Up

GSDisplayServer

Authors

Adam Fedor (fedor@gnu.org)

Abstract display server class.

Contents -

GSDisplayServer

This is an abstract class which provides a framework for a device independant window server. A window server handles the very basic control of the computer display and input. This includes basic window creation and handling, event handling, cursors, and providing miscellaneous information about the display.

Typically a backend library will provide a concrete subclass which implements the device specific methods described below.

In almost all cases, you should not call these methods directly in an application. You should use the equivalent methods available elsewhere in the library (e.g. NSWindow, NSScreen, etc).

GSDisplayServer : NSObject

Declared in:
GNUstepGUI/GSDisplayServer.h

Availability: OpenStep

Description forthcoming.

Instance Variables

Method summary

- -_printEventQueue
- -discardEventsMatchingMask:beforeEvent:
- -getEventMatchingMask:beforeDate:inMode:dequeue:
- -postEvent:atStart:
- -_setWindowOwnedByServer:
- -appOwnsMiniwindow
- -capturemouse:
- -docedited::
- -flushwindowrect::
- -freecursor:
- -hidecursor
- -imagecursor:::
- -miniwindow:

- -mouseLocationOnScreen:window:
- -mouselocation
- -movewindow::
- -nativeWindow:::::
- -orderwindow:::
- -placewindow::
- -recolorcursor:::
- -releasemouse
- -setIgnoreMouse::
- -setMouseLocation:onScreen:
- -setParentWindow:forChildWindow:
- -setShadow::
- -setWindowdevice:forContext:
- -setalpha::
- -setcursor:
- -setcursorcolor:::
- -setinputfocus:
- -setinputstate::
- -setmaxsize::
- -setminsize::
- -setresizeincrements::
- -setwindowlevel::
- -showcursor
- -standardcursor::
- -styleoffsets:::::
- -stylewindow::
- -termwindow:
- -titlewindow::
- -window:::
- -window::::
- -windowbacking::
- -windowbounds:
- -windowdepth:
- -windowdevice:
- -windowlevel:
- -windowlist
- +addDragTypes:toWindow:
- +dragTypesForWindow:
- +removeDragTypes:fromWindow:
- +serverWithAttributes:
- +setCurrentServer:
- +setDefaultServerClass:
- -addDragTypes:toWindow:
- -attributes
- -availableDepthsForScreen:
- -beep
- -boundsForScreen:
- -closeServer
- -contentsOfScreen:inRect:
- -dragInfo
- -dragTypesForWindow:
- -findWindowAt:windowRef:excluding:
- -glContextClass
- -glPixelFormatClass
- -handlesWindowDecorations
- -iconSize
- -iconTileImage
- -initWithAttributes:
- -removeDragTypes:fromWindow:
- -resolutionForScreen:
- -restrictWindow:toImage:
- -screenList
- -serverDevice
- -slideImage:from:to:
- -windowDepthForScreen:

- -windowDevice:

addDragTypes: toWindow:

+ (BOOL) **addDragTypes:** (NSArray*)types **toWindow:** (NSWindow*)win;
Availability: OpenStep

Convienience method that calls -addDragTypes:toWindow: using the server that controls *win*.

dragTypesForWindow:

+ (NSCountedSet*) **dragTypesForWindow:** (NSWindow*)win;
Availability: OpenStep

Convienience method that calls -dragTypesForWindow: using the server that controls *win*.

removeDragTypes: fromWindow:

+ (BOOL) **removeDragTypes:** (NSArray*)types **fromWindow:** (NSWindow*)win;
Availability: OpenStep

Convienience method that calls -removeDragTypes:fromWindow: using the server that controls *win*.

serverWithAttributes:

+ (GSDisplayServer*) **serverWithAttributes:** (NSDictionary*)attributes;
Availability: OpenStep

Create a window server with *attributes*, which contains key/value pairs which describe the specifics of how the window server is to be initialized. Typically these values are specific to the concrete implementation. The current set of *attributes* that can be used with GSDisplayServer is.

