## NAME

glutInitDisplayString - sets the initial display mode via a string.

## SYNTAX

void glutInitDisplayString(char *string);

## ARGUMENTS

*string* Display mode description string, see below.

## DESCRIPTION

The initial display mode description string is used when creating top-level windows, subwindows, and overlays to determine the OpenGL display mode for the to-be-created window or overlay.

The string is a list of zero or more capability descriptions separated by spaces and tabs. Each capability description is a capability name that is optionally followed by a comparator and a numeric value. For example, "double" and "depth>=12" are both valid criteria.

The capability descriptions are translated into a set of criteria used to select the appropriate frame buffer configuration.

The criteria are matched in strict left to right order of precdence. That is, the first specified criteria (leftmost) takes precedence over the later criteria for non-exact criteria (greater than, less than, etc. comparators). Exact criteria (equal, not equal compartors) must match exactly so precedence is not relevant.

The numeric value is an integer that is parsed according to ANSI C's strtol(str, strptr, 0) behavior. This means that decimal, octal (leading 0), and hexidecimal values (leading 0x) are accepeted.

The valid compartors are:

**=** Equal.

**!=** Not equal.

**<** Less than and preferring larger difference (the least is best).

**>** Greeater than and preferring larger differences (the most is best).

**<=** Less than or equal and preferring larger difference (the least is best).

**>=** Greater than or equal and preferring more instead of less. This comparator is useful for allocating resources like color precsion or depth buffer precision where the maximum precison is generally preferred. Contrast with the tilde (˜) comprator.

**˜** Greater than or equal but preferring less instead of more. This compartor is useful for allocating resources such as stencil bits or auxillary color buffers where you would rather not over allocate.

When the compartor and numeric value are not specified, each capability name has a different default (one default is to require a a compartor and numeric value).

The valid capability names are:

**alpha** Alpha color buffer precision in bits. Default is ">=1".

**acca** Red, green, blue, and alpha accumulation buffer precision in bits. Default is ">=1" for red, green, blue, and alpha capabilities.

**acc** Red, green, and green accumulation buffer precision in bits and zero bits of alpha accumulation buffer precision. Default is ">=1" for red, green, and blue capabilities, and "˜0" for the alpha capability.

**blue** Blue color buffer precision in bits. Default is ">=1".

**buffer** Number of bits in the color index color buffer. Default is ">=1".

**conformant**
Boolean indicating if the frame buffer configuration is conformant or not. Conformance information is based on GLX's EXT_visual_rating extension if supported. If the extension is not supported, all visuals are assumed conformat. Default is "=1".

**depth** Number of bits of precsion in the depth buffer. Default is ">=12".

**double** Boolean indicating if the color buffer is double buffered. Default is "=1".

**green** Green color buffer precision in bits. Default is ">=1".

**index** Boolean if the color model is color index or not. True is color index. Default is ">=1".

**num** A special capability name indicating where the value represents the Nth frame buffer configuration matching the description string. When not specified, glutInitDisplayString also returns the first (best matching) configuration. num requires a compartor and numeric value.

**red** Red color buffer precision in bits. Default is ">=1".

**rgba** Number of bits of red, green, blue, and alpha in the RGBA color buffer. Default is ">=1" for red, green, blue, and alpha capabilities, and "=1" for the RGBA color model capability.

**rgb** Number of bits of red, green, and blue in the RGBA color buffer and and zero bits of alpha color buffer precision. Default is ">=1" for the red, green, and blue capabilities, and "~0" for alpha capability, and "=1" for the RGBA color model capability.

**stencil** Number of bits in the stencil buffer.

**single** Boolean indicate the color buffer is single buffered. Double buffer capability "=1".

**stereo** Boolean indicating the color buffer is supports OpenGL-style stereo. Default is "=1".

**samples** Indicates the number of multisamples to use based on GLX's SGIS_multisample extension (for antialiasing). Default is "<=4". This default means that a GLUT application can request multipsampling if available by simply specifying "samples".

**slow** Boolean indicating if the frame buffer configuration is slow or not. Slowness information is based on GLX's EXT_visual_rating extension if supported. If the extension is not supported, all visuals are assumed fast. Note that slowness is a relative designation relative to other frame buffer configurations available. The intent of the slow capability is to help programs avoid frame buffer configurations that are slower (but perhaps higher precision) for the current machine. Default is ">=0". This default means that slow visuals are used in preference to fast visuals, but fast visuals will still be allowed.

**xvisual** Only recongized on GLUT implementations for the X Window System, this capability name matches the X visual ID by number. xvisual requires a compartor and numeric value.

**xstaticgray**
Only recongized on GLUT implementations for the X Window System, boolean indicating if the frame buffer configuration's X visual is of type StaticGray. Default is "=1".

**xgrayscale**
Only recongized on GLUT implementations for the X Window System, boolean indicating if the frame buffer configuration's X visual is of type GrayScale. Default is "=1".

**xstaticcolor**
Only recongized on GLUT implementations for the X Window System, boolean indicating if the frame buffer configuration's X visual is of type StaticColor. Default is "=1".

**xpseudocolor**
Only recongized on GLUT implementations for the X Window System, boolean indicating if the frame buffer configuration's X visual is of type PsuedoColor. Default is "=1".

**xtruecolor**
Only recongized on GLUT implementations for the X Window System, boolean indicating if the frame buffer configuration's X visual is of type TrueColor. Default is "=1".

**xdirectcolor**
Only recongized on GLUT implementations for the X Window System, boolean indicating if the frame buffer configuration's X visual is of type DirectColor. Default is "=1".

Unspecifed capability descriptions will result in unspecified criteria being generated. These unspecified criteria help glutInitDisplayString behave sensibly with terse display mode description strings.

## EXAMPLE

Here is an examples using glutInitDisplayString:

```
glutInitDisplayString("stencil~2 rgb double depth>=16 samples");
```

The above call requests a window with an RGBA color model (but requesting no bits of alpha), a depth buffer with at least 16 bits of precsion but preferring more, mutlisampling if available, and at least 2 bits of stencil (favoring less stencil to more as long as 2 bits are available).

## SEE ALSO

glutInit, glutCreateWindow, glutInitDisplayMode

## AUTHOR

Mark J. Kilgard (mjk@sgi.com)