

Window Driver Command Reference Contents

Overview

wd commands 11!:0

gl2 commands 11!:2000+n

fontspec

isigraph events

Mapping Mode

gl3 commands 11!:3000+n

OpenGL printing

Overview

The foreign family $11! : x$ is the J-Windows interface.

All $11! : x$ verbs are rank 1.

wd commands are executed when x is 0 ($11! : 0$).

gl2 commands are for 2D (GDI) graphics (x between 2000 and 2999).

gl3 commands are for 3D (OpenGL) graphics (x between 3000 and 3999)

wd commands 11! : 0

wd takes a string right argument and returns a string result. A wd argument is 0 or more statements delimited by semicolons. A statement is a command followed by 0 or more parameters.

Commands and parameters are separated by one or more whitespace characters from the set: space, carriage return, line feed, tab, vertical tab, and form feed. Simple parameters start with an alphanumeric, -, or _, and run to a ; or whitespace. A * starts a parameter at the next character, and runs to the end of the array. Delimited parameters start with ", 255{a. (EAV), or 197{a. and run to the matching delimiter. Some commands are not supported in some environments.

A wd error is a J domain error. Command qer returns the error message text and wd argument index for the last error.

The first letters of a command usually indicate a category: dde, mb (message box and common dialogs), menu, ole, p (parent), q (queries), set (setting properties), sm (session manager), and vbx.

A parameter can be an id, style, number, or text. An id identifies a parent, child, menu item, or other object. A style is a keyword, usually with a prefix. For example, bs_autoradiobutton is a button style.

A color is 3 parameters giving RGB values in the range 0 to 255. For example, 0 0 0 is black, 255 255 255 is white, and 255 0 0 is red.

A bool is 0 or 1 and is 1 if elided.

Several commands (clipcopy, smappend, smwrite, and set for edit and editm controls) have an optional parameter called trailing. If trailing is elided or 1, then trailing blanks are removed. If trailing is 0, then trailing blanks are not removed.

wd commands

acttask taskhandle ; activate task. winexec returns taskhandle.

acttitle title ; activate window with title bar that matches (case insensitive)

beep i ; beep i times

cc id class [style...] ; create child

clipcopy text [trailing] ; put text on the clipboard (line frets converted to host frets)

clipcopyx text ; put text on the clipboard (data unchanged)

clippaste ; result is text from the clipboard

cn name ; set child name after a cc command (otherwise setcaption is preferred)

creategroup id ; causes parent commands to be ignored so that, when a form definition is run, the child controls created are in the current form. The id is a control in the current form, usually a tab control, and child creates are offset from that control. Controls created under creategroup are hidden and are part of a group with the id from their ignored pc command. The setshow command with a group id hides or shows the controls in a group. A form definition loaded under creategroup is a normal form definition that can be designed and tested with the form editor. It is loaded by a main form with a form_run that is bracketed by creategroup commands. The first creategroup gives the id of a control in the main form and the final creategroup command has no parameter and ends the group.

creategroup ; ends the group started by a previous creategroup command

ddeadvise t i d ; send data to client in an advise loop. Client will get ddeadvise event.

ddecons ; return s|t active conversations

ddedis [s [t]] ; discontinue conversations

wd 'ddedis;' conversations with all servers

wd 'ddedis s;' conversations with server s

wd 'ddedis s t;' conversation with server s on topic t

ddeex s t d ; data d is sent to server s for topic t to execute

ddename id ; set dde service name (can be done only once). Command line /ddename= can also set dde service name.

ddepoke s t i d ; send data to S|T|I

ddereq s t i ; data for item i is requested from S|T

ddereqd d ; send data in response to ddereq event (must be done immediately after ddereq event)

ddestart s t i ; advise loop requested for S|T|I. New data signals ddeadvise event.

ddestop s t i ; stop advise loop on S|T|I

fontdef fontspec; default font used when a child is created.

formsize button-width , button-height , edit-width, edit-height , label-height ; sets form edit default control sizes.

makejr in.ijrs out.ijr ; converts ijs input file to a J runtime ijr file

mb title text [style...]; messagebox with styles from set (mb_arbortretryignore, mb_defbutton2, mb_defbutton3, mb_iconasterisk, mb_iconexclamation, mb_iconhand, mb_iconinformation, mb_iconquestion, mb_iconstop, mb_ok, mb_okcancel, mb_retrycancel, mb_yneno, mb_ynenocancel)

`mbcolor [colors]` ; choose color common dialog box. Result is chosen color and 16 custom colors. Colors are 3 integers giving RGB values. Optional argument sets initial chosen color and 16 custom colors (51 integer values in range 0 to 255).

`mbfont [fontspec]` ; choose font common dialog. Start with last chosen and set new one if OK. Result is a fontspec.

`mbopen title directory filename filterpairs [style...]` ; open file common dialog returns fully qualified file name selected by user. Filterpairs are delimited by |. Styles are from the set (`ofn_createprompt`, `ofn_filemustexist`, `ofn_nochangedir`, `ofn_overwriteprompt`, `ofn_pathmustexist`).

```
wd 'qs mbopen'  
wd 'mbopen'  
wd 'mbopen mytitle "" "" "Write(*.wri)|*.wri|(Word(*.doc)|*.doc"  
ofn_filemustexist'
```

`mbprinter [pd_printsetup]` ; The parameter selects the setup dialog instead of the print dialog. The result is "" if the user pressed CANCEL. If the user pressed OK, the result is the name of the selected printer followed by LF delimited values. Currently the only value after the name is the print-to-file value from the print dialog. This sets the session default printer setup that is used by `glzcreate` with no argument.