- GSDisplayName
- GSDisplayNumber
- GSScreenNumber

GSDisplayName is window server specific and shouldn't be used when creating a GSDisplayServer (although you can retrieve the value with the -attributes method). On X-Windows the value might be set to something like "host:d.s" where host is the host name, d is the display number and s is the screen number. GSDisplayNumber indicates the number of the display to open. GSScreenNumber indicates the number of the screen to display on. If not explicitly set, these *attributes* may be taked from environment variables or from other operating specific information.

In almost all applications one would only create a single instance of a window server. Although it is possible, it is unlikely that you would need more than one window server (and you would have to be very careful how you handled window creation and events in this case).

setCurrentServer:

+ (void) **setCurrentServer:** (GSDisplayServer*)server;
Availability: OpenStep

Sets the current *server* that will be handling windows, events, etc. This method must be called after a window *server* is created in order to make it available to the rest of the GUI library

setDefaultServerClass:

+ (void) **setDefaultServerClass:** (Class)aClass;
Availability: OpenStep

Set the concrete subclass that will provide the device dependant implementation.

---

addDragTypes: toWindow:

- (BOOL) **addDragTypes:** (NSArray*)types **toWindow:** (NSWindow*)win;
Availability: OpenStep

Add (increment count by 1) each drag type to those registered for the window. If this results in a change to the *types* registered in the counted set, return `YES`, otherwise return `NO`. Subclasses should override this method, call 'super' and take appropriate action if the method returns 'YES'.

---

attributes

- (NSDictionary*) **attributes**;
Availability: OpenStep

Return information used to create the server

---

availableDepthsForScreen:

- (const NSWindowDepth*) **availableDepthsForScreen:** (int)screen;
Availability: OpenStep

Returns a null terminated list of possible window depths for *screen*.

---

beep

- (void) **beep**;
Availability: OpenStep

Play the System Beep

---

boundsForScreen:

- (NSRect) **boundsForScreen:** (int)screen;
Availability: OpenStep

Returns the bounds, in pixels, for the indicated *screen* of the display.

---

closeServer

- (void) **closeServer**;
Availability: OpenStep

Causes the server to disconnect from the display. If the receiver is the current server, it removes itself and sets the current server to `nil`. Sending any more messages to the receiver after this is likely to cause severe problems and probably crash the application.

---

contentsOfScreen: inRect:

- (NSImage*) **contentsOfScreen:** (int)screen **inRect:** (NSRect)rect;
Availability: OpenStep

Returns a screenshot of the specified rectangle of the specified *screen*. The mouse cursor should be ommitted from the returned image.

dragInfo

- (id) **dragInfo**;
Availability: OpenStep

Returns an instance of a class which implements the NSDraggingInfo protocol.

dragTypesForWindow:

- (NSCountedSet*) **dragTypesForWindow:** (NSWindow*)win;
Availability: OpenStep

Returns the drag types set for the window *win*.

findWindowAt: windowRef: exc luding:

- (int) **findWindowAt:** (NSPoint)screenLocation **windowRef:** (int*)windowRef **excluding:** (int)win;
Availability: OpenStep

Description forthcoming.

glContextClass

- (id) **glContextClass**;
Availability: OpenStep

Description forthcoming.

glPixelFormatClass

- (id) **glPixelFormatClass**;
Availability: OpenStep

Description forthcoming.

handlesWindowDecorations

- (BOOL) **handlesWindowDecorations**;
Availability: OpenStep

Returns `YES` if the backend handles window decorations and `NO` if the gui library must do that instead.

iconSize

- (NSSize) **iconSize**;
Availability: OpenStep

Returns the size of icons and miniwindows for screen.

iconTileImage

- (NSImage*) **iconTileImage**;

Availability: OpenStep

Returns a display dependent NSImage which will be used as the background image for AppIcons and MiniWindows. Under Windowmaker, for example this could be a user specified gradient.

initWithAttributes:

- (id) **initWithAttributes:** (NSDictionary*)attributes;
Availability: OpenStep

This is a designated initialiser for the class.
Initializes the server. This typically causes the receiver to *connect* to the display (e.g. XOpenDisplay () on an X-Windows server).

removeDragTypes: fromWindow:

