



Linux, X.Org, and Kortek Touch

Kortek

Revision 1: 2020/07/16, Hugh Chang

Table of Contents

1. INTRODUCTION.....	2
2. IDENTIFICATION OF THE TOUCH DEVICES.....	3
2.1. HIDING UNNECESSARY INPUT DEVICES FROM X WINDOW SYSTEM.....	3
2.2. FIRMWARE CUSTOMIZATION FOR EASY IDENTIFICATION.....	5
2.3. EVENT DEVICE (EVDEV) IDENTIFICATION.....	6
3. MAPPING INPUT AND OUTPUT DEVICES IN AN EXTENDED DESKTOP ENVIRONMENT	8
3.1. EXTENDED DESKTOP.....	8
3.2. COORDINATE TRANSFORMATION MATRIX.....	9
3.3. XINPUT MAP-TO-OUTPUT.....	11
3.4. MAKING TRANSFORMATION MATRIX PERMANENT.....	13
4. MULTI-SEAT.....	14
4.1. MULTI-SEAT WITH MULTIPLE GRAPHIC CARDS (MULTI-HEADS).....	15
4.2. MULTI-SEAT WITH ONE GRAPHIC CARD (ZAPHOD HEADS).....	17
4.2.1. <i>ServerLayout with position information</i>	18
4.2.2. <i>ServerLayout without position information</i>	23
5. CONCLUSION.....	24



1. Introduction

This document describes how the Kortek touch devices can be used in the various environments of Linux and X.Org Server. There can be multiple Kortek touch monitors with other touch monitors and non-touch monitors for a machine in this environment.



2. Identification of the touch devices

2.1. Hiding unnecessary input devices from X Window System

When 2 Kortek touch devices are connected via USB for a machine, "lsusb" command shows 2 lines with Kortek USB Vendor ID (0x2965) and USB Product ID's (0x50**).

```
root# lsusb
Bus 003 Device 003: ID 2965:5024
Bus 003 Device 005: ID 2965:5023
```

However, it becomes 6 lines for "xinput list" command because a Kortek touch device is an USB composite device with 3 USB HID interfaces for multitouch, vendor extension, and mouse each where the HID vendor extension interface is shown as an UNKNOWN input device.

```
root# xinput list --short
[ Virtual core pointer          id=2  [master pointer (3)]
| ↳ Kortek Kortek Touch        id=10 [slave pointer (2)]
| ↳ Kortek Kortek Touch UNKNOWN id=11 [slave pointer (2)]
| ↳ Kortek Kortek Touch        id=12 [slave pointer (2)]
| ↳ Kortek Kortek Touch        id=13 [slave pointer (2)]
| ↳ Kortek Kortek Touch UNKNOWN id=14 [slave pointer (2)]
| ↳ Kortek Kortek Touch        id=15 [slave pointer (2)]
```

When you only use multitouch input devices, it is frustrating to see 6 lines instead of 2 lines from "xinput list" command and it would be good to hide the unnecessary USB HID interfaces. It is done by adding "ignore" option to the InputClass definition for Kortek touch devices in xorg.conf.

```
root# cat /etc/X11/xorg.conf
Section "InputClass"
    Identifier "Hide unnecessary USB HID interfaces of Kortek"
    MatchUSBID "2965:*"
    MatchIsTouchscreen "false"
    #MatchIsPointer "false"
```



```
Option "Ignore" "true"
EndSection
```

If “/etc/X11/xorg.conf” does not exist, it can be created or the InputClass definition can be put into another file for xorg configuration. For the detailed description, refer to the reference manual of xorg.conf, <https://www.x.org/releases/current/doc/man/man5/xorg.conf.5.xhtml>.