`mbsave title directory filename filterpairs [style...]` ; save file common dialog returns fully qualified file name (see `mbopen`).

`menu id text` ; add menu item

`menupop text`; add popup menu item

`menupopz` ; ends popup menu and drops down a level

`menusep` ; separator line in a popup menu

`oledlg id` ; run property dialog. State can be saved with the `olesave` command.

`oleenable id eventname [bool]` ; enable/disable event. You must enable an event in order to trigger an event in J.

`oleget id objectname property` ; return property value. Objectname is *base*, *temp*, or a name set with `oleid`. If the result is an object, it is set as the temp object. This allows a series of `wd` commands that use the temp object to get the next object.

`oleinfo id` ; return information about events, methods, properties, and constants

`oleload id filename` ; initialize properties from a file created by `olesave`. An `oleload` should only be done once before it is shown.

`olemethod id objectname method parameters....`; run a method. , is an elided parameter. A `wd` parameter of , is the same as "", except it is treated as an elided parameter where

appropriate. Some methods distinguish between a numeric parameter and a string. A simple (not delimited) string that is an integer is treated as an integer. If you want 23 to be treated as a string, use "23". If the result is an object, it is set as the temp object.

An object parameter is indicated by a simple parameter of the form:
object:formid.childid.objectname

A picture object parameter is indicated by a simple parameter of the form:
picture:filename

olesave id filename ; save properties in a file that can be used to initialize a control after it is created

oleset id objectname property value ; set property value

pective ; SetActiveWindow for selected parent

pas i j ; parent size adjusted to provide i and j margins beyond children

pc id [style...] ; parent create. Styles are: nomenu, nomin, nomax, nosize, dialog, owner, and closeok.

pcenter ; center parent on screen

pclose ; close parent. If no parent selected, nothing is done and there is no error.

phandler sentence; sentence to run as form event handler instead of normal wdhandler'

pmove x y w h ; move and resize parent. Values are in logical units and are relative to the top left corner of the screen. Value of -1 inherits current value.

pmovex x y w h ; move and size form in pixels

pn text ; name for parent window caption

psel id ; select parent id to be target for subsequent commands

psel n ; select parent with qhwnp of n for subsequent command

psel ; clear parent and child selection

pshow [style] ; style is from the set: sw_hide, sw_minimize, sw_restore, sw_show, sw_showmaximized, sw_showminimized, sw_showminnoactive, sw_showna, sw_shownoactivate, sw_shownormal. sw_shownormal is the default.

pshowx n... ; argument is sysinfo data. Shows window as it was when sysinfo was set.

ptop [bool]; set 1 so that window stays on top of other windows

q ; return event data

qabout ; returns: release, minor release, system, edition, registration.

4.01

a

Windows 95

Professional Edition

0123-456-7890

Possible system strings: Macintosh, Windows NT, Windows 95, Windows 3.1.

qbreak ; returns count of menu/button/and toolbar commands that have been ignored because J was busy. This value can be polled in a loop with a long execution time to see if the user is getting impatient.

qchildxywh ; return child position and size in units

qchildxywhx ; return child position and size in pixels

qcmdline ; return command line

qcode data ; encodes data for use in order system.

qcolor color ; return RGB for system colors defined in system\packages\color\wdcolor.ijs

qd ; return data for form. Similar to q result.

qer ; return last error information

qformsize ; returns form edit default control sizes: button width, button height, edit width, edit height, label height.

qformx ; returns form x y w h values in pixels.

qhinst ; return HINSTANCE of application

qhwndc ; return HWND (handle) of current child

qhwndp ; return HWND (handle) of current parent

qhwndx ; return HWND (handle) of application

qkeystate keys ; returns 0 or 1 (pressed) for each key in keys. The values in keys are virtual key values as defined in packages\graphics\vkeys.ijs.

qm ; return system metrics:

screen width, screen height,

x logical unit, y logical unit,

cxborder, cyborder,

cxfixedframe, cyfixedframe,
cxframe, cyframe,
cycaption, cymenu,
desktop x, desktop y,
desktop w, desktop h
More elements may be added.

qp ; return parent ids.

qprinters ; returns an LF delimited list of printers. The first entry is the session default printer. The second entry is the system default printer. The remaining entries are a list of all printers. The session default printer starts out as the system default printer and is changed by mbprinter or File|Print Setup.

qrt ; return 1 if runtime version

qrtf id; return RTF data from richedit control

qs [cmd] ; return styles or keywords used in command. cmd can be: cc, gpen, mb, mbopen, mbsave, pshow, sendkeys, smshow, or winexec

qscreen ; screen information (see glqprinter)

horzsize	width in millimeters
vertsize	height in millimeters
horzres	width in pixels
vertres	height in raster lines (pixels)
logpixelsx	horizontal pixels per logical inch
logpixelsy	vertical pixels per logical inch
bitapixel	number of color bits per pixel.
planes	number of color planes
numcolors	number of entries in the device's color table (_1 if > 8)
aspectx	relative width of a pixel
aspecty	relative height of a pixel
aspectxy	diagonal width of a pixel

qsmact ; return file name of active window

qsmall ; return file names of all windows

qsmcsize ; return size in characters of jx output window

qsmmfmove ; size and position in pixels of J session (mainframe) window

qsmmfshow ; session show value (see qsmmfshow)

qsmout ; return file name of JX output window

qsmsize ; return size of smsel window (x y w h) in pixels

qsmwh ; return size of main frame client area in pixels

qvbv id ; return information about properties and events of VBX control

rem text ; remark

reset ; close all windows, release all wd resources and objects

sbar n; create status bar with n panes

sbarset id width text; set id, width, and text of the next statusbar pane. A width of -1 leaves the width unchanged. A set command sets the text of the pane.

sbarshow [bool]; show (1) or hide statusbar

sendkeys text ; send keystrokes to active window (Win31 only)

+^%~()[]{} are special characters

+ upshifts the next character (+a is the same as A)

^ control shift

% alt shift.

