAttribs

(width, height, jslcr, head, line)

Returns the current settings of the text chart characteristics as set by the routine Graf.SetTextChartAttribs().

Graf.EnqValueAttribs

(xpos, ypos, xory, offset, angstr, jstmb, jslcr)

Returns the current settings of the value display attributes as set by the routine Graf.SetValueAttribs().

Graf.EnqVectorAttribs

(pos, vecmin, vecmax, factor)

Enquires vector chart attributes as set by Graf.SetVectorAttribs().

Graf.EnqVectorChartFrame

(limits)

Returns the current **limits** (GLIMIT) of the vector chart drawing area as set up by Graf.SetVectorChartFrame().

Graf.EnqVectorLimits

(smin, smax)

Enquires the minimum and maximum absolute vector limits as set up by Graf.SetVectorLimits().

GINOGRAF

Graf.FillAreaChart

(nareas, areas, xory, fill, line)
Draws a set of data **areas** (GAREACHART) as filled areas on the last defined set of axes using **fill** (FillStyle) and **line** (LineStyle) orientated with widths displayed on **xory** (GraphAxis).

Graf.FillBarChart

(nbars, bars, frac, fill, line)
Fills a set of **bars** (GBARCHART) using axes already defined using **fill** (Class=FillStyle) and **line** (Class=LineStyle).

Graf.FillBelowDataset

(npts, points, xylev, xory, fill, line)
Fills the area between the set of points held in array **points** (GPOINT) and a line perpendicular to the X or Y axis **xory** (Class=GraphAxis) as indicated by **xylev** using **fill** (Class=FillStyle) and **line** (Class=LineStyle).

Graf.FillBetweenDatasets

(n1, xy1, n2, xy2, fill line)
Fills the area between two sets of points on a graph held in the arrays **xy1** and **xy2** (GPOINT) using **fill** (Class=FillStyle) and **line** (Class=LineStyle).

Graf.FillHistogram

(ncols, yarray, frac, fill, line)
Area-fills histogram using axes already defined using **fill** (Class=FillStyle) and **line** (Class=LineStyle). Column heights are specified in array **yarray**.

Graf.FillMultiHistogram

(type, rdata, ndim1, ncols, ndata, frac, gap, fill, line, is1, is2)
Fills a multi-component histogram of **type** (Class=MultiHistogramType). Data is supplied in two dimensional array **rdata** of dimension **ndim1** by **ndata**. Columns are displayed using data starting at col **is1**, row **is2** using **fill** (Class=FillStyle) and **line** (Class=LineStyle) with **frac** determining the fraction and **gap** (for clustered form).

Graf.FillStepChart

(nsteps, steps, base, xory, fill, line)
Draws filled rectangles defined by the data ranges held in array **steps** (GSTEPCHART) and the **xory** (Class=GraphAxis) axis with respect to the current axes system using **fill** (Class=FillStyle) and **line** (Class=LineStyle).

Graf.MoveToGraphPoint

(x, y)
Moves to point (**x**, **y**) in graphical axis coordinates.

Graf.PlotAreaChart

(nareas, areas, scx, scy)
Draws a frame to fit the available drawing area and plots an area chart within it from the array **areas** (GAREACHART) using **scx** and **scy** (Class=PlotScaling) for the scale on each axes.

Graf.PlotBarChart

(nbars, bars, frac, scx, scy, vbeg, vend)
Draws bar chart sized to fit current drawing area or window. Start and finish values of bars are specified in array **bars** (GBARCHART) using **scx** and **scy** (Class=BarChartScaling) for the scale on each axes. The values **vbeg** and **vend** represent the range of values on the discrete axis.

Graf.PlotGraph

(npts, points, scx, scy, style, axis)
General purpose drawing routine for line, curve or symbol plots within a frame or simple axes using **scx** and **scy** (Class=PlotScaling) for the scale on each axes. The graph style is determined by **style** (Class=GraphType) and **axis** (Class=GraphAxes).

GINOGRAF

Graf.PlotHistogram

(ncols, yarray, frac, scy, vbegin, vend)
Draws histogram sized to fit current drawing area or window. Column heights are specified in real array **yarray** using **scx** (Class=PlotScaling) for the scale on the Y axis. The values **vbegin** and **vend** represent the range of values on the discrete (X) axis.

Graf.PlotPieChart

(nsegs, values, labels, fill, line)
Draws a complete annotated filled pie chart fitting the available current window or current pie chart frame as defined by Graf.SetPieChartFrame(). Data is supplied in the arrays **values** and **labels**, **fill** (Class=FillStyle) and **line** (Class=LineStyle).

Graf.PlotStepChart

(nsteps, steps, base, scx, scy, drop)
Draws a frame to fit the current drawing area and plots a step chart within it. The steps are drawn as columns with widths determined by the values in **steps** (GSTEPCHART) using **scx** and **scy** (Class=PlotScale) to scale the axes and **drop** (Class=DropStyle) the step form.

Graf.PlotXYPolarChart

(npts, points, style)
General purpose polar plot routine for line, curve or symbol plots on polar axes using style (Class=GraphType) to set the form.

Graf.RestoreAxesSettings

Resets default settings for axis positioning, scaling and annotation for both axes.

Graf.RestoreBlockChartAttribs

Restores the default block chart attribute.

Graf.RestorePieChartSettings

Restores the default pie chart frame, in which the size and position of the pie chart are calculated to fit the available drawing area or current window.

Graf.RestoreVectorSettings

Resets the clipping, positioning and scaling attributes of vector chart arrows to their default state.

Graf.ReturnLineCoeffs

(type, npts, points, ncoef, coeffs, nmax, er)
Fits a straight line of **type** (Class=FitMethod) to a set of data **points** (GPOINT) and returns their coefficients in the array **coeffs**. Total number of possible coefficients is returned in **nmax** with **er** (Class=FitResult) giving the success state. No graphics output is generated.

Graf.SetAxesAnnotation

(ndp, npower, asty, xory)
Sets numbers of decimal places - **ndp**, scale factor - **npower** and scale display type - **asty** (Class=AxisScaleFormat) for annotation of the **xory** (Class=GraphAxis) graph axes.

Graf.SetAxesAttribs

(swi, xy, nstart, nskip, aoff, angstr, jstmb, jsclr, reduc, xory)
Sets general characteristics for text and numerical axis annotation on **xory** (Class=GraphAxis) graph axes.
swi (Class=AnnotationPosition)
nstart Starting tick number (1)
nskip Labels to be skipped (0)
aoff Display offset (0.0)
Angstr Annotation string angle (0.0)
jstmb (Class=AnnotationAlignment)
jsclr (Class=AnnotationJustification)
reduc (Class=AnnotationReduction)

GINOGRAF

Graf.SetAxesPos

(or, xor, yor, axlen, xory)

Defines position (**xor**, **yor**) in terms of **or** (Class=AxisPosition) and length **axlen** of axis **xory** (Class=GraphAxis).

Graf.SetAxesScaling

(scale, nints, vbeg, vend, xory)

Defines scaling type - **scale** (Class=AxisScaling), number of intervals (**nints**) and range (**vbeg**, **vend**) of axis **xory** (Class=GraphAxis).

