**NAME**

PEXFillArea - 3D Fill Area Primitive

**SYNTAX**

void PEXFillArea(Display **display*, XID *resource_id*, PEXOCRequestType *req_type*, int *shape_hint*, int *ignore_edges*, unsigned int *count*, PEXCoord **points*)

**PARAMETERS**

| | |
|---|---|
| *display* | A pointer to a display structure returned by a successful **XOpenDisplay** call. |
| *resource_id* | The resource identifier of the renderer or structure. |
| *req_type* | The request type for the output command ( **PEXOCRender, PEXOCStore, PEXOCRenderSingle** or **PEXOCStoreSingle** ). |
| *shape_hint* | The shape of the fill area ( **PEXShapeComplex, PEXShapeNonConvex, PEXShapeConvex, PEXShapeUnknown** ). |
| *ignore_edges* | A flag that determines if surface edges are rendered ( **True** or **False** ). |
| *count* | The number of points. |
| *points* | An array of points defining the fill area. |

**RETURNS**

None

**DESCRIPTION**

This function creates a fill area output primitive.

The area is defined by the list of *points* joined together to form a planar surface. Fill areas are not strictly required to be planar, but shading artifacts may occur if a fill area is not planar or nearly so. The first vertex of the fill area is connected to the second, the second to the third, and so on. The last vertex is implicitly connected to the first.

During the rendering process, the fill area vertices are transformed to positions in device coordinates. The surface colors are affected by the reflectance calculations which uses the light state, interior style, and reflection model attributes. Surface colors are further affected by the depth-cueing computation and then mapped to device colors. Fill areas outside the currently-defined clipping volume are not displayed. Fill areas crossing the clipping volume are clipped, and only the portions inside the clipping volume are displayed.

A fill area with fewer than three vertices is considered degenerate. It is stored in a structure, but when rendered, the primitive is ignored and has no visual effect.

A fill area may cross over itself to create a complex shape. The odd-even rule is used for determining the area that lies in the interior of the fill area. The shape hint is provided to enable performance improvements for certain shapes. Fill areas that are of higher complexity than indicated by the shape hint are rendered in an implementation-dependent manner. Consequently, applications should pass **PEXShapeUnknown** as the shape unless they are certain the fill area's shape is one of the other three.

The ignore edges flag is a boolean value specifying whether surface edges are rendered. If the flag is **True,** no surface edges are rendered for the fill area. If the ignore edges flag is **False,** surface edges are rendered according to the surface edge attributes if the surface edge flag attribute is **PEXOn.** Depending on the setting of the surface edge ASF values, the surface edges, surface edge color, surface edge type, and surface edge width attributes are obtained from one of two sources. These attributes are obtained directly from the current surface edge attribute values or from the edge bundle lookup table entry specified by the current edge bundle index attribute depending on the setting of the surface edge ASF attribute.

Depending on the setting of the surface attribute ASF values, the surface color, interior style, interior style index, surface interpolation method, and reflection model attributes are obtained from one of two sources. These attributes are obtained directly from the current surface attributes values or from the interior bundle lookup table entry specified by the current interior bundle index attribute.

When a surface is rendered, the surface color and reflection attributes are used to compute the colors of the surface if it is front-facing with respect to the point of view and the current culling mode allows front-faces to be rendered. If the surface is back-facing, the current distinguish mode is **True,** and the current culling mode allows back-faces to be rendered, the corresponding back-facing attributes are used instead.

Regardless of the fill area orientation, if the interior style is **PEXInteriorStylePattern,** the pattern size, pattern reference point and pattern reference vectors are used to pattern the fill area.

## DATA STRUCTURES

See **PEXlib.h.**

## ERRORS

**BadPEXOutputCommand**
The output command contains an invalid value.

**BadPEXRenderer**
The specified renderer resource identifier is invalid.

**BadPEXStructure**
The specified structure resource identifier is invalid.

## SEE ALSO

**PEXSetInteriorStyle**(3)
**PEXSetInteriorStyleIndex**(3)
**PEXSetSurfaceColorIndex**(3)
**PEXSetSurfaceColor**(3)
**PEXSetReflectionAttributes**(3)
**PEXSetReflectionModel**(3)
**PEXSetSurfaceInterpMethod**(3)
**PEXSetBFInteriorStyle**(3)
**PEXSetBFInteriorStyleIndex**(3)
**PEXSetBFSurfaceColorIndex**(3)
**PEXSetBFSurfaceColor**(3)
**PEXSetBFReflectionAttributes**(3)
**PEXSetBFReflectionModel**(3)
**PEXSetBFSurfaceInterpMethod**(3)
**PEXSetFacetCullingMode**(3)

**PEXSetFacetDistinguishFlag**(3)
**PEXSetPatternSize**(3)
**PEXSetPatternAttributes**(3)
**PEXSetPatternAttributes2D**(3)
**PEXSetInteriorBundleIndex**(3)
**PEXSetSurfaceEdgeFlag**(3)
**PEXSetSurfaceEdgeType**(3)
**PEXSetSurfaceEdgeWidth**(3)
**PEXSetSurfaceEdgeColor**(3)
**PEXSetSurfaceEdgeColorIndex**(3)
**PEXSetEdgeBundleIndex**(3)