You can restart X server on Ubuntu with the following command:

```
root# service lightdm stop
root# service lightdm start
```

Then “xinput list” will show only 2 lines.

```
root# xinput list --short
[ Virtual core pointer          id=2  [master pointer (3)]
| ↳ Kortek Kortek Touch        id=11 [slave pointer (2)]
| ↳ Kortek Kortek Touch        id=10 [slave pointer (2)]

root# xinput list -long
| ↳ Kortek Kortek Touch        id=10 [slave pointer (2)]
    Detail for Valuator 0:
        Label: Abs MT Position X
        Range: 0.000000 - 4095.000000
    Detail for Valuator 1:
        Label: Abs MT Position Y
        Range: 0.000000 - 4095.000000
    Max number of touches: 10
| ↳ Kortek Kortek Touch        id=11 [slave pointer (2)]
    Detail for Valuator 0:
        Label: Abs MT Position X
        Range: 0.000000 - 4095.000000
    Detail for Valuator 1:
        Label: Abs MT Position Y
        Range: 0.000000 - 4095.000000
```

Max number of touches: 10

When you only use mouse devices instead of multitouch devices because your Linux does not support multitouch functionality, you can replace “MatchIsTouchscreen” to “MatchIsPointer” from the InputClass definition in xorg.conf.

```
root# cat /etc/X11/xorg.conf
Section "InputClass"
    Identifier "Hide unnecessary USB HID interfaces of Kortek"
    MatchUSBID "2965:*"
    #MatchIsTouchscreen "false"
    MatchIsPointer "false"
    Option "Ignore" "true"
EndSection
```

Then “xinput list” will show only 2 lines.

```
root# xinput list --long
| ↳ Kortek Kortek Touch      id=10 [slave pointer (2)]
    Detail for Valuator 0:
        Label: Abs X
        Range: 0.000000 - 4095.000000
    Detail for Valuator 1:
        Label: Abs Y
        Range: 0.000000 - 4095.000000
| ↳ Kortek Kortek Touch      id=11 [slave pointer (2)]
    Detail for Valuator 0:
        Label: Abs X
        Range: 0.000000 - 4095.000000
    Detail for Valuator 1:
        Label: Abs Y
        Range: 0.000000 - 4095.000000
```

2.2. Firmware customization for easy identification



You can ask for the customization of Kortek touch firmware to distinguish multiple touch devices easily. For example,

- Touch firmware customization of a main Kortek touch monitor
 - USB product string is changed to “Main5024” from “Kortek Touch”
 - USB Product ID is changed to 0x5024 from 0x5023
- Touch firmware customization of a small Kortek touch monitor
 - USB product string is changed to “Small5023” from “Kortek Touch”
 - USB Product ID is 0x5023

Then the output will be more readable.

```
root# xinput list --short
[ Virtual core pointer          id=2  [master pointer (3)]
|   ↳ Kortek Main5024          id=10 [slave pointer (2)]
|   ↳ Kortek Small5023        id=11 [slave pointer (2)]
```

2.3. Event device (evdev) identification

There are several ways to identify the related event device.

```
root# xinput list-props "Kortek Small5023"
Device 'Kortek Small5023':
  Device Product ID (256): 10597 (VID 0x2965), 20515 (PID 0x5023)
  Device Node (257):      "/dev/input/event4"
  Axis Labels (271):     "Abs MT Position X" (262), "Abs MT Position Y" (263), "None"
(0), "None" (0)

root# udevadm info --query=property --name=/dev/input/event4
DEVNAME=/dev/input/event4
ID_INPUT_TOUCHSCREEN=1
ID_MODEL=Small5023
ID_MODEL_ID=5023
ID_USB_INTERFACE_NUM=00
ID_VENDOR=Kortek
```



```
ID_VENDOR_ID=2965
```

```
root# cat /proc/bus/input/devices
```

```
I: Bus=0003 Vendor=2965 Product=5023 Version=0110
```

```
N: Name="Kortek Small5023"
```

```
P: Phys=usb-0000:00:14.0-1/input0
```

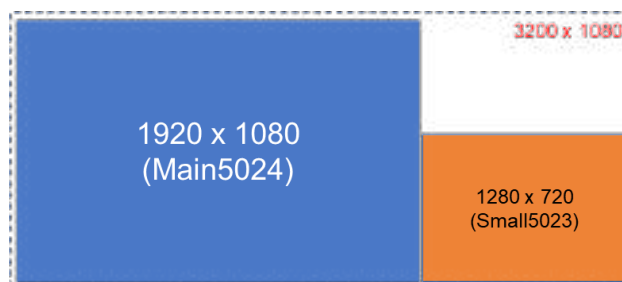
```
H: Handlers=mouse0 event4
```

```
B: ABS=2608000000000003
```

3. Mapping input and output devices in an extended desktop environment

3.1. Extended desktop

With extended desktop mode, the desktop is extended across multiple displays as one large virtual display. Each extended display has its own unique desktop area (including screen resolution and refresh rate). This setup can be used to provide additional desktop space. You can move applications from one display to the other.



