

Borland Graphics Interface (BGI) for Windows

Version 6.0, August 9, 2004

The following functions are mostly from the original Borland Graphics Interface for DOS programs. The BGI graphics functions may also be used with Windows programs created by the Borland 5.0 compiler, the free [GNU C++ compiler](https://www.gnu.org/software/gcc/), and possibly other compilers. Extra Windows functions are also available, described in www.cs.colorado.edu/~main/bgi/doc/bgi.html. These extra functions are indicated below by **WIN**. Also, any of the functions that use colors can use [RGB colors](#) in addition to the 16-color BGI palette.

Functions:

```
void arc (int x, int y, int stangle, int endangle, int radius);

void bar (int left, int top, int right, int bottom);

void bar3d (int left, int top, int right, int bottom, int depth, int topflag);

ostream bgiout; WIN

void circle (int x, int y, int radius);

void cleardevice (void);

void clearmouseclick(int kind); WIN

void clearviewport (void);

void closegraph (int window=ALL_WINDOWS); WIN

int converttorgb (int color); WIN

void delay (int millisec); WIN

void detectgraph (int *graphdriver, int *graphmode);

void drawpoly (int numpoints, int *polypoints);

void ellipse (int x, int y, int stangle, int endangle, int xradius, int yradius);

void fillellipse (int x, int y, int xradius, int yradius);

void fillpoly (int numpoints, int *polypoints);

void floodfill (int x, int y, int border);

int getactivepage (void); WIN

void getarccoords (struct arccoordstype *arccoords);

void getaspectratio (int *xasp, int *yasp);

int getbkcolor (void);

int getch (void); WIN

int getcolor (void);

int getcurrentwindow (void); WIN

struct palettetype* getdefaultpalette (void);

int getdisplaycolor (int color); WIN

char* getdrivername (void);

void getfillpattern (char *pattern);

void getfillsettings (struct fillsettingstype *fillinfo);

int getgraphmode (void);

void getimage (int left, int top, int right, int bottom, void *bitmap);

void getlinesettings (struct linesettingstype *lineinfo);

int getmaxcolor (void);

int getmaxmode (void);

int getmaxheight (void); WIN

int getmaxwidth (void); WIN

int getmaxx (void);

int getmaxy (void);

char* getmodename (int mode_number);

void getmoderange (int graphdriver, int *lmode, int *hmode);
```

```

void getmouseclick(int kind, int& x, int& y); WIN

void getpalette (struct palettetype *palette);

int getpalettesize (void);

int getpixel (int x, int y);

void gettextsettings (struct textsettingstype *texttypeinfo);

void getviewsettings (struct viewporttype *viewport);

int getvisualpage (void); WIN

int getwindowheight (void); WIN

int getwindowwidth (void); WIN

int getx (void);

int gety (void);

void graphdefaults (void);

char* grapherrormsg (int errorcode);

int graphresult(void);

unsigned imagesize (int left, int top, int right, int bottom);

void initgraph (int *graphdriver, int *graphmode, char *pathtodriver);

int initwindow (int width, int height, const char* title="Windows BGI", int left=0, int top=0, bool dbflag=false, bool closeflag=true); I

int installuserdriver (char *name, int huge (*detect)(void));

int installuserfont (char *name);

bool ismouseclick(int kind); WIN

int kbhit (void); WIN

void line (int x1, int y1, int x2, int y2);

void linere1 (int dx, int dy);

void lineto (int x, int y);

int mousex (void); WIN

int mousey (void); WIN

void moverel (int dx, int dy);

void moveto (int x, int y);

void outtext (char *textstring);

void outtextxy (int x, int y, char *textstring);

void pieslice (int x, int y, int stangle, int endangle, int radius);

void printimage (
    const char* title=NULL, double width_inches=7,
    double border_left_inches=0.75, double border_top_inches=0.75,
    int left=0, int right=0, int right=INT_MAX, int bottom=INT_MAX
); WIN

void putimage (int left, int top, void *bitmap, int op);

void putpixel (int x, int y, int color);

void readimagefile (
    const char* filename=NULL,
    int left=0, int top=0, int right=INT_MAX, int bottom=INT_MAX
);

void rectangle (int left, int top, int right, int bottom);

int registerbgidriver (void (*driver)(void));

int registerbgifont (void (*font)(void));

void registermousehandler (int kind, void h(int, int)); WIN

void restorecrtmode (void);

```

[RGB functions:](#) **WIN**

[COLOR\(r,g,b\)](#),

[RED_VALUE\(v\)](#), [GREEN_VALUE\(v\)](#), [BLUE_VALUE\(v\)](#),

[IS_BGI_COLOR\(v\)](#), [IS_RGB_COLOR\(v\)](#)

void [sector](#) (int x, int y, int stangle, int endangle, int xradius, int yradius);

void [setactivepage](#) (int page);

void [setallpalette](#) (struct palettetype *palette);

void [setaspectratio](#) (int xasp, int yasp);

void [setbkcolor](#) (int color);

void [setcolor](#) (int color);

void [setcurrentwindow](#) (int window); **WIN**

void [setmousequeuestatus](#)(int kind, bool status=true); **WIN**

void [setfillpattern](#) (char *upattern, int color);

void [setfillstyle](#) (int pattern, int color);

unsigned [setgraphbufsize](#) (unsigned bufsize);

void [setgraphmode](#) (int mode);

void [setlinestyle](#) (int linestyle, unsigned upattern, int thickness);

void [setpalette](#) (int colornum, int color);

void [setrgbpalette](#) (int colornum, int red, int green, int blue);

void [settextjustify](#) (int horiz, int vert);

void [settextstyle](#) (int font, int direction, int charsize);

void [setusercharsize](#) (int multx, int divx, int multy, int divy);

void [setviewport](#) (int left, int top, int right, int bottom, int clip);

void [setvisualpage](#) (int page);

void [setwritemode](#) (int mode);

int [showerrorbox](#) (const char *message); **WIN**

int [swapbuffers](#) (void); **WIN**

int [textheight](#) (char *textstring);

int [textwidth](#) (char *textstring);

void [writeimagefile](#) (

const char* filename=NULL,

double width_inches=7, double border_left_inches=0.75, double border_top_inches=0.75,

int left=0, int top=0, int right=INT_MAX, int bottom=INT_MAX

); **WIN**