

Defined Types

DPSTextContextRec

DECLARED IN dpsclient/dpsfriends.h

SYNOPSIS

typedef struct _t_DPSTextContextRec {
 char *priv;
 DPSSpace space;
 DPSTextProgramEncoding programEncoding;
 DPSTextNameEncoding nameEncoding;
 struct _t_DPSTextProcsRec const * procs;
 void (*textProc)();
 void (*errorProc)();
 DPSTextResults resultTable;
 unsigned int resultTableLength;
 struct _t_DPSTextContextRec *chainParent, *chainChild;
 DPSTextContextType type;
} DPSTextContextRec, *DPSTextContext;

DESCRIPTION The **DPSTextContextRec** structure represents a Display PostScript context.

DPSTextContextType

DECLARED IN dpsclient/dpsfriends.h

SYNOPSIS

typedef enum {
 dps_machServer,
 dps_fdServer,
 dps_stream
} DPSTextContextType;

DESCRIPTION These represent the context types supported by NeXT's version of Display PostScript, as used in the **type** field of a **DPSTextContextRec** structure.

DPSErrorCode

DECLARED IN dpsclient/dpsclient.h

SYNOPSIS

typedef enum _DPSErrorCode {
 dps_err_ps = DPS_ERROR_BASE,
 dps_err_nameTooLong,
 dps_err_resultTagCheck,
 dps_err_resultTypeCheck,
 dps_err_invalidContext,

DESCRIPTION	Error codes passed to a DPSErrorProc() function.
--------------------	---

DECLARED IN `dpsclient/dpsNeXT.h`

DESCRIPTION	Call-back function used to filter events.
--------------------	---

DECLARED IN `dpsclient/dpsNeXT.h`

DESCRIPTION	Call-back function used when a file descriptor is registered through DPSAddFD() .
--------------------	--

DECLARED IN `dpsclient/dpsNeXT.h`

DESCRIPTION These constants are used by the **DPSDoUserPath()** function to describe the type of numbers that are being passed.


```

    dps_ueofill,
    dps_ustroke,
    dps_ustrokepath,
    dps_inufill,
    dps_inueofill,
    dps_inustroke,
    dps_def,
    dps_put
} DPSUserPathAction;

```

DESCRIPTION These constants are convenient representations of some of the PostScript operator indices, suitable for enrollment in the action array passed to **DPSDoUserPath()**.

DPSUserPathOp

DECLARED IN `dpsclient/dpsNeXT.h`

```

SYNOPSIS
    dps_setbbox,
    dps_moveto,
    dps_rmoveto,
    dps_lineto,
    dps_rlineto,
    dps_curveto,
    dps_rcurveto,
    dps_arc,
    dps_arcn,
    dps_arct,
    dps_closepath,
    dps_ucache
} DPSUserPathOp;

```

DESCRIPTION These constants represent the PostScript operators that can be passed in **DPSDoUserPath()**'s operator array.

NXCoord

DECLARED IN `dpsclient/event.h`

SYNOPSIS `typedef float NXCoord`

DESCRIPTION	Used to represent a single coordinate in a Cartesian coordinate system.
--------------------	---

NXEvent

DECLARED IN `dpsclient/event.h`

```

SYNOPSIS                                     typedef struct _NXEvent {
    int type;
    NXPoint location;
    long time;
    int flags;

```

```
        unsigned int window;  
        NXEventData data;  
        DPSContext ctxt;  
    } NXEvent, *NXEventPtr;
```

DESCRIPTION Represents a single event; this structure is also known as the *event record*. The fields are:

type	The type of event (see "Event Types," below)
location	The event's location in the base coordinate system of its window
time	The time of the event (in hardware-dependent units) since system startup
flags	Mouse-button and modifier-key flags (see "Event Flags," below)
window	The window number of the window associated with the event
data	Additional type-specific data (see "NXEventData," below)
ctxt	The PostScript context of the event

NXEventData

DECLARED IN dpsclient/event.h

SYNOPSIS

```
struct {  
    short eventNum;  
    int click;  
    unsigned char pressure;  
} mouse;  
struct {  
    short repeat;  
    unsigned short charSet;  
    unsigned short charCode;  
    unsigned short keyCode;  
    short keyData;  
} key;  
struct {  
    short eventNum;  
    int trackingNum;  
    int userData;  
} tracking;  
struct {  
    short subtype;  
    union {  
        float F[2];  
        long L[2];  
        short S[4];  
        char C[8];  
    } misc;  
} compound;  
} NXEventData;
```

typedef union {

DESCRIPTION This structure supplies type-specific information for an event. It's a union of four structures, where the type of the event determines which structure is pertinent:

- **mouse** is used for mouse events.
- **key** is used for keyboard events.
- **tracking** is for tracking-rectangle events.
- **compound** is for system-, kit-, and application-defined events.