“xrandr” is an official configuration utility to the RandR (Resize and Rotate) X Window System extension. It can be used to query the layout of each output area in a large virtual output area (<https://wiki.archlinux.org/index.php/Xrandr>).

```
root# xrandr
Screen 0: minimum 320 x 200, current 3200 x 1080, maximum 8192 x 8192
HDMI-1 connected primary 1920x1080+0+0
DP-1 connected 1280x720+1920+360
```


3.2. Coordinate transformation matrix

In this environment, the touch position is incorrect because each input area of a touch monitor is mapped to the large virtual output area. For how a position of an input device is translated into the position of an output device, you can refer to the following URL's for the coordinate transformation matrix.

- <https://wiki.ubuntu.com/X/InputCoordinateTransformation>
 - The default is 3x3 identity matrix

```
root# xinput list-props "Kortek Small5023" | grep "Coordinate Transformation Matrix"
Coordinate Transformation Matrix (142): 1.000000, 0.000000, 0.000000,
0.000000, 1.000000, 0.000000, 0.000000, 0.000000, 1.000000
```

- <https://www.x.org/releases/current/doc/man/man5/xorg.conf.5.xhtml#heading8>
 - Option "TransformationMatrix" "a b c d e f g h i"
 - ◆ Specifies the 3x3 transformation matrix for absolute input devices. The input device will be bound to the area given in the matrix. In most configurations, "a" and "e" specify the width and height of the area the device is bound to, and "c" and "f" specify the x and y offset of the area. The value range is 0 to 1, where 1 represents the width or height of all root windows together, 0.5 represents half the area, etc. The values represent a 3x3 matrix, with the first, second and third group of three values representing the first, second and third row of the matrix, respectively. The identity matrix is "1 0 0 0 1 0 0 0 1".
- <https://unix.stackexchange.com/questions/138168/matrix-structure-for-screen-rotation>

$$\begin{bmatrix} x_{out} \\ y_{out} \\ z_{out} \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_{in} \\ y_{in} \\ z_{in} \end{bmatrix}$$

with $z_{out} = z_{in} = 1$.

```
x_out = a * x_in + b * y_in + c
y_out = d * x_in + e * y_in + f
```


3.3. xinput map-to-output

The mapping between an input device and an output device can be done using “xinput map-to-output” command on Linux in a way that is independent of device vendors and touch device mode (absolute mouse or multitouch digitizer).

That is to say, “xinput map-to-output” command can automate the calculation of coordinate transformation matrix to map an input area to an output area.

```
root# xinput map-to-output "Kortek Main5024" HDMI-1
root# xinput map-to-output "Kortek Small5023" DP-1

root# xinput list-props "Kortek Main5024" | grep "Coordinate Transformation Matrix"
    Coordinate Transformation Matrix (142): 0.600000, 0.000000, 0.000000,
0.000000, 1.000000, 0.000000, 0.000000, 0.000000, 1.000000

root# xinput list-props "Kortek Small5023" | grep "Coordinate Transformation Matrix"
    Coordinate Transformation Matrix (142): 0.400000, 0.000000, 0.600000,
0.000000, 0.666667, 0.333333, 0.000000, 0.000000, 1.000000
```

We can verify it with our manual calculation for “Kortek Main5024” and “Kortek Small5023”.

```
<Kortek Main5024>

a = 1920 / 3200 = 0.6
b = 0.0
c = 0.0
d = 0.0
e = 1080 / 1080 = 1.0
f = 0.0
g = 0.0
h = 0.0
l = 1.0

x_out = x_in * 1920 / 3200
```

$$y_{\text{out}} = y_{\text{in}} * 1080 / 1080$$

<Kortek Small5023>

$$a = 1280 / 3200 = 0.4$$

$$b = 0.0$$

$$c = 1920 / 3200 = 0.6$$

$$d = 0.0$$

$$e = 720 / 1080 = 0.666667$$

$$f = 360 / 1080 = 0.333333$$

$$g = 0.0$$

$$h = 0.0$$

$$l = 1.0$$

$$x_{\text{out}} = x_{\text{in}} * 1280 / 3200 + 1920 / 3200 * 3200$$

$$y_{\text{out}} = y_{\text{in}} * 720 / 1080 + 360 / 1080 * 1080$$

3.4. Making transformation matrix permanent

“xinput map-to-output” command can be put in an initialization script to make the mapping permanent. There is an alternative method to make the mapping permanent using “TransformationMatrix” option in xorg.conf.