~ ENTER key.

To send a special character as a normal character enclose it in {}. To send + , put {+} in text. {key n} repeats key n times. For example: {a 5} or {left 3}.

Keys without an ASCII value are indicated by keywords in braces. {tab} is the tabkey.

wd 'winexec calc.exe;'

wd 'acttitle calculator; sendkeys "123{+}4~";'

set id [parameters...] ; set object property

checkbox, radiobutton - check (1 or elided) or uncheck (0)

combobox - similar to listbox

edit, editm - set text. Takes optional trailing parameter.

isipicture - set filename of picture source. dib or bmp (device independent bitmap), wmf (windows metafile), or ico (icon) file to display.

listbox - clear contents and add items. A parameter can contain LF delimited items.

Ownerdrawfixed listbox items are picture file names.

wd 'set listbox red green blue'

wd 'set listbox red green blue "pink',LF,'orange"'

menu - check (1 or elided) or uncheck (0). If menu and toolbar button have same id, both are set.

progressbar - set value in range 0 to 100

richedit, richeditm - set RTF text

scrollbar - set min/position/max/pagesize values or just position

```
wd'set scrollbar 0 60 100 10'
```

```
wd'set scrollbar 75'
```

static - set text

statusbar - set pane text

tab - add a tab for each parameter

```
wd'set tab Jan Feb Mar Apr'
```

toolbar - check (1 or elided) or uncheck (0). If menu and toolbar button have same id, both are set.

trackbar - similar to scrollbar but has linesize in addition to pagesize

```
wd'set trackbar 0 60 100 10 2'
```

setbkgnid id color ; set background RGB color of richedit or richeditm control

setcaption id text ; set control caption. If control is ownerdraw button, text is picture filename.

setcolwidth id width ; set multicolumn listbox column width

setdelete id n ; delete tab item n

setenable id [bool] ; enable (1) or disable. Applies to menu items, toolbar buttons, and controls.

setfocus id ; set focus on control

setfont id fontspec ; set font for control

setinsert id n text ; insert tab item at position n

setinvalid id ; InvalidateRect

setlimit id n ' limit length of user typed text in edit, editm, richedit, richeditm, combodrop, and combobox controls.

setreadonly id [bool] ; set readonly (1) for edit, editm, richedit, and richeditm

setreplace id newtext ; replace selected text with newtext in edit, editm, richedit, and richeditm control

setreplace id n newtext ; replace tab item n text with newtext

setscroll id n ; scroll n to be first visible line in editm and richeditm control

setselect id [parameters...] ; set selection

edit, editm, richedit, richeditm - parameters are [start end [noscroll]]. Selects text from start to end. If start and end are elided, then all text is selected. If noscroll is elided or 1, then the selection is not scrolled into view. If noscroll is 0, then the selection is scrolled into view

combobox, listbox - -number of item to select. -1 clears selection.

trackbar - parameters are start and end of range to mark as selected

setshow id [bool] ; show (1) or hide. If it is a group id (the id of the pc command ignored by a creategroup), then all controls in the group are affected. Controls explicitly hidden are not affected by a group show.

settabstops id [n [n [n...]]] ; sets tab stops in editm and listbox controls. Values are in dialog units.

setupdate id ; UpdateWindow

setxywh id x y w h ; set child position and size in units

setxywhx id x y w h ; set child position and size in pixels

smappend text [trailing] ; appends text to smsel window

smcascade ; menu Window/Cascade

smclose ; close smsel window (does NOT save)

smfocus ; set smsel window in focus

smfont fontspec ; set font for all session manager windows

smformedit ; Edit|Form Edit for selected sm window

smgetsel ; return start and end caret selection information

smgetshow ; return sm mainframe window state: 0 normal, 1 maximized, 2 iconic

sminputlog ; return input log

smmfmove x y w h ; move and size J session in pixels. The qsmmfmove command returns the current information.

smmfshow [style] ; same style as for pshow. When J starts with a profile its starts with sw_hide and an smmfshow must be done to show the session. The qsmmfshow command returns the current information.

smove x y w y ; move and size smsel window (pixels)

smopen ; open (if not already open) file set by smsel

smprompt prompt ; used in input line recall

smread ; return text from smsel window

smsave ; save selected window (if modified)

smsbarshow bool] ; show or hide status bar

smscroll i ; scroll selected window

smsel filename ; sets file name used to select window for other commands

smselact ; selects (smsel) the active window

smselout ; selects (smsel) the output window

smsetcmd n t name sentence help ;

n is 2 to 9 and indicates a function key

t is 0 for no definition, 1 for noisy, 2 for quiet, 3 for silent

name appears on Tool menu (include TAB to put Fn at right)

sentence is passed to J for execution when function key is pressed

help appears in status bar line at bottom of main window

smsetselect [start end [noscroll]] ; Selects text from start to end. If start and end are elided, then all text is selected. If noscroll is elided or 1, then the selection is not scrolled into view. If noscroll is 0, then the selection is scrolled into view.

smshow [style] ; same style as for pshow

smtile ; menu Window/Tile

smtilea ; menu Window/Tile across

smwrite text [trailing] ; write new text to smsel

tbar filename; set filename of bmp file for toolbar

tbarset id index image; set id for a toolbar button. index is the position in the displayed toolbar and image is the index of the button bitmap to use from the toolbar bitmap.

tbarshow [bool]; show (1) or hide toolbar

timer i ; set interval timer to i milliseconds. Event systimer occurs when time has elapsed. The timer keeps triggering events until the timer is turned off. An argument of 0 turns the timer off.