- (BOOL) **removeDragTypes:** (NSArray*)types **fromWindow:** (NSWindow*)win;
Availability: OpenStep

Remove (decrement count by 1) each drag type from those registered for the window. If this results in a change to the *types* registered in the counted set, return `YES`, otherwise return `NO`. If given 'nil' as the array of *types*, remove ALL. Subclasses should override this method, call 'super' and take appropriate action if the method returns 'YES'.

resolutionForScreen:

- (NSSize) **resolutionForScreen:** (int)screen;
Availability: OpenStep

Returns the resolution, in points, for the indicated *screen* of the display.

restrictWindow: toImage:

- (void) **restrictWindow:** (int)win **toImage:** (NSImage*)image;
Availability: OpenStep

Description forthcoming.

screenList

- (NSArray*) **screenList**;
Availability: OpenStep

Returns an array of NSNumbers, where each number describes a screen that is available on this display. The default screen is listed first.

serverDevice

- (void*) **serverDevice**;
Availability: OpenStep

Returns a display dependant pointer that describes the internal connection to the display. On X-Windows, for example, this is a pointer to the `Display` variable.

slideImage: from: to:

- (BOOL) **slideImage:** (NSImage*)image **from:** (NSPoint)from **to:** (NSPoint)to;
Availability: OpenStep

Description forthcoming.

windowDepthForScr een:

- (NSWindowDepth) **windowDepthForScreen:** (int)screen;
Availability: OpenStep

Returns the default depth of windows that are created on *screen*.

windowDevice:

- (void*) **windowDevice:** (int)win;
Availability: OpenStep

Returns a display dependant pointer that describes the internal window representation for *win*. On X-Windows, for example, this is a pointer to the `Window` variable.

Instance Variables for GSDisplayServer Class

drag_types

@protected NSMapTable* **drag_types**;
Availability: OpenStep

Description forthcoming.

event_queue

@protected NSMutableArray* **event_queue**;
Availability: OpenStep

Description forthcoming.

server_info

@protected NSMutableDictionary* **server_info**;
Availability: OpenStep

Description forthcoming.

Software documentation for the GSDisplayServer(EventOps) category

GSDisplayServer(EventOps)

Declared in:
GNUstepGUI/GSDisplayServer.h

Availability: OpenStep

Description forthcoming.
Method summary

- - _printEventQueue
- -discardEventsMatchingMask:beforeEvent:
- -getEventMatchingMask:beforeDate:inMode:dequeue:
- -postEvent:atStart:

_printEventQueue

- (void) **_printEventQueue**;
Availability: OpenStep

Warning the underscore at the start of the name of this method indicates that it is private, for internal use only, and you should not use the method in your code.

discardEventsMatchingMask: befor eEvent:

- (void) **discardEventsMatchingMask:** (unsigned)mask **beforeEvent:** (NSEvent*)limit;
Availability: OpenStep

Steps through the event queue and removes all events whose timestamp is earlier than that of *limit* wand which match the supplied *mask* of event types.

getEventMatchingMask: befor eDate: inMode: dequeue:

- (NSEvent*) **getEventMatchingMask:** (unsigned)mask **beforeDate:** (NSDate*)limit **inMode:** (NSString*)mode **dequeue:** (BOOL)flag;
Availability: OpenStep

Scans through the event queue to find the first event whose type matches *mask*. If no event is found, then the current run loop is run in the specified *mode* to allow more events to arrive.
If a matching event is found, it is returned and either removed from or left in the queue according to *flag*.
If no matching event is found and the *limit* date is reached, this method returns `nil`.

postEvent: atStart:

- (void) **postEvent:** (NSEvent*)anEvent **atStart:** (BOOL)flag;
Availability: OpenStep

Posts an event to the event queue. The value of *flag* determines whether the event is inserted at the start of the queue or appended at the end.