- https://wiki.archlinux.org/index.php/Talk:Calibrating_Touchscreen
- <https://www.x.org/releases/current/doc/man/man5/xorg.conf.5.xhtml>

```
root# cat /etc/X11/xorg.conf

Section "InputClass"
    Identifier      "Hide unnecessary USB HID interfaces of Kortek"
    MatchUSBID     "2965:*"
    MatchIsTouchscreen "false"
    Option         "Ignore" "true"
EndSection

Section "InputClass"
    Identifier      "Kortek touch for Main5024"
    MatchUSBID     "2965:5024"
    MatchDevicePath "/dev/input/event*"
    Driver         "evdev"
    Option "TransformationMatrix" "0.6 0 0 0 1 0 0 0 1"
EndSection

Section "InputClass"
    Identifier      "Kortek touch for Small5023"
    MatchUSBID     "2965:5023"
    MatchDevicePath "/dev/input/event*"
    Driver         "evdev"
    Option "TransformationMatrix" "0.4 0 0.6 0 0.666667 0.333333 0 0 1"
EndSection
```

4. Multi-seat

- <https://lightofdawn.org/wiki/wiki.cgi/-wiki/LinuxMultiSeat>
 - A multi-seat system is a system with multiple input/output devices where each monitor/keyboard/mouse combination works as if it is independent computer. A three-seat system is one physical computer, with 3 screens and 3 keyboards and 3 mice, and each screen/keyboard/mouse acts as if they are separate computers (ie each screen will have its own mouse pointer, moving a mouse only moves the pointer on its own screen and not the others, same thing for keyboard input etc).
 - Output system
 - ◆ multiple graphic cards each supporting a single monitor (=multi-heads)
 - ◆ one graphic card with support for multiple monitors (=zaphod heads)
 - Input system
 - ◆ Every keyboard and every mouse plugged will have its own evdev devices. To enable multi-seat, it is enough to create a logical group somewhere that says "evdev devices for keyboard #1 and mouse #1" is for seat 1, "evdev devices for keyboard #2 and mouse #2" is for seat 2; and when it is time to combine output/input together, just use this configuration.

4.1. Multi-seat with multiple graphic cards (multi-heads)

- https://wiki.debian.org/Multi_Seat_Debian_HOWTO
 - Loginctl
 - ◆ The easy, modern way of setting up local multiseat requires no editing of configuration files at all. The init software systemd contains a utility called loginctl which can be used for setting up seats.
 - ◆ Setting up multiseat follows this pattern:
 - Use loginctl to determine available devices
 - Locate video card in device list, assign to seat
 - Reboot
 - Assign additional devices
 - ◆ When creating seats, loginctl sets up rules in the background only when a video card is a part of the original assignment. Without a videocard, no new seats will be created.

- https://wiki.archlinux.org/index.php/Xorg_multiseat
 - Multiseat is a certain setup where N users work simultaneously on one computer. This is achieved by having N monitors, N keyboards and N mice.
 - Attaching devices to a seat
 - ◆ Identify devices
 - `loginctl seat-status seat0`
 - ◆ Assign devices (graphics card)
 - `loginctl attach seat1 /sys/devices/pci0000:00/0000:00:04.0/0000:02:00.0/drm/card1`
 - `loginctl attach seat1 /sys/devices/pci0000:00/0000:00:04.0/0000:02:00.0/graphics/fb1`
 - ◆ Assign devices (two USB ports are assigned to the seat for USB input devices)
 - `loginctl attach seat1 /sys/devices/pci0000:00/0000:00:12.1/usb4/4-1`
 - `loginctl attach seat1 /sys/devices/pci0000:00/0000:00:12.1/usb4/4-2`
 - ◆ Verify seats
 - `loginctl seat-status seat0`
 - `loginctl seat-status seat1`
 - Then there will be multiple X servers on each seat with "DISPLAY=:N.0"

