XChangePointerControl, XGetPointerControl – control pointer

**XChangePointerControl**

```c
Display *display;
Bool do_accel, do_threshold;
int accel_numerator, accel_denominator;
int threshold;
```

**XGetPointerControl**

```c
Display *display;
int *accel_numerator_return, *accel_denominator_return;
int *threshold_return;
```

**accel_denominator** Specifies the denominator for the acceleration multiplier.

**accel_denominator_return** Returns the denominator for the acceleration multiplier.

**accel_numerator** Specifies the numerator for the acceleration multiplier.

**accel_numerator_return** Returns the numerator for the acceleration multiplier.

**display** Specifies the connection to the X server.

**do_accel** Specifies a Boolean value that controls whether the values for the accel_numerator or accel_denominator are used.

**do_threshold** Specifies a Boolean value that controls whether the value for the threshold is used.

**threshold** Specifies the acceleration threshold.

**threshold_return** Returns the acceleration threshold.

The **XChangePointerControl** function defines how the pointing device moves. The acceleration, expressed as a fraction, is a multiplier for movement. For example, specifying 3/1 means the pointer moves three times as fast as normal. The fraction may be rounded arbitrarily by the X server. Acceleration only takes effect if the pointer moves more than threshold pixels at once and only applies to the amount beyond the value in the threshold argument. Setting a value to −1 restores the default. The values of the do_accel and do_threshold arguments must be **True** for the pointer values to be set, or the parameters are unchanged. Negative values (other than −1) generate a **BadValue** error, as does a zero value for the accel_denominator argument.

**XChangePointerControl** can generate a **BadValue** error.

The **XGetPointerControl** function returns the pointer’s current acceleration multiplier and acceleration threshold.

**BadValue** Some numeric value falls outside the range of values accepted by the request. Unless a specific range is specified for an argument, the full range defined by the argument’s type is accepted. Any argument defined as a set of alternatives can generate this error.

*Xlib – C Language X Interface*