DECLARED IN

dpsclient/event.h

SYNOPSIS

NX_ASCIISET
NX_SYMBOLSET
NX_DINGBATSSET

DESCRIPTION

These constants represent the values that may occur in the **data.key.charSet** field of an NXEvent structure.

Compositing Operations

DECLARED IN

dpsclient/dpsNeXT.h

SYNOPSIS

NX_CLEAR
NX_COPY
NX_SOVER
NX_SIN
NX_SOUT
NX_SATOP
NX_DOVER
NX_DIN
NX_DOUT
NX_DATOP
NX_XOR
NX_PLUSD
NX_HIGHLIGHT
NX_PLUSL

DESCRIPTION

These represent the compositing operations used by **PScomposite()** and the NXImage class.

Error Code Bases

DECLARED IN

dpsclient/dpsclient.h

SYNOPSIS

DPS_ERROR_BASE
DPS_NEXT_ERROR_BASE

DESCRIPTION

These constants represent the lowest values for Display PostScript error codes.

Event Types

DECLARED IN	dpsclient/event.h	
SYNOPSIS		
Meaning		Type
NX_NULLEVENT	A non-event	
NX_LMOUSEDOWN	Left mouse-down	
NX_LMOUSEUP	Left mouse-up	
NX_LMOUSEDRAGGED	left mouse-dragged	
NX_MOUSEDOWN	Same as NX_LMOUSEDOWN	

Forever

DECLARED INdpsclient/dpsNeXT.h

SYNOPSISNX_FOREVER

DESCRIPTIONA long, long time. Typically used as the timeout argument to **DPSGetEvent()**.

Keyboard State Flags Masks

DECLARED INdpsclient/event.h

SYNOPSIS	Type
Meaning	
NX_ALPHASHIFTMASK	Shift lock
NX_SHIFTMASK	Shift key
NX_CONTROLMASK	Control key
NX_ALTERNATEMASK	Alt key
NX_COMMANDMASK	Command key
NX_NUMERICPADMASK	Number pad key
NX_HELPMASK	Help key
NX_NEXTCTRLKEYMASK	Control key
NX_NEXTLSHIFTKEYMASK	Left shift key
NX_NEXTRSHIFTKEYMASK	Right shift key
NX_NEXTLCMDKEYMASK	Left command key
NX_NEXTRCMDKEYMASK	Right command key
NX_NEXTLALTKEYMASK	Left alt key
NX_NEXTRALTKEYMASK	Right alt key

DESCRIPTIONThese masks correspond to keyboard states that might be included in an NXEvent structure's **flags** mask. The masks are grouped as device-independent (NX_ALPHASHIFTMASK through NX_HELPMASK) and device-dependent (all others).

Miscellaneous Event Flags Masks

DECLARED INdpsclient/event.h

SYNOPSIS	Type
Meaning	
NX_STYLUSPROXIMITYMASK	Stylus is in proximity (for tablets)
NX_NONCOALSESCEDMASK	Event coalescing disabled

DESCRIPTIONThese masks correspond to miscellaneous states that might be included in an NXEvent structure's **flags** mask.

Window Backing Types

DECLARED INdpsclient/dpsNeXT.h

SYNOPSIS

NX_NONRETAINED
NX_BUFFERED

NX_RETAINED

DESCRIPTION

These represent the three backing types provided by window devices (and used by the Application Kit's Window objects).

Window Screen List Placement

DECLARED IN

dpsclient/dpsNeXT.h

SYNOPSIS

NX_BELOW
NX_OUT

NX_ABOVE

DESCRIPTION

These represent the placement of a window device in the screen list.