The systimer event may be delayed if J is busy, and it is possible for several delayed events to be reported as a single event.

`vbxenable id eventname [parameter types] ; enable VBX control event`

`vbxevent eventname... ; VBX control`

`vbxget id property... ; get VBX control property`

`vbxrefresh id ; repaint VBX control`

`vbxset id property value... ; set VBX control property`

`wait ; the normal message loop executes a handler in immediate execution when an event occurs.`

This is overkill for getting input from a simple form, and the way `mb` works is simpler: the message box is displayed, other events are disabled until it is closed, and the result is the `wd` result. The `wait` command allows a form to be used in a similar way. The `wait` command shows the form, disables events except for that form, and waits for an event. The event information is given by `wd 'q'`. While a `wait` form is active events in other forms, DDE events, and timer events are disabled. They are enabled when the `wait` form is closed.

`winexec text [style]; Execute DOS or Windows program. style is same as for pshow.`

```
wd ' winexec "write.exe text.wri" sw_showmaximized;
```

`winhelp filename [context] ; run Windows help. Optional argument is integer topic number.`

Filename "" or " dictionary" (note leading blank) selects J Dictionary.

`xywh x y w h ; sets rectangle. -1 value inherits the previous value.`

gl2 commands 11! : 2000+n

gl2 commands are for 2D (GDI) drawing in an isigraph control. Script system\main\gl2.js defines gl2 verbs.

```
load 'gl2' NB. define verbs such as: glarc=:11!:2001
```

Most gl2 commands add drawing information to a buffer. When the control is painted, it is painted with these buffered commands. Some commands, such as glcursor are not buffered commands and either have an immediate effect, or change a global state.

mapping mode affects how the logical units of the gl2 commands are mapped onto the display surface.

isigraph events allow user interaction with the isigraph control with the keyboard and mouse.

gl2 commands

glarc x y w h xa ya xz yz - draw arc on the ellipse defined by rectangle. Arc starts at xa,ya and ends at xz,yz. Start and end points need not lie on the ellipse, they define a line from the center that intersects the ellipse.

glbkcolor " - set current background color to current color (last grgb). Background color is used in the gaps in styled lines and as the background for text.

glbkmode bool - set background mode for gbkcolor to OPAQUE with 0, or TRANSPARENT with 1

glbmp filename -display bitmap in glbmpxywh rectangle

glbmpxywh x y w h - set rectangle for the display of a bitmap

glbrush " - select solid brush in current color

glbrushnull " - select null brush (leaves area painted with it unchanged)

glcapture type - type is 0 release, 1 capture, 2 line band, 3 box band, 4 ellipse band, 5 vertical line, 6 horizontal line. A capture is normally done in a mouse down event and a glcapture 0 should be done in the mouse up event.

glcaret w h - create a caret at the current position with a width and height of w h.

glchord x y w h xa ya xz yz - same parameters as garc. Drawn with pen and brush.

glclear " - clear isigraph drawing buffer and reset defaults. If a printer context has been created, it is carried forward. Default values are 0 except for the following:

```
glmap anisotropic
glzmap twips
glbkcolor white
glwindowext 1000 1000
```

glplaymap anisotropic
glplayextent 1000 1000
glextent 1

glclear n - initialize graphics state. If n is " or 0, then an InvalidateRect is done, If n is 1, then it is not done.

glellipse x y w h - draw an ellipse in the rectangle with pen and brush

glextent bool - 0 selects the screen context and 1 selects the printer for glqextent and glqtextmetrics. The default is 1.

glextentfont fontspec - set font for glqextent and glqtextmetrics

glfile filename - set filename for glsave and glsavebmp. " is the clipboard.

glflood i j color - fill area around point i,j with the current brush. The area is bounded by color.

glfont fontspec - font for text commands

glgrid " - buffered drawing command that causes a grid to be painted. The grid is painted based on the state information set by the following glgrid... commands.

glgridatt attributes - A set of attributes is a list of 20 integers. The argument is the ravel of the sets of attributes.

attribute set: pen, brush, text, font, align, edit, 8 unused values

pen, brush, and text are 3 integers with RGB values

font: _1 selects font based on first character (-, _, and digit select font 1); 0 selects font 0, 1 selects font 1

align: _1 selects left or right based on first character (-, _, and digit selects right), 0 selects right, 1 selects left, 2 selects center.

edit: 0 not editable, 1 editable

glgridborder borders - a border is specified as 20 integers:

row, col, rows, cols, left b, top b, right b, bottom b

b field for each side is a type and RGB color

type is 0 to 3 and 11 for none, thin, medium, fat, and double line

glgriddrawmark " - undraw previous mark and draw new mark

glgridedit text - create edit box at mark

glgridedit 0 - destroy edit box

glgridedit 1 - enable arrow keys in edit box

glgridfill width height fixfilltype filltype - information about cells beyond the grid data

glgridfix row col - fixed rows and columns

glgridfont0 fontspec

glgridfont1 fontspec

glgridgetedit " - get text from edit box

glgridgettext row col - get text for this cell

glgridgettype row col - get type and attributes for this cell

glgridh heights - row heights for the grid data

glgridinvalidate bool - 1 causes subsequent glgridtext and glgridtype commands to invalidate the cells they change.