Graf.SetBlockChartAttribs

(coloff, azim, elev, depth, top, side)

Sets the chart block fill attributes.

coloff	Colour index offset [20]
azim	Extrusion azimuth [30]
elev	Extrusion elevation [30]
depth	Col. depth factor [1.0]
top	Top colour reduc [0.67]
side	Side colour reduc [0.33]

Graf.SetChartSegments

(offset)

Sets a segment id. offset for the generation of chart components.

Graf.SetCurveEndConds

(fin, cosfin, sinfin, xfin, yfin)

Sets the end conditions - **fin** (Class=EndCondition) of the next GINOGRAF curve drawing routine in terms of angles (**cosfin**, **sinfin**) or an extra point (**xfin**, **yfin**).

Graf.SetCurveStartConds

(beg, cosbeg, sinbeg, xbeg, ybeg)

Sets the start conditions - **beg** (Class=EndCondition) of the next GINOGRAF curve drawing routine in terms of angles (**cosbeg**, **sinbeg**) or an extra point (**xbeg**, **ybeg**).

Graf.SetDateAxesAnnotation

(fdow, fday, fmon, fyear, xory)

Defines output format for date axes annotation on **xory** (Class=GraphAxis) .

<0	Alphanumeric
GNONE	Not present
>0	Numeric

Graf.SetDateAxesScaling

(scale, dincr, dbeg, dend, xory)

Defines date axis scaling type (Class=AxisScalingLinear) in terms of increment - **dincr** (Class=DateInterval) and date range (**dbeg** to **dend**) defined as strings for **xory** (Class=GraphAxis).

Graf.SetDateFormat

(inform, insep, ouform, ousep)

Sets input and output format for numeric date strings (Class=DateFormat) and their string separator.

Graf.SetGraphCharMode

(sw)

Sets GINOGRAF character display mode (Class=GraphCharMode).

Graf.SetGraphScaling

(mode)

Sets scaling mode (Class=GraphScalingMode) for complete graph routines.

Graf.SetGridMarker

(sym)

Sets grid intersection symbol (Class=Marker).

GINOGRAF

Graf.SetPieChartAnnotation

(type, txt, per, val, tol)

Sets the pie chart annotation **type** (Class=PieChartAnnotation) and annotation flags for the complete pie chart routine.

txt (Class=TextSwitch)

per (Class=PercentSwitch)

val (Class=DataSwitch)

tol=Minimum % annotation level

Graf.SetPieChartBoundSwitch

(switch)

Determines whether pie chart boundaries are to be drawn (Class=Switch).

Graf.SetPieChartBoxType

(type, fill, line)

Controls the **type** (Class=PieChartBoxType) of the internal annotation boxes for the complete pie chart routine together with their **fill** (Class=FillStyle) and **line** (Class=LineStyle).

Graf.SetPieChartExplosion

(num, list, factor)

Explodes pie chart. Segment numbers to be exploded are specified in array **list** of **num** elements. Explosion factors (in range 0.0 to 1.0) are specified in array **factor** of **num** elements. If **num** = 0 cancels previously-set explosion.

Graf.SetPieChartFrame

(radius, xcen, ycen)

Redefines pie chart radius as **radius** and centre position as (**xcen**, **ycen**).

Graf.SetPieChartStartAngle

(angle)

Sets start angle of first pie chart segment to **angle**.

Graf.SetPlotFrame

(limits)

Defines a boundary for the GINOGRAPH drawing area (GLIMIT) when using complete graph or chart routines.

Graf.SetPolarChartAttribs

(xorp, yorp, radlen, scale)

Defines the position (**xorp**, **yorp**) in user space coordinates, its radius **radlen** and axis **scale** (Class=AxisScalingLinear) of a polar plot.

Graf.SetTextChartAttribs

(width, height, jsclr, head, line)

Defines the **width** and **height** of a text chart column, its justification-**jsclr** (Class=Justification), header switch - **head** (Class=HeaderSwitch) and frame **line** style (Class=LineStyle).

Graf.SetValueAttribs

(xpos, ypos, xory, xoff, yoff, angstr, jstmb, jsclr)

Sets the attributes for the display of data values output by the Graf.AddxxxValue() routines.

xpos (Class=ValuePositionHorizontal)

ypos (Class=ValuePositionVertical)

xory Specific value position

xoff, yoff Offset from control point

angstr String angle

jstmb (Class=Alignment)

jsclr (Class=Justification)

Graf.SetValueTags

(prefix, suffix)

Sets a **prefix** or **suffix** string to values output with the Graf.AddxxxValue() routines.

GINOGRAF

Graf.SetVectorAttribs

(position, vecmin, vecmax, factor)

Sets of the **position** (Class=VectorPosition) and scaling attributes for vector charts drawn with Graf.AddVectors().

Graf.SetVectorChartFrame

(limits)

Sets up an drawing area (GLIMIT) for vector charts.

Graf.SetVectorLimits

(smin, smax)

Defines the minimum and maximum absolute vector strength that may be drawn by subsequent calls to Graf.AddVectors().

Graf.TransformGraphPoint

(xgr, ygr, point)

Converts graphical axis coordinates (**xgr, ygr**) to user space coordinates (GPOINT).

Graf.TransformSpacePoint

(xsp, ysp, point)

Converts user space coordinates (**xsp, ysp**) to graphical axis coordinates (GPOINT).

GINOSURF

Arguments which represent common values are provided in a set of enumerator classes within the GinoGraphics namespace. The following are used in the GINOSURF library.

Enumerator Classes:

ContourAnnotationDirection	GDEFAULT	Default
	GUPRIGHT	Annotation drawn upright
	GUPSLOPE	Annotation drawn up surface slope
ContourDataSet	Contouring data set when 4D data available	
	GFIRST	Contour using Z_1 data set
	GSECOND	Contour using Z_2 data set
ContourMapFrameType	GFRAMEANDTICKS	Rectangular frame, annotation and ticks
	GFRAME	Rectangular frame only
	GNOFRAME	Suppress frame
ContourSwitch	Countour drawing switch	
	GSTRAIGHT	Use straight lines
	GSMOOTH	Draw smooth curves
CrossSectionAxis	GXAXIS	X axis displayed
	GYAXIS	Y axis displayed
	GHDIST	Horizontal distance displayed
CrossSectionFrameType	GSECTIONANDAXES	Full cross-section and axes (default)
	GOUTLINEANDAXES	Cross-section outline and axes
	GSECTIONONLY	Cross-section only
	GOUTLINEONLY	Cross-section outline only
CrossSectionProjection	GXDIRECTION	X differences displayed
	GYDIRECTION	Y differences displayed
	GDISTANCE	Horizontal distance displayed (default)
DataPointAnnotation	GNONE	Supress symbol and data value
	GSYMBOLS	Draw symbols at data points
	GSYMBOLSANDVALUES	Draw symbols and height values
DataSupplied	Flag indicating data supplied to routine	
	GXYONLY	X and Y data values
	GXYZONLY	X, Y and Z (height) values
	GALLVALUES	X, Y, Z and derivatives
DerivativeInterpolationMethod	GLINEAR	Linear approximation
	GLEASTSQUARE	Least squares approximation (default)
FrameAxesStyle	GNONE	Omit axis
	GAXISLINE	Draw axis line only
	GTICKSONLY	Draw axis and tick marks only
	GVALUESANDTICKS	Draw complete axis
GradientDisplay	GOFF	Suppress gradient field arrows
	GON	Display arrows at triangle centroids
	GVERTEX	Display arrows at triangle vertices
	GVERTEXSCALED	Display scaled arrows at triangle vertices
	GCENTROID	Display arrows at triangle centroids
	GCENTROIDSCALED	Display scaled arrows at triangle centroid
GridType	Data grid/workspace type	
	GTRIANGULATED	
	GGRIDDED	

