NAME

XParseGeometry, XWMGeometry - parse window geometry

SYNTAX

int XParseGeometry(_Xconst char *_parsestring_, int *_x_return_, int *_y_return_, unsigned int *_width_return_, unsigned int *_height_return_);

int XWMGeometry(Display *_display_, int _screen_, _Xconst char *_user_geom_, _Xconst char *_def_geom_, unsigned int _bwidth_, XSizeHints *_hints_, int *_x_return_, int *_y_return_, int *_width_return_, int *_height_return_, int *_gravity_return_);

ARGUMENTS

position

default_position

Specify the geometry specifications.

display Specifies the connection to the X server.

fheight

fwidth Specify the font height and width in pixels (increment size).

parsestring Specifies the string you want to parse.

screen Specifies the screen.

width_return

height_return Return the width and height determined.

xadder

yadder Specify additional interior padding needed in the window.

x_return

y_return Return the x and y offsets.

bwidth Specifies the border width.

hints Specifies the size hints for the window in its normal state.

def_geom Specifies the application's default geometry or NULL.

gravity_return Returns the window gravity.

user_geom Specifies the user-specified geometry or NULL.

DESCRIPTION

By convention, X applications use a standard string to indicate window size and placement. **XParseGeometry** makes it easier to conform to this standard because it allows you to parse the standard window geometry. Specifically, this function lets you parse strings of the form:

[=][<*width*>{xX}<*height*>][{+-}<*xoffset*>{+-}<*yoffset*>]

The fields map into the arguments associated with this function. (Items enclosed in <> are integers, items in [] are optional, and items enclosed in {} indicate "choose one of." Note that the brackets should not appear in the actual string.) If the string is not in the Host Portable Character Encoding, the result is implementation-dependent.

The **XParseGeometry** function returns a bitmask that indicates which of the four values (width, height, xoffset, and yoffset) were actually found in the string and whether the x and y values are negative. By convention, -0 is not equal to +0, because the user needs to be able to say "position the window relative to the right or bottom edge." For each value found, the corresponding argument is updated. For each value not found, the argument is left unchanged. The bits are represented by **XValue**, **YValue**, **WidthValue**, **HeightValue**, **XNegative**, or **YNegative** and are defined in **X11/Xutil.h**. They will be set whenever one of the values is defined or one of the signs is set.

If the function returns either the **XValue** or **YValue** flag, you should place the window at the requested position.

The **XWMGeometry** function combines any geometry information (given in the format used by **XParseGeometry**) specified by the user and by the calling program with size hints (usually the ones to be stored in WM_NORMAL_HINTS) and returns the position, size, and gravity (**NorthWestGravity**, **NorthEastGravity**, **SouthEastGravity**, or **SouthWestGravity**) that describe the window. If the base size is not set in the **XSizeHints** structure, the minimum size is used if set. Otherwise, a base size of zero is assumed. If no minimum size is set in the hints structure, the base size is used. A mask (in the form returned by **XParseGeometry**) that describes which values came from the user specification and whether or not the position coordinates are relative to the right and bottom edges is returned. Note that these coordinates will have already been accounted for in the x_return and y_return values.

Note that invalid geometry specifications can cause a width or height of zero to be returned. The caller may pass the address of the hints win_gravity field as gravity_return to update the hints directly.

SEE ALSO

XSetWMProperties(3)
Xlib - C Language X Interface