glgridmark row col rows cols - marked range of cells

glgridrc rows cols - rows and columns in grid data

glgridrchw row col rows cols - select subarray used in subsequent glgridtext and glgridtype commands

glgridscroll row col - scrolled rows and columns

glgridskip row col - rows and columns in total data that are not in grid data

glgridtext - NULL terminated strings of grid data

glgridtype types - integer value for each cell that selects its attributes

glgridw widths - column widths for the grid data

glline i j - draw line from current position to point i,j and update current position

gllined I j k l linelen spacelen [linelen spacelen ...] - draw dashed line from i,j to k,l. Works only for vertical and horizontal lines.

gllines pts - draw connected lines. pts is 2 or more points.

glmap mode - mode is one of: MM_TEXT, MM_LOMETRIC, MM_HIMETRIC, MM_LOENGLISH, MM_HIENGLISH, MM_TWIPS, MM_ISOTROPIC, MM_ANISOTROPIC. The default is anisotropic. See mapping mode.

glmapraw - turn off default isigraph mapping behavior. See mapping mode.

glmark - set mark in graphics buffer and reset pens etc.

glmarkc - clear graphics buffer back to last gmark (does not undraw)

glmove i j - move current position to i,j

glpaint " - does immediate paint of a 2D control. The painting of a 3D control is delayed until the paint event handler can be run.

glpaintx " - mark control for painting when possible. If a glpaintx is done and J continues execution for a while, then the painting will be delayed.

glpen i [style] - select pen. pen is color from last glrgb command and is i units wide. style is from the set (ps_solid, ps_dash, ps_dot, ps_dashdot, ps_dashdotdot, ps_null, ps_insideframe)

gllpie x y w h xa ya xz yz - draw pie shaped wedge with pen and brush. Same parameters as garc.

glpixel i j; draw pixel at i,j in current color

glplay filename _ the wmf file filename is played into the isigraph control. The glplaymap, glplayxywh, and glplayext commands determine how the file is played.

glplayext extx exty _ windowport extent information for scaling the wmf file

glplaymap mode _ set map mode for playing a wmf file. See mapping mode.

glplayxywh x y w h _ set rectangle for playing a wmf file. This is not a clipping rectangle and if the drawing goes beyond the w and h values it will be displayed. The w h values establish the extents that are used for scaling if the drawing is done in an isotropic or anisotropic mapping.

glpolygon pts _ draw polygon in pen and brush

glpolymode bool - 0 selects **alternate** filling of polygon and 1 selects **winding**

glqdevmode " - return printer document properties (see glzdevmode)

glqextent text - return width and height of the text in a particular font on a particular device. The glqextent command determines whether the screen or printer context is used and glqextentfont determines which font is used.

The layout (rasterization) of text is quite complicated and the final length of text is not simply the sum of the lengths of the individual characters. For this reason, calculations based on character widths are not very useful.

In addition, the length of text depends on the device. Drawing "testing 1 2 3" with an arial 240 font in a twips mapping will have different lengths on the screen, on a 150dpi printer, and on a 300dpi printer. Calculations such as line breaks, page breaks, and positioning stuff at the end of text are sensitive to the actual device.

glqprinter " - return printer information

- horzsize width in millimeters
- vertsize height in millimeters
- horzres width in pixels
- vertres height in raster lines (pixels)
- logpixelsx horizontal pixels per logical inch
- logpixelsy vertical pixels per logical inch
- bitapixel number of color bits per pixel.
- planes number of color planes
- numcolors number of entries in the device's color table (_1 if > 8)
- aspectx relative width of a pixel
- aspecty relative height of a pixel
- aspectxy diagonal width of a pixel
- physicalwidth width of the physical page. A 600 dpi printer on 8.5"x11" paper could have a physical width of 5100. The physical page is almost always greater than the printable area of the page, and never smaller.
- physicalheight height of the physical page
- physicaloffsetx distance from the left edge of the physical page to the left edge of the printable area. A 600 dpi printer on 8.5"x11" paper, that cannot print on the leftmost 0.25" of paper, has a horizontal physical offset of 150.
- physicaloffsety distance from the top edge of the physical page to the top edge of the printable area

glqtextmetrics " - return font information. The context (screen or printer) is selected by glextent and the font is selected by glextentfont. The values are: Height, Ascent, Descent, InternalLeading, ExternalLeading, AverageCharWidth, MaxCharWidth

glqwh " - return window width and height in pixels.

glrect x y w h - draw rectangle with pen and brush

glrgb color - set current color

glroundr x y w h rw rh - draw rectangle with rounded corners defined by ellipse with width rw and height rh

glsave flip w h - save drawing in WMF format in the glfile filename. Save to the clipboard if filename is empty.

flip 0 has J orientation of 0 0 as lower left corner.

flip 1 flips for applications like Word with 0 0 as upper left corner.

w h are suggested clipboard width and height in 0.01 millimeters.