GINOSURF

HeightInterpolationMethod		
	GLINEAR	One sided linear interpolation
	GLEASTSQUARE	Nearest neighbour weighted least sq.
	GWEIGHTAVERAGE	Weighted average
InactiveLevel	Level at which inactive triangles are drawn	
	GBASELEVEL	
	GSURFACELEVEL	
IntersectionAction	Action performed when data intersections found	
	GADDPPOINTS	Add additional data points
	GSWAPDIAGONALS	Swap network diagonals if possible
NetworkInterpolationMethod		
	GBILINEAR	Bi-linear approximation (default)
	GLEASTSQUARE	Nearest neighbour weighted least sq.
	GCLOUGHTOUCHER	Clough_Toucher C1 cubic
RandomInterpolationMethod		
	GDEFAULT	Weighted least squares
	GLEASTSQUARE	Least squares
	GWEIGHTAVERAGE	Weighted average
Rounding	Data limit rounding flag	
	GROUND	Rounded limits are used (default)
	GEXACT	Exact data limits are used
SetOperation	Set operation when adding network region	
	GINTERSECTION	Triangulate intersection
	GEXTERIOR	Triangulate exterior region
	GUNION	Triangulate union (default)
SurfAxis		
	GDEFAULT	Restore default axes annotation format
	GXAXIS	X axis
	GYAXIS	Y axis
	GZAXIS	Z/height axis
	GALLAXES	All axes
SurfCharMode	Character generation switch for annotation, labels etc.	
	GINOMODE	Use current GINO mode
	GSOFTWARE	Use software transformable chars (def.)
SurfLabel	Label	
	GDEFAULT	Restore default labels
	GBOTHOFF	Suppress labels and power factors
	GPOWEROFF	Suppress power factors
	GXAXIS	Define X axis label
	GYAXIS	Define Y axis label
	GZAXIS	Define Z/Height axis label
	GCONTOURHEIGHT	Define contour height label
	GCONTOURSTEP	Define contour step label
	GXLIMITSSHOWN	Define cross-section X limits label
	GYLIMITSSHOWN	Define cross-section Y limits label
	GORIZONTALDIST	Define horizontal distance label
SurfAxesScaling	Axis scaling flag	
	GEQUAL	Equal scaling on both axes (default)
	GARIABLE	Variable scaling
SurfaceBaseFlag	Surface network flag	
	GOFF	Suppress traingualtion network
	GON	Draw network on surface
	GBASE	Draw network at base level

GINOSURF

SurfaceAxesStyle	GNONE	No axis/suppression
	GAXIS	Axis drawn
	GGRID	Grid drawn
	GAXISANDGRID	Axis and grid drawn
	GLOWEST	Suppress lowest data value
	GHIGHEST	Suppress highest data value
	GLOWESTANDHIGHEST	Suppress lowest and highest values
	GALL	Suppress all values
Surface	Surface normal direction mode	
	GUPPER	Normals point away from upper surface
	GUNDER	Normals point away from under surface
SurfaceAxesLabelMode	GDEFAULT	Label at bottom corner of plot area
	GADJACENT	Labels drawn adjacent to axes
SurfaceBaseStyle	GNOLINES	No lines
	GHORIZONTAL	Horizontal lines
	GVERTICAL	Vertical lines
	GHEIGHTRELATED	Vertical lines drawn according to height
	GLEVELS	Filled contour levels drawn
SurfaceBaseType	GBASEANDAXES	Upper surface, base and axes (default)
	GNOAXES	Upper surface and base but no axes
	GBOTHSURFACES	Both surfaces without base or axes
	GUPPERSURFACE	Upper surface only
	GUNDERSURFACE	Under surface only
Switch	GON	Key is displayed
	GOFF	No Key is displayed
TextureLevel	Level at which to texture inactive surface triangles	
	GALL	Texture all inactive triangles
	GBASELEVEL	Texture inactive triangles at base level
	GSURFACELEVEL	Texture inactive triangles at surface level
TriangleSet	Surface texture set	
	GALL	Texture applied to all triangles
	GACTIVE	Texture applied to active triangles
	GINACTIVE	Texture applied to inactive triangles
TrimMode	Boundary trimming method	
	GHEIGHTBASE	Use height to base ratio
	GANGLE	Use defined (internal) angle
WeightingFactor	Interpolation weighting factor when using least squares	
	GINVERSESQUARE	Inverse square
	GINVERSE	Inverse weighting
	GDISTANCESQUARE	Distance square weighting
	GSIMPLEINVERSESQUARE	Simple inverse square weighting
WorkspaceOperation	GINITIALIZE	(Re)Initialize workspace
	GRESTORE	Restore workspace

GINOSURF Structure Classes:

Each structure class is listed in alphabetical order followed by a list of its properties each with its value type or enumerator class following in brackets (). Only those classes with an asterisk (*) contain a parametrized constructor as in the other cases there is no need.

GFILL *	Surface element fill style:	
	fill (FillStyle)	Fill style
	line (LineStyle)	Line style (1-256)

GINOSURF

GPOINT4 *	4D coordinate x (double) y (double) z (double) z2 (double)	X position Y position Z position Z2 position
GPOLYGON4	4D polygon component nvert (int) verts (GPOINT4[])	Number of vertices Pointer to vertices
GPRJAXI *	Projected surface axes style and suppression: x1 (SurfaceAxesStyle) x2 (SurfaceAxesStyle) y1 (SurfaceAxesStyle) y2 (SurfaceAxesStyle) z1 (SurfaceAxesStyle) z2 (SurfaceAxesStyle)	Style or value suppression on x1 axis Style or value suppression on x2 axis Style or value suppression on y1 axis Style or value suppression on y2 axis Style or value suppression on z1 axis Style or value suppression on z2 axis
GSURF3D *	GINOSURF 3D state: hard (Switch) proj (Switch)	3D hardware usage switch (facets) GINO viewing state usage switch
GTRiangle *	Triangle vertices nv1 (int) nv2 (int) nv3 (int)	First vertex Second vertex Third vertex
GWORKSPACE	GINOSURF workspace: nw (int) ww (double[])	Workspace size Pointer to workspace

GINOSURF

Surf.ActivateTriangles

(ntri, triangles, nttt)

Activates set of inactive triangles in triangulation network **nttt**.

Surf.AddBreakLines

(nopt, nint, npol, polylines3, derivs , nttt, [swap])

Adds a set of break polylines into a triangulation network **nttt**. The argument **nopt** (Class=DataSupplied) indicates the data supplied in **polylines3** (GPOLYGON3) and **derivs** (GPOLYGON), with **nint** (Class=HeightInterpolationMethod) specifying the interpolation option where heights need to be calculated internally. An optional argument **swap** (Class=IntersectionAction) affects

Surf.AddBreakLines4D

(nopt, nint, npol, polylines4, derivs , derivs2, nttt, [swap])

Adds a set of break polylines into a 4D triangulation network **nttt**. The argument **nopt** (Class=DataSupplied) indicates the data supplied in **polylines4** (GPOLYGON4) and **derivs** (GPOLYGON), with **nint** (Class=HeightInterpolationMethod) specifying the interpolation option where heights need to be calculated internally.

