NAME

inputtest - An X.Org input driver for testing

SYNOPSIS

Section "InputDevice"
Identifier "*devname***"**
Driver "inputtest"
Option "SocketPath" "*path***"**
...
EndSection

DESCRIPTION

inputtest is an Xorg input driver that passes events received over a socket on to the server as input events. It supports the full set of the xf86 driver APIs exposed by Xorg. The primary use cases of this input driver are various integration tests that need to interface with the input subsystem.

CONFIGURATION DETAILS

Please refer to xorg.conf(5) for general configuration details and for options that can be used with all input drivers. This section only covers configuration details specific to this driver.

External process can communicate with the input driver via a named socket that is created after the driver is initialized. The paths to the socket is passed via input driver options.

The following driver **Options** are supported:

Option "SocketPath" "*string***"**
Sets the path where the driver will create a named socket. Any existing file at that location will be removed.

Option "DeviceType" "*string***"**
Sets the type of the device to be emulated.

Keyboard Initializes a keyboard device.

Pointer Initializes a relative-mode pointer device. It will have four valuators - a "Rel X" valuator at axis 0 and a "Rel Y" valuator at axis 1. A horizontal scroll valuator will be set up at axis 2. A vertical scroll valuator will be set up at axis 3.

PointerAbsolute Initializes an absolute-mode pointer device. It will have four valuators - an "Abs X" valuator at axis 0 and an "Abs Y" valuator at axis 1. A horizontal scroll valuator will be set up at axis 2. A vertical scroll valuator will be set up at axis 3.

PointerAbsoluteProximity Initializes an absolute-mode pointer device with proximity support. The valuators are initialized in the same way as for **PointerAbsolute** type.

Touch Initializes a touch device. It will have 5 valuators: an "Abs MT Position X" at axis 0, an "Abs MT Position Y" valuator at axis 1, a horizontal scroll valuator on axis 2, a vertical scroll valuator on axis 3 and an "Abs MT Pressure" valuator at axis 4.

Option "TouchCount" "*int***"**
Sets the maximum number of simultaneous touches for touch devices.

Option "PointerButtonCount" "*int***"**
Sets the maximum number of buttons in pointer devices.

Option "PointerHasPressure" "*bool***"**
Selects whether "Abs Pressure" is available at the axis 4 in pointer devices.

INTERFACE WITH THE DRIVER

The communication with the driver is a binary protocol defined in include/xf86-input-inputtest-protocol.h

At the beginning, the client process that communicates with the driver must connect to the socket that is created by the driver at SocketPath. Once the connection is established, it must write a xf86ITEventClientVersion event and read a xf86ITResponseServerVersion response where the driver specifies the protocol version supported by it. If this version is lower than requested by the client, then the driver will disconnect.

After receiving xf86ITResponseServerVersion message the client may send events to the driver. Each event is an instance of one of the **xf86ITEvent*** structs. The length field defines the full length of the struct in bytes and the event field defines the type of the struct.

The responses from the server follow the same structure. Each response is an instance of one of the **xf86ITResponse*** structs. The length field defines the full length of the struct in bytes and the event field defines the type of the struct.

The synchronization with Xorg is performed via **xf86ITEventWaitForSync** event. After sending such

event, the client must read of a **xf86ITResponseSyncFinished***event***from***the***socket***without***sending***additional* events. The completion of the read operation indicates that Xorg has fully processed all input events sent to it so far.

AUTHORS

Povilas Kanapickas <povilas@radix.lt>

SEE ALSO

Xorg(1), xorg.conf(5), Xserver(1), X(7)