The wmf file is saved without a "placeable metafile header". Some applications require this 22 byte header at the front of a wmf file. The utility `addwmfheader` from `script\system\main\winutil.js` adds a header to a wmf file so that it is suitable for use by WORD and similar applications.

glsavebmp width height - save drawing as bitmap (24bit color) to glfile filename. Save to clipboard if filename is empty.

glssel id - select isigraph child of currently selected parent for graphic commands

glssel hwndc - hwndc is the result of wd'qhwndc' for an isigraph control. Parent and child are selected as the wd and gl command targets.

glshow " - add new commands (since last glshow or glshowx) to the 2D paint buffer and immediately paints just the new commands. A glshow immediately after creating and showing a form can result in extra painting and flicker. The glshow paints the new stuff, which in most cases is everything, and then the normal window painting mechanism requests painting the new window and the painting will be done again. It is generally better to use glshowx and glpaintx.

glshowx " - add new commands to the 2D paint buffer with no explicit painting

gltext text - write text in the glfont font. Where and how the text is displayed is affected by the gltextalign and gltextxy commands.

gltextalign TA_value - the TA_value affects how gltext displays text

TA_LEFT TA_CENTER TA_RIGHT
TA_TOP TA_BASELINE TA_BOTTOM
TA_NOUPDATECP TA_UPDATECP

The first option in each group is the default. TA_NOUPDATECP causes gltext to use the position set by gltextxy. TA_UPDATECP causes gltext to use and update the current position.

Values from each of the 3 groups can be combined:

gltextalign TA_BOTTOM + TA_UPDATEDCP

gltextcolor " - glrgb color is set as color of text for gtext

gltextjustify space nblanks - space is the amount of space to insert for nblanks. This space per blank is inserted for EACH blank in subsequent gtext commands. Because of rounding error there may be unused space at the end of the text. To allow spacing multiple texts on the same line this unused space is carried over into the next qlgextent calculations. Justifying a line should be sure to start with this unused space set to 0. A gltextjustify 0 0 sets this unused space to 0. Each section of a line should be measured with glqextent and then a gltextjustify should be done with the required space and the number of blanks in that section of text, and finally the gltext should be done. At the end of the justifying a line it is a good practice to do a gltextjustify 0 0 to clear the unused space value and to be sure that subsequent text is not inadvertently spaced out.

gltext commands painted with glshow after gltextjustify commands have been processed will not be justified unless the entire control is repainted. glpaint forces a repaint.

The Macintosh runs the `gltextjustify` command without error, but text justification is not supported and the text is displayed without justification.

`gltextxy x y` - position for `gltext` with `gltexalign` of `TA_NOUPDATECP`

`gltop topvalue` - `topvalue` is in logical units and ties that `y` coordinate value to the top of the drawing. This can be used to make a window display the top part of an image, rather than the bottom part.

`glwantresize` - request resize event for `gl2` control.

`glwindowext x y` - default is 1000 1000. Change these values to change the scale (zoom) in anisotropic and isotropic mappings

`glwindoworg x y` - default is 0 0. Change these values to scroll the drawing in the viewport

`glzabortdoc "` - cancel the print job

`glzcreate printername` - name from the set of printers returned by `wd qprinters`. If no name is given then the session default printer is used. If a printer context already exists, it is deleted before creating the new context. The session default printer is the system default printer when `J` is started. This can be changed with `wd mbprinter`.

`glzdelete "` - deletes printer context created by `glzcreate`. There is no error if there is no context.

`glzdevmode "` - modifies printer document properties such as orientation and copies. The command `glqdevmode` returns the properties that can be changed by `glzdevmode` and their current values. You can determine arguments for `glzdevmode` by using `mbprinter` to create the desired printer document setup and then use `glqdevmode` to get those values. This command must be done before the `glzstartpage` command.

The Macintosh does not support `glzdevmode` and printer properties can not be changed.

The information returned is: orientation, papersize, paperlength, paperwidth, scale, copies, source, quality, color, duplex, yresolution, truetype

`glzenddoc "` - ends a document.

`glzendpage "` - ends a page.

`glzmap mode` - set mapping mode used in printing. Default is `twips`.

`glzprint "` - draws the image on the printer.

`glzstartdoc 'jobname [filename]'` - the job name can be up to 31 characters and appears in print manager. The filename indicates to print to a file instead of the printer. This file can be printed with the print command. The Macintosh does not support the second parameter of file name.

```
glzstartdoc "first job"
```

```
glzstartdoc "my job" tofile.prn'
```

glzstartpage " - starts a page. Each page must be bracketed by a glzstartpage and a glzendpage.

fontspec

A fontspec is several parameters that specify a font. A fontspec is used in some wd, gl2 and gl3 commands.

A fontspec is a font name, size, and optional styles:
name size [italic] [bold] [underline] [oem] [angle]

For example: "lucida console" 15 italic

A positive size gives cell height and negative gives character height. In both cases a font defined interline spacing is used. It is recommended to use cell height (positive) as this maps more directly and accurately to physical fonts. This is particularly true for printers.

The size is in logical units. For wd commands this is generally point size. In gl2, size is in mapping mode units. In gl3, size is in pixels.

Fonts can have different character sets. The standard Windows character set is ANSI and does not include boxes. The old PC character set is OEM and does contain boxes. Fonts such as ISIJ and MS Linedraw contain a single DEFAULT character set. The oem parameter selects an OEM font. The default parameter selects the DEFAULT font. If neither oem nor default is supplied, an ANSI font is selected. If there is not an exact match, you get an arbitrary font. There are differences in distributed fonts and in the way fonts are selected in different hosts.

isij 12 default	isij with boxes
"ms linedraw" 12 default	linedraw with boxes
"lucida console" 12 oem	console with boxes (if the font has oem)
"courier new" 12 oem	courier with boxes (if the font has oem)

You may have to experiment to find the best font for J boxes with your particular host, fonts, screen, and printer.

