

Types and Constants

Defined Types

DPSTextContextRec

```
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;
```

DPSTextContextType

```
typedef enum {  
    dps_machServer,  
    dps_fdServer,  
    dps_stream  
} DPSTextContextType;
```

DPSErrorCode

```
typedef enum _DPSErrorCode {  
    dps_err_ps = DPS_ERROR_BASE,  
    dps_err_nameTooLong,  
    dps_err_resultTagCheck,  
    dps_err_resultTypeCheck,  
    dps_err_invalidContext,  
    dps_err_select = DPS_NEXT_ERROR_BASE,  
    dps_err_connectionClosed,  
    dps_err_read,  
    dps_err_write,  
    dps_err_invalidFD,  
    dps_err_invalidTE,  
    dps_err_invalidPort,  
    dps_err_outOfMemory,  
    dps_err_cantConnect  
} DPSErrorCode;
```

DPSEventFilterFunc

```
typedef int (*DPSEventFilterFunc)(NXEvent *ev );
```

DPSFDProc

```
typedef void (*DPSFDProc)( int fd, void *userData );
```

DPSNumberFormat

```
typedef enum _DPSNumberFormat {  
#ifdef __BIG_ENDIAN__  
    dps_float = 48,  
    dps_long = 0,  
    dps_short = 32  
#else  
    dps_float = 48+128,  
    dps_long = 0+128,  
    dps_short = 32+128  
} DPSNumberFormat;
```

DPSPingProc

```
typedef void (*DPSPingProc)  
    (DPSTimedEntry ctxt,  
    void *userData);
```

DPSPortProc

```
typedef void (*DPSPortProc)  
    ( msg_header_t *msg,  
    void *userData );
```

DPSTimedEntry

```
typedef struct __DPSTimedEntry *DPSTimedEntry;
```

DPSTimedEntryProc

```
typedef void (*DPSTimedEntryProc)  
    (DPSTimedEntry timedEntry,  
    double now,  
    void *userData );
```

DPSUserPathAction

```
typedef enum _DPSUserPathAction {  
    dps_uappend = 176,  
    dps_ufill = 179,  
    dps_ueofill = 178,  
    dps_ustroke = 183,  
    dps_ustrokepath = 364,  
    dps_inufill = 93,  
    dps_inueofill = 92,  
    dps_inustroke = 312,  
    dps_def = 51,  
    dps_put = 120  
} DPSUserPathAction;
```

DPSUserPathOp

```
typedef enum _DPSUserPathOp {
    dps_setbbox = 0,
    dps_moveto,
    dps_rmoveto,
    dps_lineto,
    dps_rlineto,
    dps_curveto,
    dps_rcurveto,
    dps_arc,
    dps_arcn,
    dps_arct,
    dps_closepath,
    dps_ucache
} DPSUserPathOp;
```

NXCoord

```
typedef float NXCoord
```

NXEvent

```
typedef struct _NXEvent {
    int type;
    NXPoint location;
    long time;
    int flags;
    unsigned int window;
    NXEventData data;
    DPSTextContext ctxt;
} NXEvent, *NXEventPtr;
```

NXEventData

```
typedef union {
    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 reserved;
        short subtype;
    } reserved;
}
```

```
        union {
            float F[2];
            long L[2];
            short S[4];
            char C[8];
        } misc;
    } compound;
} NXEventData;
```

NXPoint

```
typedef struct _NXPoint {
    NXCoord x;
    NXCoord y;
} NXPoint;
```

NXSize

```
typedef struct _NXSize {
    NXCoord width;
    NXCoord height;
} NXSize;
```

Symbolic Constants

All Contexts

DPS_ALLCONTEXTS

Alpha Constants

NX_DATA
NX_ONES

Character Set Values

NX_ASCIISET
NX_SYMBOLSET
NX_DINGBATSSET

Compositing Operations

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

Error Code Bases

DPS_ERROR_BASE
DPS_NEXT_ERROR_BASE

Event Types

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
NX_MOUSEUP	Same as NX_LMOUSEUP
NX_MOUSEDRAGGED	Same as NX_LMOUSEDRAGGED
NX_RMOUSEDOWN	Right mouse-down
NX_RMOUSEUP	Right mouse-up
NX_RMOUSEDRAGGED	Right mouse-dragged
NX_MOUSEMOVED	Mouse-moved
NX_MOUSEENTERED	Mouse-entered
NX_MOUSEEXITED	Mouse-exited
NX_KEYDOWN	Key-down
NX_KEYUP	Key-up event
NX_FLAGSCHANGED	Flags-changed
NX_KITDEFINED	Appkit-defined
NX_SYSDEFINED	System-defined
NX_APPDEFINED	Application-defined
NX_TIMER	Timer used for tracking
NX_CURSORUPDATE	Cursor tracking
NX_JOURNALEVENT	Event used by journaling
NX_FIRSTEVENT	The smallest-valued event constant
NX_LASTEVENT	The greatest-valued event constant
NX_ALLEVENTS	A value that includes all event types

Meaning

Event Type Masks

NX_NULLEVENTMASK
NX_LMOUSEDOWNMASK
NX_LMOUSEUPMASK
NX_RMOUSEDOWNMASK
NX_RMOUSEUPMASK
NX_MOUSEMOVEDMASK
NX_LMOUSEDRAGGEDMASK
NX_RMOUSEDRAGGEDMASK
NX_MOUSEENTEREDMASK
NX_MOUSEEXITEDMASK
NX_KEYDOWNMASK
NX_KEYUPMASK
NX_FLAGSCHANGEDMASK
NX_KITDEFINEDMASK
NX_APPDEFINEDMASK
NX_SYSDEFINEDMASK
NX_TIMERMASK
NX_CURSORUPDATEMASK
NX_MOUSEDOWNMASK
NX_MOUSEUPMASK
NX_MOUSEDRAGGEDMASK

NX_JOURNALEVENTMASK

Forever

NX_FOREVER

Keyboard State Flags Masks

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

Event Flags Masks

Meaning

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

Window Backing Types

- NX_RETAINED
- NX_NONRETAINED
- NX_BUFFERED

Window Screen List Placement

- NX_ABOVE
- NX_BELOW
- NX_OUT