4.2. Multi-seat with one graphic card (zaphod heads)

- http://www.odorik.cz/w/linux_multihead
 - More screens on one graphic card
 - ◆ To create X Window screen for each graphic card output (DISPLAY=":0.0", ":0.1") special parameter ZaphodHeads in xorg.conf must be used.
- http://web.archive.org/web/20130429073111/http://en.gentoo-wiki.com/wiki/X.Org/Dual_Monitors#Single_graphics_card.2C_Multiple_X_screens_with_ZaphodHeads
 - Single graphics card, Multiple X screens with ZaphodHeads
 - ◆ This will, for instance, allow you to launch one program on a given display with DISPLAY=:0.0, and on another with DISPLAY=:0.1.

4.2.1. ServerLayout with position information

- <https://www.x.org/releases/X11R7.0/doc/html/DESIGN2.html>
 - ServerLayout section
 - ◆ The preferred way of specifying the layout is to explicitly specify the screen's location in absolute terms or relative to another screen.
- <https://www.x.org/releases/current/doc/man/man5/xorg.conf.5.xhtml>
 - SERVERLAYOUT SECTION
 - ◆ Screen screen-num "screen-id" position-information
 - The position-information field describes the way multiple screens are positioned.

When you use one graphic card for multiple screens using ZaphodHeads option, the layout of the screens is similar to that of the extended desktop setup.

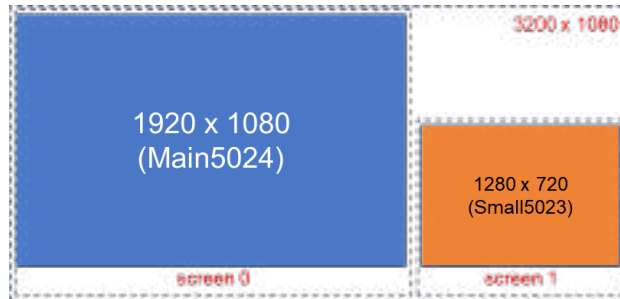
```
root# cat /etc/X11/xorg.conf

Section "ServerLayout"
    Identifier "LayoutDual"
    Screen 0 "Screen0" 0 0
    Screen 1 "Screen1" 1920 360
EndSection

Section "Screen"
    Identifier "Screen0"
    SubSection "Display"
        Modes "1920x1080"
    EndSubSection
EndSection

Section "Screen"
    Identifier "Screen1"
    SubSection "Display"
        Modes "1280x720"
```

```
EndSubSection
EndSection
```



However, the origin of each output device will be (0, 0).

```
root# xrandr -d :0.0 -q

Screen 0: minimum 8 x 8, current 1920 x 1080, maximum 32767 x 32767
HDMI1 connected primary 1920x1080+0+0

root# xrandr -d :0.1 -q

Screen 1: minimum 8 x 8, current 1280 x 720, maximum 32767 x 32767
DP1 connected primary 1280x720+0+0
```

So, the same TransformationMatrix setting with the extended desktop setup will work for ZaphodHeads setup if ServerLayout is written with position information for each screen.

```
root# cat /etc/X11/xorg.conf

Section "InputClass"
    Identifier "Kortek touch for screen0"
    MatchUSBID "2965:5024"
    MatchDevicePath "/dev/input/event*"
    Driver "evdev"
    Option "TransformationMatrix" "0.6 0 0 0 1 0 0 0 1"
EndSection
```



```
Section "InputClass"
    Identifier "Kortek touch for screen1"
    MatchUSBID "2965:5023"
    MatchDevicePath "/dev/input/event*"
    Driver "evdev"
    Option "TransformationMatrix" "0.4 0 0.6 0 0.666667 0.333333 0 0 1"
EndSection
```

