NAME

PEXQuadrilateralMesh - 3D Quadrilateral Mesh Primitive

SYNTAX

void PEXQuadrilateralMesh(Display **display*, XID *resource_id*, PEXOCRequestType *req_type*, int *shape_hint*, unsigned int *facet_attributes*, unsigned int *vertex_attributes*, int *color_type*, PEXArrayOfFacetData *facet_data*, unsigned int *col_count*, unsigned int *row_count*, PEXArrayOfVertex *vertices*)

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 which describes all of the quadrilaterals (**PEXShapeComplex, PEXShapeNonConvex, PEXShapeConvex, PEXShapeUnknown**).
facet_attributes	A mask indicating the facet attributes provided (**PEXGANone, PEXGAColor, PEXGANormal**).
vertex_attributes	A mask indicating the vertex attributes provided (**PEXGANone, PEXGAColor, PEXGANormal**).
color_type	The type of color data provided (**PEXColorTypeIndexed, PEXColorTypeRGB, PEXColorTypeCIE, PEXColorTypeHSV, PEXColorTypeHLS, PEXColorTypeRGB8, PEXColorTypeRGB16**).
facet_data	An array of facet data.
col_count	The number of columns in the vertex array.
row_count	The number of rows in the vertex array.
vertices	A two-dimensional (row-major) array of vertices defining the quadrilateral mesh.

RETURNS

None

DESCRIPTION

This function creates a quadrilateral mesh output primitive.

The quadrilateral mesh surface is created from the vertices. The vertex array is accessed in row major order (i.e. the column number varies fastest as vertices are accessed). The (ith, jth), (i+1th, jth), (i+1th, j+1th), and (ith, j+1th) vertices are connected to create a single facet. Adjacent vertices are interconnected until the entire facet network is processed.

Normals for quadrilaterals, if not provided explicitly, are computed by taking the cross product of the diagonals. For a quadrilateral with the above vertices, the cross product would be formed as follows:

normal(i,j) = (point(i+1, j+1) - point(i, j)) X (point(i, j+1) - point(i+1, j))

Normals are assumed to be unit length vectors. If not unit length, the result is implementation-dependent.

There must be an entry in the facet data array for each facet, if facet data is indicated by the facet attributes.

All other aspects of this primitive are the same as **PEXFillAreaSetWithData**(3)**.**

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)