**XGetDeviceControl, XChangeDeviceControl − query and change input device controls**

**XDeviceControl \* XGetDeviceControl(***display***,** *device***,** *control***)**
   **Display \****display***;**
   **XDevice \****device***;**
   **int \****controlType***;**

**int XChangeDeviceControl(***display***,** *device***,** *controlType***,** *control***)**
   **Display \****display***;**
   **XDevice \****device***;**
   **int** *controlType***;**
   **XDeviceControl \****control***;**

*display* **Specifies the connection to the X server.** *device* **Specifies the device whose control is to be interrogated or modified.** *controlType* **Specifies the type of control to be interrogated or changed.** *control* **Specifies the address of an** *XDeviceControl* **structure that contains the new values for the Device.**

**These requests are provided to manipulate those input devices that support device control. A** *BadMatch* **error will be generated if the requested device does not support any device controls.**

Valid device control types that can be used with these requests include the following: DEVICE_RESOLUTION Queries or changes the resolution of valuators on input devices.

The *XGetDeviceControl* request returns a pointer to an *XDeviceControl* structure.

*XGetDeviceControl* can generate a *BadDevice* or *BadMatch* error.

The *XChangeDeviceControl* request modifies the values of one control on the specified device. The control is identified by the id field of the *XDeviceControl* structure that is passed with the request.

*XChangeDeviceControl* can generate a *BadDevice*, *BadMatch*, or *BadValue* error.

**Each control is described by a structure specific to that control. These structures are defined in the file** *XInput.h***.**

*XDeviceControl* is a generic structure that contains two fields that are at the beginning of each class of control:

```
typedef struct {
        XID class;
        int length;
} XDeviceControl;
```

The *XDeviceResolutionState* structure defines the information that is returned for device resolution for devices with valuators.

```
typedef struct {
        XID     control;
        int     length;
        int     num_valuators;
        int     *resolutions;
        int     *min_resolutions;
        int     *max_resolutions;
} XDeviceResolutionState;
```

The *XDeviceResolutionControl* structure defines the attributes that can be controlled for keyboard Devices.

```
typedef struct {
        XID    control;
        int    length;
        int    first_valuator;
        int    num_valuators;
        int    *resolutions;
} XDeviceResolutionControl;
```

*BadDevice* **An invalid device was specified.  The specified device does not exist or has not been opened by this client via** *XOpenInputDevice***.  This error may also occur if some other client has caused the specified device to become the X keyboard or X pointer device via the** *XChangeKeyboardDevice* **or** *XChangePointerDevice* **requests.** *BadMatch* **This error may occur if an** *XGetDeviceControl* **request was made specifying a device that has no controls or an** *XChangeDeviceControl* **request was made with an** *XDeviceControl* **structure that contains an invalid Device type.  It may also occur if an invalid combination of mask bits is specified (***DvKey* **but no** *DvAutoRepeatMode* **for keyboard Devices), or if an invalid KeySym is specified for a string Device.** *BadValue* **Some numeric value falls outside the range of values accepted by the** *XChangeDeviceControl* **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.**

*Programming With Xlib*