The following example is a complete "xorg.conf" that was verified to work.

```
root# cat /etc/X11/xorg.conf

Section "ServerLayout"
    Identifier "LayoutDual"
    Screen 0 "Screen0" 0 0
    Screen 1 "Screen1" 1920 360
EndSection

Section "ServerFlags"
    Option "DefaultServerLayout" "LayoutDual"
EndSection

Section "InputClass"
    Identifier "Hide unnecessary USB HID interfaces of Kortek"
    MatchUSBID "2965:*"
    MatchIsTouchscreen "false"
    #MatchIsPointer "false"
    Option "Ignore" "true"
EndSection

Section "InputClass"
    Identifier "Kortek touch for screen0"
    MatchUSBID "2965:5024"
    MatchDevicePath "/dev/input/event*"
EndSection
```

```
Driver "evdev"
Option "TransformationMatrix" "0.6 0 0 0 1 0 0 0 1"
EndSection

Section "InputClass"
Identifier "Kortek touch for screen1"
MatchUSBID "2965:5023"
MatchDevicePath "/dev/input/event*"
Driver "evdev"
Option "TransformationMatrix" "0.4 0 0.6 0 0.666667 0.333333 0 0 1"
EndSection

Section "Screen"
Identifier "Screen0"
Device "Intel0"
Monitor "Monitor0"
SubSection "Display"
Modes "1920x1080"
EndSubSection
EndSection

Section "Screen"
Identifier "Screen1"
Device "Intel1"
Monitor "Monitor1"
SubSection "Display"
Modes "1280x720"
EndSubSection
EndSection

Section "Monitor"
Identifier "Monitor0"
VendorName "Vendor0"
ModelName "Model0"
EndSection
```

Section "Monitor"

Identifier "Monitor1"

VendorName "Vendor1"

ModelName "Model1"

EndSection

Section "Device"

Identifier "Intel0"

Driver "intel"

BusID "PCI:0:2:0"

Screen 0

Option "ZaphodHeads" "HDMI1"

Option "NoAccel" "true"

EndSection

Section "Device"

Identifier "Intel1"

Driver "intel"

BusID "PCI:0:2:0"

Screen 1

Option "ZaphodHeads" "DP1"

Option "NoAccel" "true"

EndSection

4.2.2. ServerLayout without position information

When you use one graphic card for multiple screens using ZaphodHeads option, what happens if position information is not provided in ServerLayout of xorg.conf?

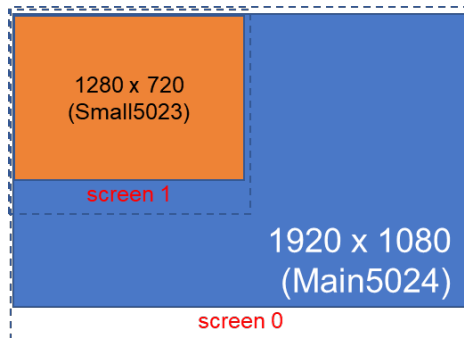
```
root# cat /etc/X11/xorg.conf

Section "ServerLayout"
    Identifier "LayoutDual"
    Screen 0 "Screen0"
    Screen 1 "Screen1"
EndSection

Section "Screen"
    Identifier "Screen0"
    SubSection "Display"
        Modes "1920x1080"
    EndSubSection
EndSection

Section "Screen"
    Identifier "Screen1"
    SubSection "Display"
        Modes "1280x720"
    EndSubSection
EndSection
```

- <https://www.x.org/releases/X11R7.0/doc/html/DESIGN2.html>
 - ServerLayout section
 - ◆ In the absolute case, the upper left corner's coordinates are given after the Absolute keyword. If the coordinates are omitted, a value of (0, 0) is assumed.



This results in the overlapped screens and there are multiple possible positions on multiple overlapped screens for a (x, y) position. When a finger touches the screen with a touch controller, the visual pointer will be shown on a screen with higher resolution.

The proper configuration of input device in xorg.conf with this setup is not found at least to me.

5. Conclusion

When there are multiple input and output devices in various Linux and X.Org environments, the touch positions should be scaled and shifted to match the corresponding output device. It can be done by setting TransformationMatrix option in xorg.conf and “xinput map-to-output” command can help to get the values of the transformation matrix. It is independent of device vendors and touch device mode (absolute mouse or multitouch digitizer). There is no need to make a proprietary daemon program for a touch device that reads touch events, calculates new positions, and puts them to uinput virtual device because the equivalent action can be done by setting transformation matrix values in xorg.conf.

The only exception is when ServerLayout is described without position information for multi-seat setup with one graphic card (zaphod heads). A proper solution for input device configuration is not found yet.

Revision History

2.1 HIDNG U	ECS.	ARSYI PH	T HSI PV
Revision 1	2020/07/16	Hugh Chang	Initial revision