Software documentation for the GSDisplayServer(WindowOps) category

GSDisplayServer(WindowOps)

Declared in:
GNUstepGUI/GSDisplayServer.h

Availability: OpenStep

Description forthcoming.
Method summary

- -_setWindowOwnedByServer:
- -appOwnsMiniwindow
- -capturemouse:
- -docedited::
- -flushwindowrect::
- -freecursor:
- -hidecursor
- -imagecursor:::
- -miniwindow:
- -mouseLocationOnScreen:window:
- -mouselocation
- -movewindow::
- -nativeWindow:::::
- -orderwindow:::
- -placewindow::
- -recolorcursor:::
- -releasemouse
- -setIgnoreMouse::
- -setMouseLocation:onScreen:
- -setParentWindow:forChildWindow:
- -setShadow::
- -setWindowdevice:forContext:
- -setalpha::
- -setcursor:
- -setcursorcolor:::
- -setinputfocus:
- -setinputstate::
- -setmaxsize::
- -setminsize::
- -setresizeincrements::
- -setwindowlevel::
- -showcursor
- -standardcursor::
- -styleoffsets:::::
- -stylewindow::
- -termwindow:
- -titlewindow::
- -window:::
- -window::::
- -windowbacking::
- -windowbounds:
- -windowdepth:
- -windowdevice:
- -windowlevel:
- -windowlist

_setWindowOwnedByServer:

- (void) **_setWindowOwnedByServer:** (int)win;
Availability: OpenStep

Warning the underscore at the start of the name of this method indicates that it is private, for internal use only, and you should not use the method in your code.
Tells the receiver that it owns the window described by *win*. Concrete subclasses must call this function when creating a window. Do not call this method in any other case, particularly for a window that has already been created

appOwnsMiniwindow

- (BOOL) **appOwnsMiniwindow**;
Availability: OpenStep

Returns `YES` if the application should create the miniwindow counterpart to the full size window and own it. Some display systems handle the miniwindow themselves. In this case the backend subclass should override this method to return `NO`.

---

capturemouse:

- (BOOL) **capturemouse:** (int)win;
Availability: OpenStep

Grabs the pointer device so that all future mouse events will be directed only to the window *win*. If successful, the return value is `YES` and this message must be balanced by a -releasemouse message.

---

docedited: :

- (void) **docedited:** (int)edited **:** (int)win;
Availability: OpenStep

Sets the document *edited* flag for the window

---

flushwindowrect: :

- (void) **flushwindowrect:** (NSRect)rect **:** (int)win;
Availability: OpenStep

Causes buffered graphics to be flushed to the screen. The value of *rect* is expressed in OpenStep window coordinates.

---

freecursor:

- (void) **freecursor:** (void*)cid;
Availability: OpenStep

Free the cursor given by the *cid* representation.

---

hidecursor

- (void) **hidecursor**;
Availability: OpenStep

Hides the cursor

---

imagecursor: : :

- (void) **imagecursor:** (NSPoint)hotp **:** (NSImage*)image **:** (void**)cid;
Availability: OpenStep

Create a cursor from an *image*. Returns a pointer to the internal device representation that can be used later to make this cursor the current one

---

miniwindow:

- (void) **miniwindow:** (int)win;
Availability: OpenStep

Miniaturizes the window

---

mouseLocationOnScreen: window:

- (NSPoint) **mouseLocationOnScreen:** (int)aScreen **window:** (int*)win;
Availability: OpenStep

Returns the current mouse location on *aScreen*. If the pointer is not on *aScreen*, this method acts like -mouselocation . If *aScreen* is -1, then the location of the mouse on any screen is returned. The *win* pointer returns the window number of the GNUstep window that the mouse is in or 0 if it is not in a window.
The location is expressed as an offset from the bottom left corner of the screen.

---

mouselocation

- (NSPoint) **mouselocation**;
Availability: OpenStep

Returns the current mouse location on the default screen. If the pointer is not on the default screen, an invalid point (-1,-1} is returned.
The location is expressed as an offset from the bottom left corner of the screen.

---

movewindow: :

- (void) **movewindow:** (NSPoint)loc **:** (int)win;
Availability: OpenStep

Moves the bottom left corner of the window (including any border) to *loc*.
The position is expressed as an offset from the bottom left corner of the screen.

---

nativeWindow: : : : :

- (int) **nativeWindow:** (void*)winref **:** (NSRect*)frame **:** (NSBackingStoreType*)type **:** (unsigned int*)style **:** (int*)screen;
Availability: OpenStep

Create all the backend structures for a reference to a native window and return the extend, backing *type*, *style* and *screen* for that window.