Surf.AddContourMapGrid

(limits, xstep, ystep)

Superimposes a rectangular grid (GLIMIT) on a contour map with grid intervals (**xstep**, **ystep**).

Surf.AddContourPolyline

(n, points)

Draws a polyline on top of a previously drawn contour map.

Surf.AddContourPolylineSet

(npol, polylines, nttt)

Draws a polyline set on triangulated contour map. Polylines may be of type GPOLYGON, GPOLYGON3 or GPOLYGON4 but only X and Y coords are used.

Surf.AddCrossSectionGrid

(limits, hstep, vstep)

Draws a rectangular grid on to of a cross-section.

Surf.AddCrossSectionPolyline

(npts, points)

Draws a polyline on top of a cross-section.

Surf.AddGriddedContour

(ch, smooth, nwww)

Superimposes a single additional contour at level (**ch**) on a pre-defined contour map. Contour is drawn in straight line segments or as a smooth curve according to **smooth** (Class=ContourSwitch).

Surf.AddGriddedContourFill

(zll zhh, index, visible, smooth, nwww)

Fills the region between contour levels **zll**, **zhh** in the fill style defined by **index** (GFILL). Contours either side are optionally drawn **visible** (Class=Visibility) in straight line segments or as a smooth curve according to **smooth** (Class=ContourSwitch).

Surf.AddNetworkGrid

(type, limits, numx, numy, nttt)

Adds a **type** (Class=GridType) of grid of points to an existing triangulation over the specified **limits** and with **numx** by **numy** divisions.

Surf.AddNetworkPoint

(nopt, point, derivative, nttt)

Adds a **point** (GPOINT3) and **derivative** (GPOINT) (with **nopt** (Class=DataSupplied) indicating the relevant data) to the triangular network.

GINOSURF

Surf.AddNetworkPoint4D

(nopt, point4, deriv1, deriv2, nttt)

Adds a 4D **point** (GPOINT4) and **derivatives** (GPOINT) (with **nopt** (Class=DataSupplied) indicating the relevant data) to the triangular network.

Surf.AddNetworkRegion

(nopt, npts, points, derivs, nreg, nttt, [swap])

Specifies a data set region which undergoes a set operation with an existing triangulation.

nopt (Class=DataSupplied) Relevant data in points (GPOINT3) and derivs

nreg (Class=SetOperation)

swap (Class=IntersectionAction) Optional

Surf.AddNetworkRegion4D

(nopt, npts, points4, derivs, derivs2, nreg, nttt, [swap])

Specifies a 4D data set region which undergoes a set operation with an existing triangulation.

nopt (Class=DataSupplied) Relevant data in points (GPOINT4), derivs and derivs2

nreg (Class=SetOperation)

swap (Class=IntersectionAction) Optional

Surf.AddSurfaceNetwork

(net)

Draws a triangular **net** (Class=SurfaceBaseFlag) on a surface.

Surf.AddSurfacePolyline

(n, points3)

Draws a polyline on top of a previously drawn perspective projection.

Surf.AddSurfacePolylineHidden

(n, points2, line, nwww)

Draws multiple polylines (GPOINT) on top of a projected surface.

Surf.AddTriangulatedContour

(ht, smooth, nttt)

Superimposes a single additional contour at level (**ht**) on a pre-defined contour map. Contour is drawn in straight line segments or as a smooth curve according to **smooth** (Class=ContourSwitch).

Surf.AddTriangulatedContourFill

(zll, zhh, index, visible, smooth, nttt)

Fills the region between contour levels **zll**, **zhh** in the fill style defined by **index** (GFILL). Contours either side are optionally drawn **visible** (Class=Visibility) in straight line segments or as a smooth curves according to **smooth** (Class=ContourSwitch).

Surf.CloseSurf

Terminates GINOSURF.

Surf.DeactivateTriangles

(ntri, triangles, nttt)

Deactivates a set of triangles in triangulation network.

Surf.DefinePerspProjection

(limits, rad, theta, phi, nwww)

Sets up view point and **limits** (GLIMIT) for subsequent perspective surface drawing, contour map projections and drawing of perspective axes using workspace **nwww**.

Surf.DrawContourSurface3D

(limits, numx, numy, z1, rad, theta, phi, ncont, key, nwww)

Draws **ncont** contours on a projected surface using height data in **z1**.

Surf.DrawContourSurface4D

(limits, numx, numy, z1, z2, rad, theta, phi, ncont, key, nwww)

Draws **ncont** '4D' contours on a projected surface using height data in **z1** and 4D data in **z2**.

GINOSURF

Surf.DrawGriddedContourMap

(limits, numx, numy, az, ncont, smooth, nwww)

Draws an annotated frame and contour map from gridded data passed in **az**, scaled to fit the current drawing area. Contours are drawn in straight line segments or as a smooth curves according to **smooth** (Class=ContourSwitch).

Surf.DrawGriddedCrossSection

(xstart, ystart, xfin, yfin, nwww)

Draws a cross-section from (**xstart**, **ystart**) to (**xfin**, **yfin**) after drawing a contour map or surface projection.

Surf.DrawLineStyleSurface

(limits, numx, numy, az, rad, theta, phi, ncont, index, key, nwww)

Draws a height-related wire-frame perspective projection from regular grid data **az** with optional **key** (Class=Switch).

Surf.DrawNetwork

(type, symb, nttt, [inactive, trinum, nodenum, gradient])

Draws the triangulated data network stored in the triangulated workspace **nttt** using **type** (Class=DataPointAnnotation), **symb** (Class=Marker) and the optional arguments to determine the associated information displayed with it.

inactive (Class=Switch)

Inactive triangles

trinum (Class=Switch)

Triangle numbers

nodenum (Class=Switch)

Node numbers

gradient (Class=GradientDisplay)

Gradient field arrows

Surf.DrawPolylineCrossSection

(npts, points, line, nwww)

Draws a multiple cross-section on a polyline base passed in **points** (GPOINT) using **line** styles.

Surf.DrawSurfaceAxes

(nwww)

Draws the axes for a perspective projection defined by the routine Surf.SetPerspProjection().

Surf.DrawTriangulatedContourMap

(limits, ncont, smooth, nttt)

Draws an annotated frame and contour map from triangulated data, scaled to fit the current drawing area. Contours are drawn in straight line segments or as a smooth curves according to **smooth** (Class=ContourSwitch).

Surf.DrawWireFrameSurface

(limits, numx, numy, az, rad, theta, phi, nwww)

Draws a wire-frame perspective projection from regular grid data **az**.

Surf.EnqContourAxesLimits

(limits)

Returns current contour map axes **limits** (GLIMIT).

Surf.EnqContourLevels

(ncont, zhght, [gsLineStyle])

Returns current (**ncont**) contour levels (**zhght**) and optionally their linestyles.

Surf.EnqCrossSectionAxesLimits

(naxis, limits)

Returns current cross-section **axis** (Class=CrossSectionAxis) and its **limits** (GLIMIT).

Surf.EnqCrossSectionFrame

(axis, axmin, axmax)

Returns current cross-section X **axis** (Class=CrossSectionAxis) and its range

GINOSURF

Surf.EnqSurfaceAxesLimits

(limits)

Returns current surface plot axes **limits** (GLIMIT3).

