

## XGetFeedbackControl, XChangeFeedbackControl – query and change input device feedbacks

```
XFeedbackState * XGetFeedbackControl(display, device, num_feedbacks)
    Display *display;
    XDevice *device;
    int *num_feedbacks;
```

```
int XChangeFeedbackControl(display, device, mask, control)
    Display *display;
    XDevice *device;
    Mask mask;
    XFeedbackControl *control;
```

*display* Specifies the connection to the X server. *device* Specifies the device whose feedbacks are to be queried or modified. *num\_feedbacks* Specifies an address into which the number of feedbacks supported by the device is to be returned. *mask* Specifies a mask specific to each type of feedback that describes how the feedback is to be modified. *control* Specifies the address of an *XFeedbackControl* structure that contains the new values for the feedback.

These requests are provided to manipulate those input devices that support feedbacks. A *BadMatch* error will be generated if the requested device does not support feedbacks. Whether or not a given device supports feedbacks can be determined by examining the information returned by the *XOpenDevice* request. For those devices that support feedbacks, *XOpenDevice* will return an *XInputClassInfo* structure with the *input\_class* field equal to the constant *FeedbackClass* (defined in the file *XI.h*).

The *XGetFeedbackControl* request returns a pointer to a list of *XFeedbackState* structures. Each item in this list describes one of the feedbacks supported by the device. The items are variable length, so each contains its length to allow traversal to the next item in the list.

The feedback classes that are currently defined are: *KbdFeedbackClass*, *PtrFeedbackClass*, *StringFeedbackClass*, *IntegerFeedbackClass*, *LedFeedbackClass*, and *BellFeedbackClass*. These constants are defined in the file *XI.h*. An input device may support zero or more classes of feedback, and may support multiple feedbacks of the same class. Each feedback contains a class identifier and an id that is unique within that class for that input device. The id is used to identify the feedback when making an *XChangeFeedbackControl* request.

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

The *XChangeFeedbackControl* request modifies the values of one feedback on the specified device. The feedback is identified by the *id* field of the *XFeedbackControl* structure that is passed with the request. The fields of the feedback that are to be modified are identified by the bits of the mask that is passed with the request.

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

Each class of feedback is described by a structure specific to that class. These structures are defined in the file *XInput.h*. *XFeedbackState* and *XFeedbackControl* are generic structures that contain three fields that are at the beginning of each class of feedback:

```
typedef struct {
    XID class;
    int length;
    XID id;
} XFeedbackState, XFeedbackControl;
```

The *XKbdFeedbackState* structure defines the attributes that are returned for feedbacks equivalent to those on the X keyboard.

```
typedef struct {
    XID class;
    int length;
    XID id;
    int click;
    int percent;
    int pitch;
    int duration;
    int led_mask;
    int global_auto_repeat;
    char auto_repeats[32];
} XKbdFeedbackState;
```

The *XPtrFeedbackState* structure defines the attributes that are returned for feedbacks equivalent to those on the the X pointer.

```
typedef struct {
    XID class;
    int length;
    XID id;
    int accelNum;
    int accelDenom;
    int threshold;
} XPtrFeedbackState;
```

The *XIntegerFeedbackState* structure defines attributes that are returned for integer feedbacks.

```
typedef struct {
    XID class;
    int length;
    XID id;
    int resolution;
    int minVal;
    int maxVal;
} XIntegerFeedbackState;
```

The *XStringFeedbackState* structure defines the attributes that are returned for string feedbacks.

```
typedef struct {
    XID class;
    int length;
    XID id;
    int max_symbols;
    int num_syms_supported;
    KeySym *syms_supported;
} XStringFeedbackState;
```

The *XBellFeedbackState* structure defines the attributes that are returned for bell feedbacks.

```
typedef struct {
    XID class;
    int length;
    XID id;
    int percent;
    int pitch;
    int duration;
} XBellFeedbackState;
```

The *XLedFeedbackState* structure defines the attributes that are returned for LED feedbacks.

```
typedef struct {
    XID class;
    int length;
    XID id;
    int led_values;
} XLedFeedbackState;
```

The *XPtrFeedbackControl* structure defines the attributes that can be controlled for pointer feedbacks.

```
typedef struct {
    XID class;
    int length;
    XID id;
    int accelNum;
    int accelDenom;
    int threshold;
} XPtrFeedbackControl;
```

The *XKbdFeedbackControl* structure defines the attributes that can be controlled for keyboard feedbacks.

```
typedef struct {
    XID class;
    int length;
    XID id;
    int click;
    int percent;
    int pitch;
    int duration;
    int led_mask;
    int led_value;
    int key;
    int auto_repeat_mode;
} XKbdFeedbackControl;
```

The *XStringFeedbackControl* structure defines the attributes that can be controlled for string feedbacks.

```
typedef struct {
    XID class;
    int length;
    XID id;
    int num_keysyms;
    KeySym *syms_to_display;
} XStringFeedbackControl;
```

The *XIntegerFeedbackControl* structure defines the attributes that can be controlled for integer feedbacks.

```
typedef struct {
    XID class;
    int length;
    XID id;
    int int_to_display;
} XIntegerFeedbackControl;
```

The *XBellFeedbackControl* structure defines the attributes that can be controlled for bell feedbacks.

```
typedef struct {
    XID class;
    int length;
    XID id;
    int percent;
    int pitch;
    int duration;
} XBellFeedbackControl;
```

The *XLedFeedbackControl* structure defines the attributes that can be controlled for LED feedbacks.

```
typedef struct {
    XID class;
    int length;
    XID id;
    int led_mask;
    int led_values;
} XLedFeedbackControl;
```

***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 *XGetFeedbackControl* request was made specifying a device that has no feedbacks, or an *XChangeFeedbackControl* request was made with an *XFeedbackControl* structure that contains an invalid feedback type. It may also occur if an invalid combination of mask bits is specified (*DvKey* but no *DvAutoRepeatMode* for keyboard feedbacks), or if an invalid *KeySym* is specified for a string feedback. ***BadValue*** Some numeric value falls outside the range of values accepted by the *XChangeFeedbackControl* 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*