Fonts such as WINGDINGS, do not have an ANSI character set, and you need to specify default to get the default character set for the font.

```
glfont 'wingdings 100 default'
```

The angle style gives an angle in 10ths of degrees clockwise from the baseline. For example:

```
glfont 'arial 240 angle900'
```

isigraph events

An isigraph control supports character and mouse events. The Code Dialog in the form editor lists all events for an isigraph control.

char	character event
size	resize event (OpenGL, gl2 must request with glwantresize)
paint	paint event (OpenGL only)
mmove	mouse move event
mbldbl	mouse button left double-click
mbldown	mouse button left down
mblup	mouse button left up
mbrdbl	mouse button right double-click
mbrdown	mouse button right down
mbrup	mouse button right up

For mouse events the wdhandler variable sysdata contains:

x y width height leftbutton rightbutton ctrl shift

The x y width and height are in pixels and the other values are 1 if the corresponding button or key was down when the event occurred.

For a char event sysdata contains the value of the character. Characters, such as HOME, END, or the arrow keys are returned as 128+VK_name (virtual key) as defined in packages\graphics\vkeys.js.

Mapping Mode

gl2 command values are in logical units. The mapping mode affects how logical units are mapped onto the display surface.

There are 8 mapping modes:

mode	unit	x	y
MM_TEXT	pixel	right	down
MM_LOENGLISH	0.01 inch	right	up
MM_LOMETRIC	0.1 mm	right	up
MM_HIENGLISH	0.001 inch	right	up
MM_TWIPS	%1440 inch	right	up
MM_HIMETIRC	0.01 mm	right	up
MM_ISOTROPIC	x=y	?	?
MM_ANISOTROPIC	?	?	?

glmap sets the mapping mode. The default is anisotropic.

The modes from loenglish to himetric are referred to as metric modes.

The text mapping mode is not called text because it is for text, but because it has x y coordinates that go across and down as in reading English text. The text mode is different from the metric modes in that its y axis goes down, rather than up.

A twip is a "twentieth of a point". A point is approximately %72 of an inch and in computer systems a twip is considered to be exactly %1440 of an inch. This mode is particularly suited to laying out text and graphs for printing.

The isotropic and anisotropic modes can be used to scale logical units into the available device units. The isotropic mode preserves the aspect ratio of the image when it is scaled.

Mapping coordinates

The default behavior is a logical drawing surface that has 0 0 as its lower left corner and 1000 1000 as its upper right corner. This can be modified as described in this section, but if you want something different, it might be easier to use glmapraw as described in the next section.

The **windowport** is the drawing represented by the drawing commands. The **viewport** is the surface (screen window or printer paper) that the drawing is drawn on.

The origin of the windowport can be set to cause the drawing to scroll on the viewport.

The scaling done in the anisotropic and isotropic mode is done based on the ratio between the x and y extents of the viewport and windowport. It is the ratios of these extents that is important, not the individual values. The extent values are ignored in the other mapping modes.

The viewport (drawing surface) origin is fixed as 0 0 in the lower left corner. The viewport extent is fixed as the pixels in the x and y directions. These values can not be changed by commands. However, as the window on the screen is resized, the viewport extents change.

The default windowport (drawing) is 0 0 and the default windowport extent is 1000 1000. The `glwindoworg` command changes the origin and the `glwindowext` command changes the extent.

With a mapping mode of anisotropic these default values result in an image that is drawn with logical coordinates in the range 0 to 1000 scaled to fit the viewport. The lower left corner of the viewport is 0 0. The x axis starts at 0 and is positive to the right. The y axis starts at 0 and is positive upwards.

For the metric modes, this results in the drawing displayed with its lower left corner at the lower left corner of the viewport. The x axis starts at 0 and is positive to the right. The y axis starts at 0 and is positive upwards.

With text mode the Y axis is down and you must use negative y values to be seen in the viewport.

glmapraw

`glmapraw` turns off the default behavior described above. With `MM_TEXT`, drawing command values are in pixels and the drawing is directly to the pixels on the drawing surface. The resize event (`glwantresize`) can be used to redraw based on the new pixel size of the control.

Mapping and repainting

The mapping commands `glmap`, `glmapraw`, `glwindoworg`, `glwindowext`, and `gltop` change information on how to draw the image. They do not cause a repaint of the image.

Changing the mapping information requires repainting the image to see the change. A `glshow` with nothing new to show won't do this. A `glpaint` command forces repainting.

gl3 commands 11! : 3000+n

gl3 commands are for 3D (OpenGL) drawing in an isigraph control. Script system\main\gl3.ijs defines gl3 verbs.

```
load 'gl3' NB. defines verbs such as: glAccum=:11!:3001
```

The syntax of most OpenGL verbs should be clear from standard OpenGL API documentation, even though that documentation is oriented towards C programmers.

You are encouraged to start learning OpenGL programming with the **OpenGL Introduction** lab from the **Studio|Labs|** menu command.

This section documents the gla... commands that are not part of the OpenGL specification or where the syntax needs clarification. The gla... commands are concerned with how OpenGL fits into J or with Windows extensions.

Most 2D commands do not work in a 3D isigraph control. The following are the commands that do work: glshow, glshowx, glpaint, glpaintx, glqwh, glcapture.