Surf.ExportTriangulationNetwork

(mode, npts, points, derivs, ntri, triangles, nttt)

Outputs the triangulation data structure **nttt** according to mode (Class=TriangleSet). Points (**npts**) returned in **points** (GPOINT3), derivatives returned in **derivs** (GPOINT) and triangles (**ntri**) returned in **triangles** (GTRIANGLE).

Surf.ExportTriangulationNetwork4D

(mode, npts, points4, derivs, derivs2, ntri, triangles, nttt)

Outputs the triangulation data structure **nttt** according to mode (Class=TriangleSet). Points (**npts**) returned in **points4** (GPOINT4), derivatives returned in **derivs** and **derivs2** (GPOINT) and triangles (**ntri**) returned in **triangles** (GTRIANGLE).

Surf.FillContourSurface3D

(limits, numx, numy, z1, rad, theta, phi, ncont, index, key, nwww)

Displays a filled contour map with **ncont** contours and optional **key** (Class=Switch) on a projected surface using grided height data passed in **z1**.

Surf.FillContourSurface4D

(limits, numx, numy, z1, z2, rad, theta, phi, ncont, index, key, nwww)

Displays a 4D filled contour map with **ncont** contours and optional **key** (Class=Switch) on a projected surface using two sets of grided height data passed in **z1** and **z2**.

Surf.FillGriddedContourMap

(limits, numx, numy, az, ncont, smooth, index, key, nwww)

Draws filled contour map, frame and optional **key** (Switch) from gridded data in **az**, scaled to fit the current drawing area. The contours and optional **key** are filled using fill styles in **index** (GFILL).

Surf.FillProjectedPolygon

(npts, points, height, index)

Fills a polygonal area defined by **points** (GPOINT) representing the boundary of a contour plot in fill style **index** (GFILL).

Surf.FillSurface

(limits, numx, numy, az, rad, theta, phi, index, nwww)

Draws a filled perspective projection from regular grid, **az** using fill styles in **index** (GFILL).

Surf.FillTriangulatedContourMap

(limits, ncont, smooth, index, key, nttt)

Draws an annotated frame, filled contour map and optional **key** (Class=Switch) from triangulated data, scaled to fit the current drawing area using fill styles in **index** (GFILL). Contours are drawn in straight line segments or as a smooth curves according to **smooth** (Class=ContourSwitch).

Surf.FillTriangulatedContourSurf3D

(limits, rad, theta, phi, ncont, index, key, nttt)

Draws a filled contour map and optional **key** (Class=Switch) on the surface defined by a triangulation using fill styles in **index** (GFILL).

Surf.FillTriangulatedContourSurf4D

(limits, rad, theta, phi, ncont, index, key, nttt)

Draws a 4D filled contour map and optional **key** (Class=Switch) on the surface defined by a 4D triangulation using fill styles in **index** (GFILL).

Surf.FillTriangulatedSurface

(limits, rad, theta, phi, numtri, index, nttt)

Draws a filled perspective projection of the surface defined by a triangulation using fill styles in **index** (GFILL).

GINOSURF

Surf.FreeWorkspace

(nwww)

Frees up memory used by internal workspace **nwww**.

Surf.GenerateConstrainedNetwork

(noppts, numpts, points, noppol, npoly, polylines, nopreg, nreg, noptrg, regions, numint, nttt, [gsDerivs1, gsDerivs2, gsDerivs3], [gsSwap])
Computes a triangulated network using combinations of supplied (**numpts**) data points in **points** (GPOINT3), (**npoly**) breaklines in **polylines** (GPOLYGON3) and (**nreg**) **regions** (GPOLYGON3) each with optional derivatives.

Surf.GenerateConstrainedNetwork4D

(noppts, numpts, points4, noppol, npoly, polylines4, nopreg, nreg, noptrg, regions4, numint, nttt, [gsDerivs1, gsDerivs2, gsDerivs3, gsDerivs4, gsDerivs5, gsDerivs6], [gsSwap])
Computes a 4D triangulated network using combinations of supplied (**numpts**) data points in **points4** (GPOINT4), (**npoly**) breaklines in **polylines4** (GPOLYGON4) and (**nreg**) **regions4** (GPOLYGON4) each with optional derivatives for both Z and Z2 coordinates.

Surf.GenerateNetworkGrid

(limits, numx, numy, az, nttt, nwww)

Computes a rectangular grid of points (**az**) from a triangulation previously created by Surf.GenerateRandomGrid().

Surf.GenerateRandomGrid

(npts, points, limits, numx, numy, az, nwww)

Uses the randomly spaced **points** (GPOINT) and interpolates from them to form a regular grid (**az**).

Surf.GenerateRandomNetwork

(maxpts, npts, points, nttt, [derivatives])

Forms a triangulated network using the randomly spaced **points** (GPOINT3) and optional **derivatives** (GPOINT). **Maxpts** should be set to the maximum number of points the network is ever likely to contain.

Surf.GenerateRandomNetwork4D

(maxpts, npts, points4, nttt, [deriv1, deriv2])

Forms a 4D triangulated network using the randomly spaced **points** (GPOINT4) and optional derivatives (GPOINT). **Maxpts** should be set to the maximum number of points the network is ever likely to contain.

Surf.ImportTriangulationNetwork

(maxpts, npts, points, ntri, triangles, nttt, [derivatives])

Inputs a triangulation data structure consisting of **npts points** (GPOINT3), **ntri triangles** (GTRIANGLE) and optionally **npts derivatives** (GPOINT). **Maxpts** should be set to the maximum number of points the network is ever likely to contain.

Surf.ImportTriangulationNetwork4D

(maxpts, npts, points4, ntri, triangles, nttt, [derivs, deriv2])

Inputs a triangulation data structure consisting of **npts points** (GPOINT3), **ntri triangles** (GTRIANGLE) and optionally **npts derivatives** (GPOINT). **Maxpts** should be set to the maximum number of points the network is ever likely to contain.

Surf.InitializeWorkspace

(mode, type, n, nwww)

Reinstates or initialises a GINOSURF workspace of size **n**.

mode =GINITIALIZE	Initialize workspace
mode =GRESTORE	Restore workspace
type =GGRIDDED	Gridded data
type =GTRIANGULATED	Triangulated data

Surf.OpenSurf

Initializes and resets all GINOSURF defaults.

GINOSURF

Surf.PolylineCutandFill

(*npts*, *points*, *arecut*, *arefil*, *nwww*)

Calculates areas of cut and fill between current surface and polyline passed in **points** (GPOINT3).

Surf.PreserveSurfaceScaling

(*switch*)

Saves the scaling parameters of a surface projection for reuse (Class=Switch).

Surf.RemoveNetworkPoint

(*xp*, *yp*, *nttt*)

Removes point from triangulation network.

Surf.RestoreWorkspace

(*filename*, *nwww*)

Read GINOSURF workspace from disk file.

Surf.ReturnContourDataset

(*chght*, *smooth*, *visible*, *npts*, *points*, *numseg*, *lenseg*, *nwww*)

Returns vertices used to draw contour of height **chght** in array **points** (GPOINT). Array **lenseg** contains number of points in **numseg** line segments.

Surf.ReturnContourSpotHeight

(*xint*, *yint*, *zint*, *nwww*)

Computes the height (**zint**) of the point (**xint**, **yint**) on contour map.

Surf.ReturnCrossSectionArea

(*hbase*, *arepos*, *areneg*, *nwww*)

Calculates the areas above and below supplied base line on cross section.

