

XListInputDevices, XFreeDeviceList – list available input devices

XDeviceInfo *XListInputDevices(*display*, *ndevices_return*)

Display **display*;
int **ndevices_return*;

XFreeDeviceList(*list*)

XDeviceInfo **list*;

display Specifies the connection to the X server. *ndevices_return* Specifies a pointer to a variable where the number of available devices can be returned. *list* Specifies the list of devices to free. The *XFreeDeviceList* function frees the list of available extension input devices.

The *XListInputDevices* request lists the available extension input devices. This list includes the X pointer and X keyboard, any other input devices that are currently accessible through the X server, and any input devices that are not currently accessible through the X server but could be accessed if requested.

Some server implementations may make all input devices available at the time the server is initialized. Others may wait until requested by a client to access an input device. In the latter case, it is possible that an input device will be listed as available at one time but not at another.

For each input device available to the server, the *XListInputDevices* request returns an *XDeviceInfo* structure. That structure contains a pointer to a list of structures, each of which contains information about one class of input supported by the device.

The *XDeviceInfo* structure is defined as follows:

```
typedef struct _XDeviceInfo
{
    XID      id;
    Atom     type;
    char     *name;
    int      num_classes;
    int      use;
    XAnyClassPtr  inputclassinfo;
} XDeviceInfo;
```

The *id* is a number in the range 0-128 that uniquely identifies the device. It is assigned to the device when it is initialized by the server.

The *type* field is of type *Atom* and indicates the nature of the device.

The *name* field contains a pointer to a null-terminated string that corresponds to one of the defined device types. The name will correspond to one of the following strings (defined in the header file *XI.h*):

XI_MOUSE
XI_TABLET
XI_KEYBOARD
XI_TOUCHSCREEN
XI_TOUCHPAD
XI_BUTTONBOX
XI_BARCODE
XI_TRACKBALL
XI_QUADRATURE
XI_ID_MODULE
XI_ONE_KNOB
XI_NINE_KNOB
XI_KNOB_BOX
XI_SPACEBALL
XI_DATAGLOVE
XI_EYETRACKER
XI_CURSORKEYS
XI_FOOTMOUSE

These names may be directly compared with the name field of the XDeviceInfo structure, or used in an XInternAtom request to return an atom that can be compared with the type field of the XDeviceInfo structure.

The num_classes field is a number in the range 0-255 that specifies the number of input classes supported by the device for which information is returned by ListInputDevices. Some input classes, such as class Focus and class Proximity do not have any information to be returned by ListInputDevices.

The use field specifies how the device is currently being used. If the value is *IsXKeyboard*, the device is currently being used as the X keyboard. If the value is *IsXPointer*, the device is currently being used as the X pointer. If the value is *IsXExtensionDevice*, the device is available for use as an extension device.

The inputclassinfo field contains a pointer to the first input-class specific data. The first two fields are common to all classes.

The class field is a number in the range 0-255. It uniquely identifies the class of input for which information is returned. Currently defined classes are KeyClass, ButtonClass, and ValuatorClass.

The length field is a number in the range 0- 255. It specifies the number of bytes of data that are contained in this input class. The length includes the class and length fields.

The XKeyInfo structure describes the characteristics of the keys on the device. It is defined as follows:

```
typedef struct _XKeyInfo {
    XID          class;
    int          length;
    unsigned short min_keycode;
    unsigned short max_keycode;
    unsigned short num_keys;
} XKeyInfo;
```

min_keycode is of type KEYCODE. It specifies the minimum keycode that the device will report. The minimum keycode will not be smaller than 8.

max_keycode is of type KEYCODE. It specifies the maximum keycode that the device will report. The maximum keycode will not be larger than 255.

num_keys specifies the number of keys that the device has.

The XButtonInfo structure defines the characteristics of the buttons on the device. It is defined as follows:

```
typedef struct _XButtonInfo {
    XID      class;
    int      length;
    short    num_buttons;
} XButtonInfo;
```

num_buttons specifies the number of buttons that the device has.

The XValuatorInfo structure defines the characteristics of the valuator on the device. It is defined as follows:

```
typedef struct _XValuatorInfo {
    XID      class;
    int      length;
    unsigned char    num_axes;
    unsigned char    mode;
    unsigned long    motion_buffer;
    XAxisInfoPtr     axes;
} XValuatorInfo;
```