---

orderwindow: : :

- (void) **orderwindow:** (int)op **:** (int)otherWin **:** (int)win;
Availability: OpenStep

Causes the window to be ordered onto or off the screen depending on the value of *op*. The window is ordered relative to *otherWin* .

The effect of the various combinations of *op* and *otherWin* are:

op is NSWindowOut
    The window is removed from the display and otherWinm is ignored.
op is NSWindowAbove and *otherWin* is zero
    The window is placed above all other windows at the same level unless doing the current key window is at this level (in which case the window will be placed immediately below that).
op is NSWindowAbove and *otherWin* is minus one

The window is placed above all other windows at the same level even if doing that would place it above the current key window.
This is a special feature that [NSWindow -orderWindow:relativeTo:] uses to place the window correctly.

op is NSWindowBelow and *otherWin* is zero
The window is placed above all other windows at the same level.

op is NSWindowAbove and *otherWin* is a window on the display
The level of the window is set to be the same as that of *otherWin* and the window is placed immediately above *otherWin*.

op is NSWindowBelow and *otherWin* is a window on the display
The level of the window is set to be the same as that of *otherWin* and the window is placed immediately below *otherWin*.

---

placewindow: :

- (void) **placewindow:** (NSRect)frame **:** (int)win;
Availability: OpenStep

Moves and resizes the window on the screen as described by *frame*. The value of *frame* is a rectangle containing the entire window, including any border/decorations. Its position is expressed as an offset from the bottom left corner of the screen.

---

recolorcursor: : :

- (void) **recolorcursor:** (NSColor*)fg **:** (NSColor*)bg **:** (void*)cid;
Availability: OpenStep

Recolour the cursor given by the *cid* representation into having a foreground color *fg* and a background color *bg*.

---

releasemouse

- (void) **releasemouse**;
Availability: OpenStep

Release a previous captured mouse from -capturemouse:

---

setIgnoreMouse: :

- (void) **setIgnoreMouse:** (BOOL)ignoreMouse **:** (int)win;
Availability: OpenStep

Description forthcoming.

---

setMouseLocation: onScreen:

- (void) **setMouseLocation:** (NSPoint)mouseLocation **onScreen:** (int)aScreen;
Availability: OpenStep

Set mouse cursor position.

---

setParentWindow: forChildWindow:

- (void) **setParentWindow:** (int)parentWin **forChildWindow:** (int)childWin;
Availability: OpenStep

Description forthcoming.

---

setShadow: :

- (void) **setShadow:** (BOOL)hasShadow **:** (int)win;
Availability: OpenStep

Sets the window shadow

setWindowdevice: forContext:

- (void) **setWindowdevice:** (int)win **forContext:** (NSGraphicsContext*)ctxt;
Availability: OpenStep

Sets the window device information for the NSGraphicsContext, typically by calling [NSGraphicsContext -GSSetDevice:::] , although depending on the concrete implmentation, more information than this may need to be exchanged.

setalpha: :

- (void) **setalpha:** (float)alpha **:** (int)win;
Availability: OpenStep

Sets the transparancy value for the whole window

setcursor:

- (void) **setcursor:** (void*)cid;
Availability: OpenStep

Set the cursor given by the *cid* representation as being the current cursor.

setcursorcolor: : :

- (void) **setcursorcolor:** (NSColor*)fg **:** (NSColor*)bg **:** (void*)cid;
Availability: OpenStep

Set the cursor given by the *cid* representation as being the current cursor. The cursor has a foreground color *fg* and a background color *bg*. To keep the default color for the cursor, pass `nil` for *fg* and *bg*.

setinputfocus:

- (void) **setinputfocus:** (int)win;
Availability: OpenStep

Forces focus to the window so that all key events are sent to this window

setinputstate: :

- (void) **setinputstate:** (int)state **:** (int)win;
Availability: OpenStep

Sets the input *state* for the window given by the GSWindowInputState constant. Instructs the window manager that the specified window is 'key', 'main', or just a normal window.