Surf.ReturnCrossSectionHeight

(*horint*, *zint*, *zdiff*, *nwww*)

Calculates the height (**zint**) and gradient (**zdiff**) of a point on along (**horint**) cross-section.

Surf.ReturnInternalNetworkBound

(*npol*, *polygons*, *visible*, *nttt*)

Returns set of polylines (GPOLYGON3) representing internal boundaries and optionally drawing them according to **visible** (Class=Visibility).

Surf.ReturnNetworkBound

(*npts*, *points*, *visible*, *nttt*)

Returns the boundary of the convex hull defined by the data set which has been previously stored in **nttt** and optionally draws it according to **visible** (Class=Visibility).

Surf.ReturnNetworkInfo

(*npts*, *ntri*, *nttt*)

Returns number of points (**npts**) and number of triangles (**ntri**) in triangulation workspace **nttt**.

Surf.ReturnNetworkSpotHeight

(*xint*, *yint*, *zint*, *deriv*, *nttt*)

Computes the height (or Z value) and its derivative at the point (**xint**, **yint**) by interpolation from a triangulated network.

Surf.ReturnNetworkSpotHeight4D

(*xint*, *yint*, *zint*, *zint2*, *deriv*, *deriv2*, *nttt*)

Computes the heights (or Z values) and derivatives at the point (**xint**, **yint**) by interpolation from a 4D triangulated network.

Surf.ReturnSurfaceArea

(*area*, *nwww*)

Calculates area of current surface plot.

GINOSURF

Surf.ReturnSurfaceProfile

(**npts**, **points**, visible, **nwww**)

Returns **npts** points in array **points** (GPOINT) used to draw the visible profile of the surface and optionally drawing it according to **visible** (Class=Visibility).

Surf.ReturnSurfaceSpotHeight

(**xint**, **yint**, **zint**, **nwww**)

Computes the height or Z value of the point (**xint**, **yint**) by interpolation from gridded workspace.

Surf.ReturnSurfaceSpotHeight4D

(**xint**, **yint**, **zint**, **zint2**, **nwww**)

Computes the heights or Z values of the point (**xint**, **yint**) by interpolation from 4D gridded workspace.

Surf.ReturnSurfaceVolume

(**hbase**, **volpos**, **volneg**, **nwww**)

Calculates volume above and below supplied base line in current surface plot.

Surf.SaveWorkspace

(filename, **nwww**)

Saves GINOSURF workspace to a disk file.

Surf.SelectNetworkContourDataSet

(dim)

Switches triangulated contour routines to use 3/4D data (Class=ContourDataSet).

Surf.SelectTriangle

(**xp**, **yp**, **triangle**, **vertex**, **side**, **nttt**)

Returns **triangle** (GTRIANGLE), **vertex** and **side** information from coordinate selection point.

Surf.SetContourAnnotation

(**skip**, **gaplen**, **chhght**, **nwid**, **nplace**, **npow**, **ncol**, **anndir**)

Specifies the **skip**, gap length (**gaplen**), character height (**chhght**), field width (**nwid**), number of decimal places (**nplace**), power factor (**npow**), colour (**ncol**) and direction (**anndir**) of annotation on contours. Format and power factors also apply to subsequent cross-section height values.

anndir (Class=ContourAnnotationDirection)

Surf.SetContourDrawingSwitch

(switch)

Switches the calculation and drawing of contour line boundaries for surface area fill routines (Class=Switch).

Surf.SetContourKeyPos

(limits, [**nlabels**, **labels**], [**side**, **chfrac**, **anncol**, **keyend**])

Defines the attributes of the contour key box when drawn with contour maps or surfaces. Its position is set by **limits** (GLIMIT). Optional labels can be supplied in **gsLabels**, with annotation being drawn on **side** (Class=Justification), at **chfrac** proportion of current character size, in **anncol** (Class=Colour) colour and with **gsKeyend** (Rounding) determining the end conditions.

Surf.SetContourLevels

(**ncont**, **zhght**, [**linestyles**])

Specifies that **ncont** contour levels at heights **zhght** and optional **linestyles** (Class=LineStyle) are to be drawn (overriding **ncont** in contour drawing routines).

Surf.SetContourLimits

(**zll**, **zuu**)

Specifies lower and upper contour levels to be drawn on subsequent contour maps.

GINOSURF

Surf.SetContourMapAxesStyle

(x1,x2,y1,y2)

Sets style of contour plot axes (Class=FrameAxesStyle) with respect to tickmarks and values.

Surf.SetContourMapFrame

(limits)

Specifies the exact size in space coordinates of where a contour map frame (GLIMIT) is drawn.

Surf.SetContourMapFrameScale

(xincr, yincr, cincr, difs1, difs2)

Specifies annotation intervals for X, Y and height for contour and cross-sectional plots using **difs1** (Class=SurfAxesScaling) and **difs2** (Class=Rounding).

Surf.SetContourMapFrameType

(frame)

Specified contour map frame type (Class=ContourMapFrameType).

Surf.SetContourMapHeight

(switch, height)

Sets **switch** (Switch) for perspective projection to be used for contour maps and defines the **height** at which it is to be drawn.

Surf.SetContourMapLineStyle

(nline1, nline2, nline3, skip)

Define line style used in contour drawing routines. The first and every subsequent (**skip**+1)th contour is highlighted.

nline1 (Class=LineStyle) Frame

nline2 (Class=LineStyle) Unhighlighted contours

nline3 (Class=LineStyle) Highlighted contours.

Surf.SetCrossSectionAxesStyle

(x1,x2,y1,y2)

Sets style of cross-section axes (Class=FrameAxesStyle) with respect to tickmarks and values.

Surf.SetCrossSectionFrame

(limits)

Specifies **limits** (GLIMIT) of drawing area for subsequent cross-sections.

Surf.SetCrossSectionFrameType

(frame)

Specifies style of subsequent cross-sections (Class=CrossSectionFrameType).

Surf.SetDerivativesInterpolation

(nopt)

Sets method of calculating derivatives (Class=DerivativeInterpolationMethod).

Surf.SetHeightRatio

(ratio)

Specifies **ratio** of height units to base units in subsequent projections. Default = 0.5.

Surf.SetInactiveSurfaceDisplayMode

(mode)

Sets display **mode** (Class=InactiveLevel) of inactive surface triangles.

Surf.SetInterpolationPitch

(nlp)

Specifies average number of points used in interpolation patch for gridded data.

GINOSURF

Surf.SetMissingValueMode

(mode, zlev)

Sets missing value **mode** (Class=Switch) and its data threshold (**zlev**).

Surf.SetNetworkContourStep

(stplen)

Selects the average step size in millimetres to be used when contouring through a triangulated data network.

Surf.SetNetworkFrame

(limits, nttt)

Sets up **limits** (GLIMIT) and scaling for subsequent triangulated data contour maps.

Surf.SetNetworkInterpolation

(numint, nearpp, nweigh)

Offers a number of interpolation methods, **numint** (Class=NetworkInterpolationMethod), number of near points to be used, **nearpp** and weighting factors, **nweigh** (Class=WeightingFactor) for interpolation from random data onto triangular network.

Surf.SetNetworkLineStyle

(ngrid, nsymb, nvalue, numnod, numtri, narrow)

Sets triangulation contour map feature line styles (Class=LineStyle) for grid lines, symbols, values, node numbers, triangles numbers and gradient field arrows.

