NAME

PEXEncodedText – Encoded 3D Text Primitive

SYNTAX

void PEXEncodedText(Display **display*, XID *resource_id*, PEXOCRequestType *req_type*, PEXCoord **origin*, PEXVector **vector1*, PEXVector **vector2*, unsigned int *count*, PEXEncodedTextData **encoded_text*)

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**).
origin	The origin of the text string.
vector1	A vector defining the positive x-direction of the text local coordinate system.
vector2	A vector defining the positive y-direction of the text local coordinate system.
count	The number of encoded text strings.
encoded_text	An array of encoded text strings.

RETURNS

None

DESCRIPTION

This function creates a text output primitive.

This function is similar to **PEXText**(3)**,** except that multiple encoded text strings are specified. Each text string in encoded text array has a character set, a character set width, an encoding state, and a list of characters.

The character set is an index into the current font group. Font groups contain individual fonts which are numbered starting at one; a value of three selects the third font in the font group currently being used. If a character set is not available in the current font group then the entire string will be rendered using the default font group. If a character set value is not available in the default font group then that portion of the string will be rendered in an implementation-dependent manner. The character set width indicates the width or size of characters in the strings. Valid values for character set width are **PEXCSByte, PEXCSShort** and **PEXCSLong.** The encoding state is provided for use by the application and is not interpreted by the PEX server.

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

DATA STRUCTURES

```
typedef struct {
        unsigned short      character_set;
        unsigned char       character_set_width;    /* PEXCSByte, PEXCSShort,
                                                       PEXCSLong */
        unsigned char       encoding_state;
        unsigned short      reserved;
```

```
        unsigned short      length;
        char                *ch;
} PEXEncodedTextData;
```

See also **PEXlib.h.**

ERRORS

BadPEXRenderer
The specified renderer resource identifier is invalid.

BadPEXStructure
The specified structure resource identifier is invalid.

SEE ALSO

PEXSetTextFontIndex(3)
PEXSetTextPrecision(3)
PEXSetCharExpansion(3)
PEXSetCharSpacing(3)
PEXSetTextColorIndex(3)
PEXSetTextColor(3)
PEXSetCharHeight(3)
PEXSetCharUpVector(3)
PEXSetTextPath(3)
PEXSetTextAlignment(3)
PEXSetTextBundleIndex(3)