NAME

PEXPickAll - Pick All Traversal

SYNTAX

PEXPickPath *PEXPickAll(Display **display*, Drawable *drawable*, PEXRenderer *renderer*, int *method*, int *max_hits*, int *pick_device_type*, PEXPickRecord **pick_record*, int **status_return*, int **more_return*, unsigned long **count_return*)

PARAMETERS

display	A pointer to a display structure returned by a successful **XOpenDisplay** call.
drawable	The resource identifier of a drawable.
renderer	The resource identifier of the renderer.
method	The pick all method (**PEXPickAllAll** or **PEXPickAllVisible**).
max_hits	The maximum number of hits to be returned.
pick_device_type	The pick device type (**PEXPickDeviceDCHitBox** or **PEXPickDeviceNPCHitVolume**).
pick_record	A pointer to the pick data record.
status_return	Returns the status of the pick operation.
more_return	Returns the status of remaining picks.
count_return	Returns the number of pick paths.

RETURNS

An array of pick paths; a null pointer if unsuccessful or no pick (see also *status_return).*

DESCRIPTION

This function traverses the structure network specified by the renderer's current pick start path. Hit testing begins after the last element specified in the renderer's current pick start path and continues until one of two conditions is met: the maximum number of hits is reached, or the last element of the first structure in the pick start path is processed. If the pick start path does not define a valid hierarchical path, a **BadPEXPath** error is sent and a null pick path is returned.

Standard pick all methods are **PEXPickAllAll** and **PEXPickAllVisible.** The supported pick device types are inquirable via **PEXGetEnumTypeInfo**(3).

If one or more primitives were picked, a pick status of **PEXPick** is returned along with the pick paths. The hierarchical pick path is equivalent to the renderer's current path at the time the picked primitive was processed. If no primitives were picked, the returned pick status is **PEXNoPick,** and the returned pick path is a null pointer. If the renderer's drawable was destroyed or resized during the pick operation, the returned pick status is **PEXAbortedPick** and the returned pick path is a null pointer.

The paths of all hit primitives are recorded until reaching the maximum number of hits or until the server maximum number of recordable hits is reached. Once the maximum number of paths is recorded, subsequent primitives may be ignored and the results returned.

If all possible hits were recorded, then **PEXNoMoreHits** is returned and the renderer's pick start path will be empty. If the maximum number of hits was reached and additional hits were detected, then **PEXMoreHits** is returned and the renderer's pick start path will be set to the last recorded hit primitive. If, after reaching the maximum number of hits, subsequent output commands were ignored, then **PEXMayBeMoreHits** is returned and the renderer's pick start path will be set to the last element processed by the renderer before it started ignoring primitives.

If the specified drawable does not have the same root and depth as the drawable used to create the renderer, or, if the specified drawable is not one of the supported drawables returned by **PEXMatchRenderingTargets ,** a match error is generated. If the renderer state is set to **PEXRendering** or **PEXPicking** when this function is called, then the operation in progress is aborted, the **PEXPickAll** function is completed, and a **BadPEXRendererState** error returned.

PEXlib allocates memory for the return value. **PEXFreePickPaths**(3) should be called to deallocate the memory.

DATA STRUCTURES

```
typedef  XID     PEXRenderer;

typedef union {
        PEXPDNPCHitVolume  volume;
        PEXPDDCHitBox      box;
        PEXPickDataRecord  data;
} PEXPickRecord;

typedef  PEXNPCSubVolume    PEXPDNPCHitVolume;

typedef struct {
        PEXCoord       min;
        PEXCoord       max;
} PEXNPCSubVolume;

typedef struct {
        float    x;
        float    y;
        float    z;
} PEXCoord;

typedef struct {
        PEXDeviceCoord2D    position;
        float               distance;
} PEXPDDCHitBox;

typedef struct {
        short    x;
        short    y;
```

```
} PEXDeviceCoord2D;

typedef struct {
        unsigned short          length;  /* number of bytes in record */
        char                    *record;
} PEXPickDataRecord;

typedef struct {
        unsigned long           count;   /* number of elements */
        PEXPickElementRef       *elements;
} PEXPickPath;

typedef struct {
        PEXStructure    sid;
        unsigned long   offset;
        unsigned long   pick_id;
} PEXPickElementRef;

typedef  XID     PEXStructure;
```

ERRORS

BadAlloc
The server failed to allocate resources necessary to complete request.

BadDrawable
The specified drawable resource identifier is invalid.

BadMatch
The specified drawable is unsupported, or the specified renderer resource was not created with a compatible drawable.

BadPEXRenderer
The specified renderer resource identifier is invalid.

BadPEXRendererState
The specified renderer was in an invalid state.

BadPEXPath
The renderer pick start path is invalid.

BadValue
The pick record contains invalid data, or the pick device type is invalid.

SEE ALSO

PEXBeginPickAll(3)
PEXEndPickAll(3)
PEXFreePickPaths(3)