Surf.SetPerspProjection

(limits, rad, theta, phi)

Sets up **limits** (GLIMIT) and scaling for subsequent perspective surface drawing, contour map projections and drawing of perspective axes.

Surf.SetPolylineCrossSectionProj

(naxis)

Sets the projection of a polyline cross section (Class=CrossSectionProjection).

Surf.SetRandomInterpolation

(nopt, factor)

Specifies method, **nopt** (Class=RandomInterpolationMethod) and **factor** when interpolating gridded data from random:

Surf.SetRandomPointMarker

(type, symb)

Marks random data points on next contour map or surface projection using **type** (Class=DataPointAnnotation) and **symb** (Class=Marker).

Surf.SetSurf3DState

(state)

Sets 3D hardware **state** (GSURF3D) for GINOSURF indicating whether to use 3D drawing primitives and or transformations.

Surf.SetSurfaceAxesIntervals

(xincr, yincr, zincr)

Specifies scaling intervals for projected axes and subsequent cross-sections.

Surf.SetSurfaceAxesLabelMode

(mode)

Sets position of surface axes labels (Class=SurfaceAxesLabelMode).

Surf.SetSurfaceAxesRatio

(ratio)

Varies the **ratio** of the X and Y perspective axes.

GINOSURF

Surf.SetSurfaceAxesStyle

(style, sup)

Sets the **style** (GPRJAXI) of axis for a projected plot and switches the suppression (GPRJAXI) of annotation on an individual axis.

Surf.SetSurfaceBaseStyle

(style, index)

Sets the **style** (Class=SurfaceBaseStyle) and fill style, **index**[3] (GFILL) for filling to be used for the visible parts of the base of a projected surface.

Surf.SetSurfaceBaseType

(frame)

Defines projected surface **frame** (Class=SurfaceBaseType) type:

Surf.SetSurfaceFrame

(limits)

Specifies **limits** (GLIMIT) of drawing area for all projections.

Surf.SetSurfaceGridInterval

(nxstep, nystep)

Specifies selection of section lines on projections at intervals **nxstep**, **nystep**.

Surf.SetSurfaceGridMode

(section)

Specifies the type of **section** (Class=SurfAxis) lines to be drawn on projections:

Surf.SetSurfaceLevelLimits

(zll, zhh)

Specifies the precise lower and upper data values to be used when drawing a projected surface.

Surf.SetSurfaceLineShading

(polylines, contours, gridlines)

Switches shading (Class=Switch) for surface features.

Surf.SetSurfaceLineStyle

(nline1, nline2, nline3)

Allows the user to change the default line style used in any projection.

nline1 (Class=LineStyle)	Axes
nline2 (Class=LineStyle)	Upper surface
nline3 (Class=LineStyle)	Lower surface

Surf.SetSurfaceNormalDirMode

(mode)

Sets surface normal direction **mode** (Class=SetSurfaceNormalDirMode).

Surf.SetSurfaceTextureMode

(set, level)

Sets surface texture drawing triangle **set** (Class=TriangleSet) and **level** (Class=TextureLevel).

Surf.SetSurfAxesAnnotation

(numaxi, nwidth, nplace, npow)

Sets up user-specified annotation formats for numaxi (Class=SurfAxis) axis in terms of files width, number of decimal places and power factor.

Surf.SetSurfCharMode

(mode)

Determines which character mode (Class=SurfCharMode) to use.

Surf.SetSurfLabels

(axtype, string)

Specifies a **string** to override or suppress the default annotation on axtype (Class=SurfLabel) label.

GINOSURF

Surf.SetSurfPlotTitle

(string)

Specifies title to be used on subsequent drawings.

Surf.SwapDiagonal

(triangle, side, nttt)

Swap diagonal of specified quadrilateral within triangulation network.

Surf.SwitchCrossSectionScaling

(switch)

Switches (Class=Switch) use of the new cross-section facilities

Surf.TransformCrossSectionCoord

(hor, vert, space)

Converts from cross-section data coordinates (**hor**, **vert**) to space coordinates (GPOINT)

Surf.TransformMapCoord

(xcon, ycon, space)

Converts contour map coordinates (**xcon**, **ycon**) to space coordinates (GPOINT).

Surf.TransformSurfaceCoord

(x, y, z, point)

Converts perspective projection coordinates (**x**, **y**, **z**) to space coordinates (GPOINT).

Surf.TransformToCrossSectionCoord

(xspace, yspace, point)

Converts from space coordinates (**xspace**, **yspace**) to cross-section data coordinates (GPOINT).

Surf.TransformToMapCoord

(xspace, yspace, con)

Converts space coordinates (**xspace**, **yspace**) to contour map coordinates (GPOINT).

Surf.TrimNetworkBound

(nopt, value, niter, visible, npts, points, nttt)

Adjusts the boundary of a convex region by removing 'long-thin' triangles according to **nopt** (Class=TrimMode). Angle or ratio is passed in **value** with **niter** setting the number of iterations. The new boundary is returned in array **points** (GPOINT) and optionally drawn according to **visible** (Class=Visibility).

DRIVERS

Drivers

A complete range of device drivers are available with GINO, some are supplied as standard and some are optional depending on your system installation. Common drivers are as follows:

- Gino.ColourEps ()
Gino.Eps (icflag, xmarg, ymarg, xsize, ysize, xdef, ydef)
Gino.Epsexp (xsize, ysize, xmarg, ymarg, n, prop)
Generic device drivers for creating encapsulated Postscript for outputting to printers (Gino.Eps) or saving to file (Gino.Epsexp).
icflag +ve=colour, -ve=grey, 0=mono
xmarg X margin
ymarg Y margin
xsize Max width of drawing area
ysize Max length of drawing area
xdef, ydef Default size of drawing area
n,prop No. of and property settings
- Gino.Jpeg ()
Driver for creating JPEG images.
- Gino.Mwinw (ix, iy, iw, ih)
Gino.Mwin ()
Gino.Mwinao (hWnd, hDC)
Gino.Mwindc (hWnd, hDC)
Gino.Mwinp ()
Gino.Mwindp ()
Gino.Mwinpp (mode, devnam, filnam, n, prop)
The family of device drivers for Microsoft Windows GDI driver. Gino.Mwinw() and Gino.Mwin() create a new window, Gino.Mwinao() and Gino.Mwindc() (private bitmap) sends graphics to an existing window or .NET object. Gino.Mwinp() opens the printer dialog, Gino.Mwindp() sends the output to the default printer and Gino.Mwinpp() sets printer characteristics.
ix, iy Pixel position of top left corner of window
iw, ih Width and height of window in pixels
hWnd Instance of existing window
hDC Device Context of existing window or .NET object
- Gino.Png ()
Driver for creating web-based graphics based on the Portable Network Graphics protocol.
- Gino.Woglw (ix, iy, iw, ih)
Gino.Wogl ()
Gino.Woglao (hWnd, hDC)
Gino.Wogldc (hWnd, hDC)
Gino.Woglp ()
Gino.Wogldp ()
Gino.Woglpp (mode, devnam, filnam, n, prop)
The family of device drivers for creating OpenGL graphics under Microsoft Windows. All arguments are the same as for Mwin driver.

