Multi-monitor and device support

UPDD supports both multi-monitor and multiple pointer device configurations. This means that UPDD can cater with a system that has more than one monitor or more than one pointer device and any combination thereof can be supported.

This document describes the multi-monitor and device considerations for each of the supported desktop operating systems.

The driver utilises a specific, cross-platform, graphics function to retrieve monitor layout information. For successful UPDD multi-monitor operation this layout information must correctly reflect your monitor setup. Therefore the UPDD driver can only support configurations that are identifiable by this function and in most cases we would expect this to handle most multi-monitor configurations.

To view the monitor layout as retrieved by our driver using this function use the monitors option of our command line utility.

When using UPDD in a multi-monitor environment the following issues need to be considered.

**Multi-monitor or device configurations**

There are a number of possible multi-monitor / multiple device configurations.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloned</td>
<td>Identical displays on all monitors. In this case UPDD associates all the touch devices with the same monitor.</td>
</tr>
<tr>
<td>Extended</td>
<td>The desktop supports individual monitor(s) and lists each monitor in the desktop properties and individual touch screens are to be associated with the full monitor video area. In this case each UPDD supported device is associated with an individual monitor; monitor 1, monitor 2 etc or a specified device name.</td>
</tr>
<tr>
<td>Segmented</td>
<td>The desktop supports individual monitor(s) but lists virtual monitor(s) in the desktop properties (e.g. as implemented by specialist hardware and custom video driver) which may relate to two or more physical monitors and individual touch screens are to be associated with the full monitor video area. In this case each UPDD supported device is associated with a video segment of the virtual monitor that relates to the physical monitor.</td>
</tr>
<tr>
<td>Stretched</td>
<td>The desktop supports individual monitor(s) but lists virtual monitor(s) in the desktop properties (as implemented by a stretched driver) which may relate to 2 or more physical monitors and individual touch screens are to be associated with the full monitor video area. In this case each UPDD supported device is associated with a video segment of the virtual monitor that relates to the physical monitor.</td>
</tr>
</tbody>
</table>
Multi-monitor and device support

Custom

Span

Limitations

UPDD supports up to 32 monitors but has no restriction on the number of devices it can handle. As long as the device can be connected via one of the supported hardware ports UPDD can be configured to support it.

Desktop /Device associations

Based on the monitor layout this table indicates the how the touch devices are associated with the desktop area:

Display name binding

If monitor binding is employed and two monitors of the same name are connected then binding is likely to be incorrect. However, in this situation using the Configure utility should correctly assign the touch / desktop association.

A touch screen monitor is two totally independent devices, a video display and a touchscreen, with no known relationship between the devices.

Therefore when a touch screen monitor is connected to a system, with the independent video and touch device cables, two devices will appear on the system. The video driver will handle the display and UPDD will handle the touch.

When a touch device is initially discovered then by default touch will be used to move the cursor on the primary monitor, 'monitor 1', because there is no foolproof way for the driver to determine what monitor/desktop is associated with the touchscreen. On a single monitor system then obviously the touch/desktop association will be correct but in a multi-monitor layout it is likely that the association is incorrect, hence the need to run Configure to make the correct association.

However, in situations where the monitor is listed in the OS with a unique display name, this name can be preset against the touch device such that the driver will automatically associate the touch device with the named monitor. With version 6.0.277 we have introduced a setting, monitor_bind.display_name, to allow this automatic binding of a
Multi-monitor and device support

touchscreen to a display name such that the association will be made between the
touchscreen and the monitor whenever it is discovered on the system.

This is best understood by the example below. We have a Lenovo, ThinkVision LT1423
touch monitor. In UPDD we have configured this device with the same name. When this
device is plugged into a Mac system the monitor is listed in the display properties as LEN
LT1423pwC so we have set this name in the monitor bind key setting. Now, when the
monitor and touch screen are plugged in to a Mac system the touch device will automatically
be associated with the correct monitor/desktop. In the example below, the monitor is a
secondary monitor, monitor 2, and the touch / desktop association has been correctly made.

The UPDD Console and the command line interface functions devices or monitors can be
used to list the monitor names.

Note that the monitor_bind.display_name specifies a display (screen) name substring. If a
value is specified then the device will associate itself with a screen with a name containing
that value. The matching is case insensitive.

e.g. if the value is 'XYZ' and a display named 'XYZ Multi touch' is found, this will be used.

This binding is checked every time there is a setting change notification; so the binding is
dynamic, i.e. will track monitor layout changes.

Note that in some cases the name of a monitor as reported by the OS to the driver might
differ from that seen in various system tools so you need to set the name as listed by the
driver's functions, such as listed in the UPDD Status.

Currently only supported in Windows and MacOS.

This feature is especially useful when the value is preset in the driver package as delivered
to the end user such that the touch / desktop association will be correctly made when the
device is connected to the system.
Hardware port binding

In a multi-device environment a hardware port will be required for each pointer device.

In a multi-monitor / multi-touch device environment it is important to be able to associate a specific touch device with a given monitor or desktop segment and ensure that the monitor/touch device association is retained over a reboot and, as best as possible, if devices switch ports or additional similar devices are added.

Each USB device handled by the driver is allocated a connection key. The format of this key is dependent on the characteristics of the controller, the port or hub on which it is physically connected and other settings, as seen in these two examples:

```
C:\Program Files (x86)\UPDD>upddutils get connection_key
```

```
C:\Program Files (x86)\UPDD>upddutils get connection_key
connection_key: 2149:5a77..bus:0001-port:0008.
```

The UPDD device setting 'bindmode' is a bit mask setting that indicates the components that are used to construct a bindkey. By default this is set to 32767 (7FFF) so all standard components are used in the bindkey.

```
#define BIND_VIDPID 0x01
#define BIND_USB_SERIAL 0x02
#define BIND_USB_LOCATION 0x04
```

Currently not all bits represent a component and are reserved for future use.

For USB devices the device/desktop binding method is different depending on the number of devices in use and the characteristics of the USB device as follows:

**Single device**

In all cases the binding between the controller and desktop will be maintained based on the USB vendor and product identification.
Multi-monitor and device support

**Unique devices**

In all cases where the USB device holds a unique serial number the binding between the controller and desktop will be maintained based on the USB vendor, product id and the serial number.

**Multiple devices**

The main issue arises when the USB devices are identical in that they all share the same vendor and product id but have no unique serial number. In this instance as long as the device remains in the same port the desktop and device association will be retained. This cannot be guaranteed in cases when, for whatever reason, the port number switches.

**API considerations**

When using the UPDD API in a multi-monitor environment you may need to associate received touch data with the source monitor in which case you need to be aware of the UPDD monitor metrics setting.

**Interlock functionality**

When dealing with multiple touch devices in desktop systems that are designed for single user usage the driver needs to cater for simultaneous touches being made on different devices and ensure there is a clean switch between one device and another.

The driver implements an interlock feature as described [here](#).

It is catered for if UPDD Gestures / UPDD Commander is used in Mac OS X or UPDD Virtual HID interface in Windows and implements an interlock feature such that control can only be given to a device if there is currently no touch data being received from a different device.

**Operating system specifics**

Please now select the appropriate link for further details on multi-monitor usage within each different operating system:

Touch-Base Support