setmaxsize: :

- (void) **setmaxsize:** (NSSize)size **:** (int)win;
Availability: OpenStep

Set the maximum *size* (pixels) of the window

setminsize: :

- (void) **setminsize:** (NSSize)size **:** (int)win;
Availability: OpenStep

Set the minimum *size* (pixels) of the window

setresizeincrements: :

- (void) **setresizeincrements:** (NSSize)size **:** (int)win;
Availability: OpenStep

Set the resize incremenet of the window

setwindowlevel: :

- (void) **setwindowlevel:** (int)level **:** (int)win;
Availability: OpenStep

Set the *level* of the window as in the [NSWindow -setLevel] method.
The use of window levels organises the window hierarchy into groups of windows at each *level*. It effects the operation of the -orderwindow::: method in the case where the position is 'above' or 'below' and the other window number is zero.

showcursor

- (void) **showcursor**;
Availability: OpenStep

Show a previously hidden cursor

standardcursor: :

- (void) **standardcursor:** (int)style **:** (void**)cid;
Availability: OpenStep

Create a standard cursor (such as an arror or IBeam). Returns a pointer to the internal device representation that can be used later to make this cursor the current one

styleoffsets: : : : :

- (void) **styleoffsets:** (float*)l **:** (float*)r **:** (float*)t **:** (float*)b **:** (unsigned int)style;
Availability: OpenStep

Returns the dimensions of window decorations added outside the drawable window frame by a window manager or equivalent. For instance, *t* gives the height of the title bar for the window.
If -handlesWindowDecorations returns `NO`, there are no decorations outside the drawable window frame and this method shouldn't be called.

stylewindow: :

- (void) **stylewindow:** (unsigned int)style **:** (int)win;
Availability: OpenStep

Sets the *style* of the window. See [NSWindow -styleMask] for a description of the available styles

termwindow:

- (void) **termwindow:** (int)win;
Availability: OpenStep

Destroys the representation of the window and frees and memory associated with it.

titlewindow: :

- (void) **titlewindow:** (NSString*)window_title **:** (int)win;
Availability: OpenStep

Sets the window title

window: : :

- (int) **window:** (NSRect)frame **:** (NSBackingStoreType)type **:** (unsigned int)style;
Availability: OpenStep

Creates a window whose location and size is described by *frame* and whose backing store is described by *type*. This window is not mapped to the screen by this call.
Note that *frame* is the *frame* of the entire GNUstep window including borders, titlebar and other standard decorations.
If -handlesWindowDecorations returns `YES`, the backend will produce (and return the identifier of) a smaller drawable window inside this decorated area.
Use -styleoffsets::::: to determine the extent of the decorations and determine the size of the drawable area inside them.

window: : : :

- (int) **window:** (NSRect)frame **:** (NSBackingStoreType)type **:** (unsigned int)style **:** (int)screen;
Availability: OpenStep

Like -window::: only there is an additional argument to specify which *screen* the window will display on

windowbacking: :

- (void) **windowbacking:** (NSBackingStoreType)type **:** (int)win;
Availability: OpenStep

Changes window's the backing store to *type*

windowbounds:

- (NSRect) **windowbounds:** (int)win;
Availability: OpenStep

Returns the frame of the window on the screen.
The value of frame is a rectangle containing the entire window, including any border/decorations. Its position is expressed as an offset from the bottom left corner of the screen.

windowdepth:

- (int) **windowdepth:** (int)win;
Availability: OpenStep

Returns the depth of the window

windowdevice:

- (void) **windowdevice:** (int)winNum;
Availability: OpenStep

Sets the window device information for the current NSGraphicsContext, typically by calling [NSGraphicsContext -GSSetDevice:::] , although depending on the concrete implmentation, more information than this may need to be exchanged.

windowlevel:

- (int) **windowlevel:** (int)win;
Availability: OpenStep

Returns the window level as in [NSWindow -level]

windowlist

- (NSArray*) **windowlist**;
Availability: OpenStep

Backends can override this method to return an array of window numbers ordered front to back. The front most window being the first object in the array. The default implementation returns the visible windows in an unspecified order.

Up