GLOBAL ATTRIBUTES

Character escape sequences

Fonts and other character attributes can either be selected with individual routines or by way of the 'escape' mechanism in character output routines such as `Gino.DisplayStr()`. The escape sequences are as follows:

<code>*L</code> or <code>*l</code>	Shift into lower case
<code>*U</code> or <code>*u</code>	Shift into upper case
<code>*Fn</code> or <code>*fn</code>	Temporary change to font n where n=000 -199 (3 digits)
<code>*N</code> or <code>*n</code>	Move to next line
<code>*E</code> or <code>*e</code>	Activate current exponent position and size settings
<code>*I</code> or <code>*i</code>	Activate current index position and size settings
<code>*O</code> or <code>*o</code>	Centre next character over previous character
<code>*S</code> or <code>*s</code>	Start underscore
<code>*W</code> or <code>*w</code>	Increase current font weight by 3
<code>*\</code>	Set italic angle to -15 degrees
<code>* </code>	Reset italic angle to that on entry
<code>*/</code>	Set italic angle to +15 degrees
<code>*A</code> or <code>*a</code>	Set current position on base line as an alignment position
<code>*B</code> or <code>*b</code>	Move back to last align position on base line
<code>*:</code>	Displays umlaut form of character if followed by <code>au</code> <code>AOU</code>
<code>*:S</code>	Displays German sz character if available in current font
<code>**</code>	Means output*

The escape character `*` can be set to any other character by the `Gino.SetEscapeChar()` routine.

Broken Line Styles

GINO provides 16 default broken line styles out of a possible 256 (the first 8 of these may be accessed by name using the enumerator class `BrokenLineStyle`). Further styles can be defined by calling `Gino.DefineBrokenLineStyle()`.

0	_____		
1	-----	9	-----
2	10	-----
3	-----	11	-----
4	-----	12	-----
5	- - - - -	13	-----
6	-----	14	-----
7	15	-----
8	-----	16	-----

Markers

Nine standard markers are available in GINO as follows with their names being made available through the enumerator class `Marker`, with additional symbols being available via the symbol fonts.

<code>GDOT</code>	<code>GUP</code>	<code>GDOWN</code>	<code>GPLUS</code>	<code>GCROSS</code>	<code>GBOX</code>	<code>GDIAMOND</code>	<code>GCRICLE</code>	<code>GSTAR</code>
●	△	▽	+	×	□	◇	○	✱

GLOBAL ATTRIBUTES

Fonts

GINO provides a variety of software fonts, symbols and dingbats plus access to all commonly available hardware fonts, to be used with all GINO packages. Fonts may be accessed by name using the enumerator class `FontFace`, or by number:

Font 0	The default fixed-pitch stick font
Fonts 1-14	Proportional stick fonts
Fonts 15-20	Proportional polygonal fonts
Fonts 21-25	Proportional Greek fonts
Fonts 70-79	Symbols and Dingbats
Font 100	Fixed-pitch hardware font (Courier)
Fonts 101-108	Proportional hardware fonts

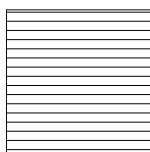
1: Roman Simplex	2: Roman Duplex
3: Roman Complex	4: Roman Triplex
5: <i>Italic Complex</i>	6: <i>Italic Triplex</i>
7: <i>Script Simplex</i>	8: <i>Script Complex</i>
9: Γρεεκ Συμπλεχ	10: Γρεεκ Ψομπλεχ
11: Gothic English	12: Gothic German
13: Gothic Italian	14: Вшсиллив Вомплдч
15: Swiss Solid	16: Dutch Solid
17: WESTERN	18: Сomputег
19: Display	20: Latin
101: Helvetica	102: Times
103: Avant Garde	104: Lublin Graph
105: New Century	106: Souvenir
107: Palatino	108: <i>Chancery</i>

GLOBAL ATTRIBUTES

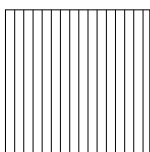
Fill Styles

GINO routines that use fill styles can use any of the 16 built-in patterns in GINO out of a possible 256. These are available by name using the enumerator class `FillStyle` or by number. In addition to the list below, `FillStyle.GHOLLOW` refers to an empty box and `FillStyle.GSOLID` refers to a solid box.

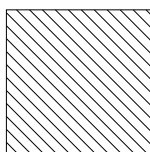
Further fill styles can be obtained by calling `Gino.DefineHatchStyle()` to define your own patterns.



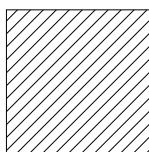
GFINEHORIZONTAL



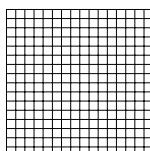
GFINEVERTICAL



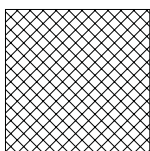
GFINELEFTDIAGONAL



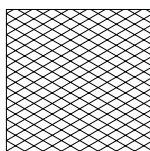
GFINERIGHTDIAGONAL



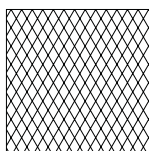
GFINEHORIZONTALGRID



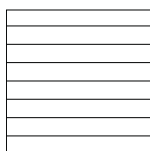
GFINEDIAGONALGRID



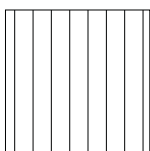
GFINEHORIZONTALMESH



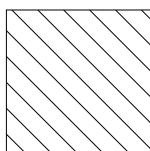
GFINEDIAGONALMESH



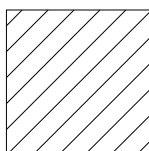
GCOARSEHORIZONTAL



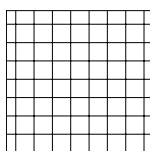
GCOARSEVERTICAL



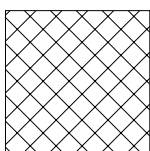
GCOARSELEFTDIAGONAL



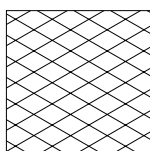
GCOARSERIGHTDIAGONAL



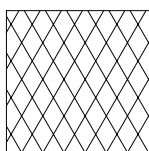
GCOARSEHORIZONTALGRID



GCOARSEDAGONALGRID



GCOARSEHORIZONTALMESH



GCOARSEDAGONALMESH

Colour

Colours may be defined in a colour table or passed as 24 bit RGB triplets (using the function `Gino.TrueCol()`). The first 10 colours of the GINO colour table are initialized to the following colours with their names being available in the enumerator class `Colour`:

GBACKGROUND	Background colour (grey on screens, white on printers)
GBLACK	Black
GREY	Red
GORANGE	Orange
GYELLOW	Yellow
GGREEN	Green
GCYAN	Cyan
GBLUE	Blue
GMAGENTA	Magenta
GBROWN	Brown
GWHITE	White

These names are also attached to the enumerator class `LineStyle` (with the exception of `GBACKGROUND`) as this table is initially set to match the colour table but may be altered using routine `Gino.DefineLineStyle()`.

GLOBAL ATTRIBUTES

ASCII Character Table

NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	' 96	p 112
SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
EOT 4	DC4 20	\$ 36	4 52	D 68	T 84	d 100	t 116
ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
HT 9	EM 25) 41	9 57	I 73	Y 89	i 105	y 121
LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	 124
CR 13	GS 29	- 45	= 61	M 77] 93	m 109	} 125
SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	
SI 15	US 31	/ 47	? 63	O 79	_ 95	o 111	DEL 127

GINO