OpenGL Printing shows how to print 3D images.

gl3 commands

glFont fontspec - font for glUseFontBitmaps and glUseFontOutlines

glGetErrors " - return recent errors reported by Quadric, Nurbs, and Tess error callbacks. The error callbacks must be enabled with the appropriate gluQuadricCallback, gluNurbsCallback, or gluTessCallback.

glARC " - create OpenGL render context for drawing on an isigraph window. Fails if:
control is in use by 2D graphics
unable to initialize the OpenGL dlls (opengl32.dll and glu32.dll)
control missing ws_clipchildren and ws_clipsiblings styles

glARC type w h - create OpenGL render context for drawing on a bitmap: type must be 1. w h give width and height of a 24bit color bitmap.

A bitmap RC is associated with an isigraph window but doesn't display anything in the window so normally this window should be hidden.

If the window is hidden, the `form_isigraph_size` and `form_isigraph_paint` handlers are not automatically called and they must be called explicitly.

The size handler for a bitmap RC should not use the window size (`glqwh'`) for setting the viewport. It should use the width and height used to create the bitmap in `glARC` .

glSaveBMP filename - save bitmap of an isigraph control with a bitmap RC. If the filename is '' , the bitmap is saved to the clipboard.

`glSwapBuffers` " - the front-buffer is the one displayed on the screen. Drawing is done into the back-buffer. A `glSwapBuffers` makes the back-buffer the front-buffer.

`glUseFontBitmaps` 0 first count listbase - create display lists based on `glFont`. A display list is created for count characters starting at first. The display list numbers start at listbase.

`glUseFontOutlines` 0 first count listbase deviation extrusion format - create display lists for drawing 3D characters based on `glFont`.

first - first of the set of characters

count - number of characters used to create display lists

listbase - starting display list

deviation - maximum chordal deviation from the original outlines. When deviation is zero, the chordal deviation is equivalent to one design unit of the original font. The value must be equal to or greater than 0.

extrusion - how much a font is extruded in the negative *z* direction. The value must be equal to or greater than 0.

format - `WGL_FONT_LINES` or `WGL_FONT_POLYGONS`

Returns a matrix with a row for each character. The row contains:

`BlackBoxX` - black box (smallest rectangle that contains the glyph) width

`BlackBoxY` - black box height

`OriginX` - black box upper-left x coordinate

`OriginY` - black box upper-left y coordinate

`IncX` - horizontal distance from current cell origin to the next cell

`IncY` - vertical distance from current cell origin to the next cell

`glBitmap` - argument is boxed list where the last argument is the data argument

`glDrawPixels` - argument is boxed list where the last argument is the data argument

`glCallLists` integer_data

`glCallLists` character_data

`glClearColor` R G B [A] - sets clear color red, green, blue, alpha. If A is elided, it is set to 1.

`glColor` R G B [A] - sets color red, green, blue, alpha. If A is elided it is set to 1.

`glFeedbackBuffer` integer type - feedback buffer size in floating values. This command is only allowed once. This avoids potential crashes with the buffer being changed at the wrong time.

`glGetError` " - return OpenGL error flags (there may be more than one). `gluErrorString` returns a string for an error number.

`glPixelStore` - `GL_UNPACK_ALIGNMENT` and `GL_PACK_ALIGNMENT` are the only pname arguments supported.

`glSelectBuffer` size - select buffer size in integers. This command is only allowed once. This avoids potential crashes with the buffer being changed at the wrong time.

The result data from the select buffer is recorded for commands done after a `glRenderMode GL_SELECT`. The result buffer is returned as the result of a `glRenderMode any_other_mode`. OpenGL defines the select buffer as returning minimum and maximum z values as unsigned integers.

unsigned int value - J signed int value
0 - smallest negative int
max 32 bit unsigned int - largest positive int

To convert the J signed integer range to a float range from 0 and up, add 2^{32} to the negative integers.

`glTexImage1d` - argument is boxed list where the last argument is the data argument

`glTexImage2d` - argument is boxed list where the last argument is the data argument

`glVertex X Y [Z [W]]` - sets vertex coordinates. If W (rarely used scaling value) is elided it is set to 1. If Z is elided it set to 0.

`gluBuild1dMipmaps` - argument is boxed list where the last argument is the data argument

`gluBuild2dMipmaps` - argument is boxed list where the last argument is the data argument

`gluPwlCurve nurb type data` - count and stride parameters are fixed based on type and length of data.

`gluNurbsCurve nurb uknots udata ustride uorder type data`

`gluNurbsCurve nurb uk ud vk vd ustride vstride uorder type data`

`gluScaleImage` - argument is boxed list where the last argument is the data argument

OpenGL printing

Printing OpenGL images is not directly supported. However, it is possible to get reasonably good printing of OpenGL images by creating a bitmap RC, drawing to the bitmap, and then printing the bitmap. If the bitmap is not stretched to much to fit the printing area the quality of the printed image is good.

The steps are as follows:

1. create temporary form and isigraph control
2. create a bitmap RC
3. size image - the size handler, but with bitmap size, not glqwh"
4. draw image - the paint handler
5. save the bitmap to file

Something like the following could be used to print an OpenGL image drawn on form a in isigraph control.

```
print=: verb define
wd 'pc p;cc g isigraph ws_clipchildren ws_clipsiblings;'
BMPSIZE=.2$1000          NB. big bitmap for little stretch
glaRC 1,BMPSIZE
NB. do initialization required by size and paint
size BMPSIZE           NB. a_g_size calls size with glqwh''
a_g_paint''
glaSaveBMP 't.bmp'
wd'pclose'
NB. print t.bmp file with 2D printing
)
```