X11 Input Extension Library Specification

X Consortium Standard

X Version 11, Release 6

Mark Patrick **Ardent Computer**

George Sachs **Hewlett-Packard**

Copyright © 1989, 1990, 1991 by Hewlett-Packard Company, Ardent Computer.

Permission to use, copy, modify, and distribute this documentation for any purpose and without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. Ardent, and Hewlett-Packard make no representations about the suitability for any purpose of the information in this document. It is provided ``as is'' without express or implied warranty.

Copyright (c) 1989, 1990, 1991, 1992 X Consortium

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ''Software''), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ''AS IS'', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

X Window System is a trademark of X Consortium, Inc.

1. Input Extension Overview

This document describes an extension to the X11 server. The purpose of this extension is to support the use of additional input devices beyond the pointer and keyboard devices defined by the core X protocol. This first section gives an overview of the input extension. The following sections correspond to chapters 7 and 8, "Window Manager functions" and "Events and Event-Handling Functions" of the "Xlib - C Language Interface" manual and describe how to use the input extension.

1.1. Design Approach

The design approach of the extension is to define functions and events analogous to the core functions and events. This allows extension input devices and events to be individually distinguishable from each other and from the core input devices and events . These functions and events make use of a device identifier and support the reporting of n-dimensional motion data as well as other data that is not currently reportable via the core input events.

1.2. Core Input Devices

The X server core protocol supports two input devices: a pointer and a keyboard. The pointer device has two major functions. First, it may be used to generate motion information that client programs can detect. Second, it may also be used to indicate the current location and focus of the X keyboard. To accomplish this, the server echoes a cursor at the current position of the X pointer. Unless the X keyboard has been explicitly focused, this cursor also shows the current location and focus of the X keyboard.

The X keyboard is used to generate input that client programs can detect.

The X keyboard and X pointer are referred to in this document as the *core devices*, and the input events they generate (**KeyPress**, **KeyRelease**, **ButtonPress**, **ButtonRelease**, and **MotionNotify**) are known as the *core input events*. All other input devices are referred to as *extension input devices* and the input events they generate are referred to as *extension input events*.

Note

> This input extension does not change the behavior or functionality of the core input devices, core events, or core protocol requests, with the exception of the core grab requests. These requests may affect the synchronization of events from extension devices. See the explanation in the section titled "Event Synchronization and Core Grabs".

Selection of the physical devices to be initially used by the server as the core devices is left implementation-dependent. Functions are defined that allow client programs to change which physical devices are used as the core devices.

1.3. Extension Input Devices

The input extension controls access to input devices other than the X keyboard and X pointer. It allows client programs to select input from these devices independently from each other and independently from the core devices. Input events from these devices are of extension types (**DeviceKeyPress**, **DeviceKeyRelease**, **DeviceButtonPress**, **DeviceButtonRelease**, **DeviceMotionNotify**, etc.) and contain a device identifier so that events of the same type coming from different input devices can be distinguished.

Extension input events are not limited in size by the size of the server 32-byte wire events. Extension input events may be constructed by the server sending as many wire sized events as

necessary to return the information required for that event. The library event reformatting routines are responsible for combining these into one or more client XEvents.

Any input device that generates key, button or motion data may be used as an extension input device. Extension input devices may have 0 or more keys, 0 or more buttons, and may report 0 or more axes of motion. Motion may be reported as relative movements from a previous position or as an absolute position. All valuators reporting motion information for a given extension input device must report the same kind of motion information (absolute or relative).

This extension is designed to accommodate new types of input devices that may be added in the future. The protocol requests that refer to specific characteristics of input devices organize that information by **input device classes**. Server implementors may add new classes of input devices without changing the protocol requests.

All extension input devices are treated like the core X keyboard in determining their location and focus. The server does not track the location of these devices on an individual basis, and therefore does not echo a cursor to indicate their current location. Instead, their location is determined by the location of the core X pointer. Like the core X keyboard, some may be explicitly focused. If they are not explicitly focused, their focus is determined by the location of the core X pointer.

1.3.1. Input Device Classes

Some of the input extension requests divide input devices into classes based on their functionality. This is intended to allow new classes of input devices to be defined at a later time without changing the semantics of these functions. The following input device classes are currently defined: