Diebold Nixdorf

User Manual

UTD

Universal Touch Driver

Linux (DNL)

UTD

Universal Touch Driver

Linux (DNL)

User Manual

Edition 2.3.3 Revision 1, March 2023

Content

Introduction

The Universal Touch Driver for DNL 1 supports the Diebold Nixdorf touch screens D1xxx /pcTouch, D2xxx /pcTouch, BA9x /pcTouch, BA9x /rTouch and the touch controllers of Diebold Nixdorf All-In-One systems BEETLE A1xxx, BEETLE / iPOS plus (Advanced/XL/SL) , BEETLE /iScan, Diebold Nixdorf kiosk and others (see chapter “Identification of Touch Monitors” on page 6).

It includes the following features:

- Different touch modes
- Beep and sound on touch
- Multi monitor support
- Calibration
- Edge acceleration
- Touch delay
- Left click duration

This manual applies to version 2.3.3 of the driver.

Used Symbols

Notes and important information in this manual are marked by this symbol.

Identification of the Driver Version

Run the command

```
rpm -qi dntouch dntouch-support dntouch-udev
```

to identify the installed driver version.

Identification of Touch Monitors

This driver supports the following Diebold Nixdorf touch screens. They can be identified by the name, the type label or by its USB IDs. The USB ID can be read in the operating system.

Touch screen name	Type label	USB IDs
- D2150 - D2156 - D1185, A12xx - D1150, A10xx - D1156, A11xx		Vendor=222A Product=546A Product=546B Product=546C Product=546D Product=546E
- D1101 - D1150, A10xx - D1156, A11xx		Vendor=0EEF Product=C11C or Product=C11E or Product=C11D
- BA90 /pcTouch - BA92 /pcTouch - BA93w /pcTouch - iPOS+ /pcTouch		Vendor=0eef Product=7200 Product=B10D Product=B114 Product=B115 Product=B116 Product=C116 Product=C113 Product=C114
- BA92 /rTouch - BA93 /rTouch - BA9x /rTouch		Vendor=0eef Product=0001
- BA91w /pcTouch		Vendor=0eef Product=B10C or Product=B110

- BEETLE iPOS+ Advanced /rTouch - BEETLE iPOS+ Advanced /pcTouch - BEETLE iPOS+ Advanced /pcTouch 10pt		Vendor=0eef Product=0001 or Product=7201 or Product=B100 or Product=B10E or Product=B10F
- BEETLE iPOS+ SL - BEETLE iPOS+ XL		Vendor=0EEF Product=B113 or Product=C111
- Kiosk Display	15" Display 19" PCAP Display 22" PCAP Display 27" PCAP Display 32" PCAP Display	Vendor=0EEF Product=72C4 Product=7904 Product=A100 Product=C000
- BA82 /irTouch - BA82 /e /irTouch - BA83 /irTouch - BA83 /e /irTouch		Vendor=6615 Product=0012
- BA80 /rTouch		Vendor=1bfd Product=1688

Installation of the Touch Driver on DNL 1

- Use files from `noarch` and `x86_64`

Installing RPM

- open a terminal / shell as root user
- delete any preinstalled dntouch packages
  >> `rpm -e dntouch`
  >> `rpm -e dntouch-support`
  >> `rpm -e dntouch-udev`
  or whatever name the installed rpm has when listed by
  >> `rpm -qa | grep touch`
- install platform-independent scripts and settings (e.g. `dntouch-udev-xyz.rpm`)
  >> `rpm -i dntouch-udev-xyz.rpm`
  (or: `rpm -iv dntouch-udev-xyz.rpm`)
  or whatever name and version the rpm has
- install current touch driver and support package (e.g. `dntouch-xyz.rpm` and `dntouch-support-xyz.rpm`)
  >> `rpm -i dntouch-xyz.rpm dntouch-support-xyz.rpm`
  (or: `rpm -iv dntouch-xyz.rpm`)
  or whatever name and version the rpm has.
  For the supporting graphical configuration tools, the QT5 libraries need to be installed (os-dependent packages, i.e.
  `yum install qt5-qtbase-gui`
  on DNL 1)
- The dntouch daemon process will be started automatically by the rpm installation process. To (re-)start the daemon manually run
  >> `/etc/rc.d/init.d/dntouch restart`
  The resulting output should contain '[OK]'
  Please note that some settings on older OS versions require the X-Server to be restarted just once after the initial installation, in order for the device auto detection to become effective. The driver may

need some time to allocate the raw touch devices and transfer control to the user interface.

Configuring the Daemon

The directory `/etc/opt/dn/dntouch` will be filled with template files during installation. If it is empty, the driver will still be able to run, but using default settings (default multi-screen setup, no advanced settings, no calibration).

The easiest way to change the initial settings is opening a terminal and running the touchscreen configuration tool as administrator.

>> `sudo /opt/dn/dntouch/bin/dnconfig`

Select the respective devices you want to use, touch mode, beep mode and the like and click 'OK'.

Note how the two files `/etc/opt/dn/dntouch/devices` and `/etc/opt/dn/dntouch/options` have been created.

Configuring the Touch-Input device and Monitor mappings

The supporting tool for mapping input devices on monitors, thus scaling the devices resolutions with the monitor screen offset and its own resolution for a perfect match between screens touch position and graphical cursor, is `/opt/dn/dntouch/bin/dnmapping`.
This tool is also called by the dntouch daemon and the systems hardware detection daemon udev whenever touchscreen-related hardware changes occur.

dnmapping communicates with the calibration interface of the driver to create a temporary mapping of monitors and touch point offsets based on monitor resolution. If fixed calibration files exist in /etc/opt/dn/dntouch/calibrationX, the daemon will use these calibration settings instead of attempting auto-calibration.

An initial setting can be created manually by

>> sudo `/opt/dn/dntouch/bin/dnmaping --list --out`

which appends a suggestion for the currently detected setup to the mappings configuration file `/etc/opt/dn/dntouch/mappings`, which should then be edited to reflect the real situation.

Run `/opt/dn/dntouch/bin/dnmapping` again after changing file `/etc/opt/dn/dntouch/mappings`.

In some graphic configurations it can be necessary to call additional xinput commands. For these cases a script-hook to `/etc/opt/dn/dntouch/postmapping.sh` is implemented in `dnmaping`. A sample can be found in the same directory `/etc/opt/dn/dntouch/postmapping.sh.template`

Device Configuration File

The device file is needed to define a specific order in which the driver addresses the touch controllers. This is important to configure multi-screen setups. The file is located at `/etc/opt/dn/dntouch/devices`. It can be created by the configuration tool (see above).

It has to contain the sysfs_paths of the devices to be handled by the driver daemon, which represent the USB ports the touch devices are connected to. These are handed to the daemon as command line parameters by the daemon start script.

The device file should be empty in case only one touch screen is connected to the system.
The devices configuration file needs to be created again if there is any change in the USB port configuration.

If you intend to create it manually, you can run (as root user) the following command to get a list of supported devices in the devices file, and then deleted the devices you do not need or want.

```
>> /opt/dn/dntouch/sbin/dntouch --list >
/etc/opt/dn/dntouch/devices
```

/etc/opt/dn/dntouch/devices example:

```
/sys/devices/pci0000:00/0000:00:1a.7/usb1/1-2/1-2.2/1-
2.2.1/1-2.2.1:1.0
/sys/devices/pci0000:00/0000:00:1a.7/usb1/1-2/1-2.1/1-
2.1.1/1-2.1.1:1.0
```

Options File

The options file is located at `/etc/opt/dn/dntouch/options`.
It can be created by the configuration tool (see above).
It contains options handed to the driver as command line parameters by the daemon start script.
It may contain the following options and settings:

Setting the touch mode

`TOUCH_MODE=(touch|release|mouse|multitouch|passthrough)`
The default mode is 'mouse'.
Please note that the user interface may hide the mouse pointer in touch and multitouch modes, depending on the window manager.

`touch` Execute click event when finger touches the screen.

`release` Execute click event when finger is released from the screen.

`mouse` Execute pressed, moved and released events like a mouse. The user interface may show a mouse pointer in this mode, regardless of the fact that a touch device is used where the pointer is normally hidden. This is the desired behavior.

`multitouch` Pass multitouch events (simultaneously pressed locations on screen in parallel), still applying calibration and filters such as edge acceleration.

`passthrough` Pass through all events from the attached touch device unfiltered, no calibration and edge acceleration are applied. In multi-monitor setups, this may be the wrong mode since the OS will try to map each touch surface to the entire size of the virtual screen.

Beep on click settings

Option a – global setting:

`BEEP_PER_DEVICE` has to be commented out with #
be not present in the options file be initialized to 0, no, off

Option a1 – PC-Speaker:

```
BEEP_TYPE=system
BEEP_DEVICE=<console>
```
(e.g.: `BEEP_DEVICE= /dev/console`)

Option a2 – Soundcard:

```
BEEP_TYPE=alsa
SND_FILE=/path/to/file/sound.wav
PCM_NAME=<sound device>
```
(e.g.: `PCM_NAME=default` or `PCM_NAME=hw:1,0`)

Option b – setting per touch device

Please note, that the parameters have to be numbered with _1 und _2 and not with _0 und _1.

```
BEEP_PER_DEVICE=(1|TRUE|ON|YES)
```

Option b1 – PC-Speaker:

```
BEEP_TYPE_1=system
```
and/or
```
BEEP_TYPE_2=system
```

```
BEEP_DEVICE_1=<console>
```
and/or
```
BEEP_DEVICE_2=<console>
```
(e.g.: `BEEP_DEVICE_1= /dev/console`)

Option b2 – Soundcard(s):

```
BEEP_TYPE_1=alsa
```
and/or
```
BEEP_TYPE_2=alsa
```

```
SND_FILE_1=/path/to/file/some_sound.wav
```
and/or
```
SND_FILE_2=/path/to/file/another_sound.wav
```

```
PCM_NAME_1=<sound device>
```
and/or
```
PCM_NAME_2=<sound device>
```
(e.g.: `PCM_NAME_2=default` or `PCM_NAME_2=hw:1,0`)

Combinations of b1 and b2 are possible.

Edge acceleration settings

Option 1 – global settings:

`EDGE_RANGE_V=<edge height in Y direction [screen px]>`
(e.g.: `EDGE_RANGE_V=50`)

`EDGE_RANGE_H=<edge width in X direction [screen px]>`
(e.g.: `EDGE_RANGE_H=73`)

`EDGE_MUL=<acceleration multiplier for all edges>`
(e.g.: `EDGE_MUL=1.5`)

Option 2 – settings per touch device

Please note, that the parameters have to be numbered with _1 und _2 and not with _0 und _1.
`EDGE_PER_DEVICE=(1|TRUE|ON|YES)`
Set the following values (if desired) as described above:

```
EDGE_RANGE_V_1=
EDGE_RANGE_H_1=
EDGE_MUL_1=

EDGE_RANGE_V_2=
EDGE_RANGE_H_2=
EDGE_MUL_2=
```

Device selection settings

The daemon will try to find input queues that correspond to the sysfs paths provided on the command line. If started via runcontrol script these are the ones provided in `/etc/opt/dn/dntouch/devices`.
If there are less than two devices given in, it will try to find further supportable devices connected to the system.
Please be aware that the devices given take up their respective slots even if not connected to the system, thus modifying the order and gaps in the numbering of emulated dntouch0 … dntouch7 devices.

To avoid devices not defined on the command line from being supported, use the following setting:

```
STRICT_DEVICE_SELECTION=(1|TRUE|ON|YES)
```

Touch/Release delay settings

The touch/release delay will prevent short touches (in click-on-touch mode) and releases (in click-on-release mode) to issue a click event.

The default setting for the delay is 50 msec.
The setting can be adjusted with the following parameter
`TOUCH_DELAY=<touch/release delay [msec]>` (e.g. `TOUCH_DELAY=75`)
The minimum delay is 0 msec.
The maximum is 300 msec.
Please note that delays may be modified by the graphics server in use, i.e. Xorg or Wayland.

MinLeftclickDuration setting

In click-on-touch mode the minimum left click duration defines the time (in ms) between the faked touch and release events passed to the OS when the touch device was touched.

The default setting is 10 msec.
The setting can be adjusted with the following parameter
`MIN_LEFTCLICK_DURATION=<min. left click duration [msec]>`
`(e.g. MIN_LEFTCLICK_DURATION=15)`
The minimum value is 0 msec.
The maximum value is 2000 msec.

MaxLeftclickDuration setting

In click-on-release mode the maximum left click duration defines the time (in ms) between the faked touch and release events passed to the OS when the touch device was released.

The default setting is 200 msec.
The setting can be adjusted with the following parameter
`MAX_LEFTCLICK_DURATION=<max. left click duration [msec]>`
`(e.g. MAX_LEFTCLICK_DURATION=250)`
The minimum value is 0 msec.
The maximum value is 2000 msec.

Debug Level setting

The dntouch daemon can be set to different levels of writing runtime information to its logfile at `/var/log/dn/dntouch/dntouch.log` using the `DEBUG_LEVEL` variable:

`0` - Nothing will be logged
`1` - Fatal errors and attached devices (default)
`2` - Also show warnings
`3` - Also show events and decoded touch/mouse protocol information

Example

`/etc/opt/dn/dntouch/options` example:

```
TOUCH_MODE=touch
DEBUG_LEVEL=2
EDGE_RANGE_H=75
EDGE_RANGE_V=75
EDGE_MUL=1.5
BEEP_TYPE=system
TOUCH_DELAY=20
MIN_LEFTCLICK_DURATION=15
MAX_LEFTCLICK_DURATION=250
```

Daemon Calibration

After installation and configuration, touch devices should match the corresponding monitors, depending on the positioning and connector sequence detected by dnmapping. In rare cases, touch position and cursor position may be off by a few pixels, or the desired positioning of monitors does not match the touch screen position. In these cases, the calibration tool can be used while the dntouch daemon is running:

```
>> /opt/dn/dntouch/bin/dncalibrate
```

By now, there should be files `/etc/opt/dn/dntouch/calibrationX` for every active device (X is the device number: 'calibration0', 'calibration1' and so on). The saved calibration files will override dnmapping's attempted auto-configuration of screens and touch devices.

The calibration files contain position and size of

- the virtual screen (measured in touchscreen coordinates)
- the actual screen (measured in display pixel)
- the virtual screen (measured in display pixel)

If the calibration was properly conducted, all necessary calibration data is written to the respective calibration files.

In DNL 1, the (default) Wayland graphics system does not provide a possibility to auto-map touch devices and monitors, so the calibration feature of dntouch is used not only for correcting small offsets of touch points and pointer positions, but also for assigning monitors to touch devices. See “Touchcontroller vs. Monitor Configuration”.

Failure to match a touch device with a freshly connected monitor is usually caused by a misconfiguration of the `mappings` or `devices` file in `/etc/opt/dn/dntouch`.

See chapter “Calibration Tool” for more information.

X-Server Configuration

Now even though the driver daemon might be running (after a restart with the correct settings), the X-Server may not yet know about the inputs it provides.

During installation, a configuration file is being added to `/etc/X11/xorg.conf.d/`:

```
Section "InputClass"
        Identifier  "dntouch"
        Driver      "evdev"
        MatchDevicePath   "/dev/input/dntouch*"
EndSection
```

This entry tells the X-Server to automatically add devices emulated by the dntouch daemon. Other entries `/etc/X11/xorg.conf` in are not needed.

If the dntouch packages were installed for the first time after the X-Server has already been started already, it may be necessary to restart X11 because the X-Server only reads its configuration files once during start up.

Touchcontroller vs. Monitor Configuration

`/opt/dn/dntouch/bin/dnmapping` is a tool for mapping touch controllers to monitors, automatically setting resolution and position. Its configuration file is `/etc/opt/dn/dntouch/mappings`, containing combinations of touch controller and monitor socket names, separated by a single space character, no quotes. The available monitor socket names depend on the graphic chip and can be listed with the `xrandr` command. A sample initial setup based on "first detected, first listed" can be created by
`>> sudo /opt/dn/dntouch/bin/dnmapping --list --out`
which will append the current findings to
`/etc/opt/dn/dntouch/mappings`.

The dnmapping tool has several options which will be used in different scenarios:

>> `/opt/dn/dntouch/bin/dnmapping --help`

```
Usage: /opt/dn/dntouch/bin/dnmapping options
 -a   | --auto        Automatic choice of dntouch-only or general mappings
 -l   | --list        Print autoconfigured mappings
 -o   | --out         Append current mappings to config file
 -n   | --nolvds      Skip internal LVDS entries
 -r   | --reverse     Reverse order of screens
 -w   | --dntouch     Only map dntouch devices from mappings file
 -d   | --debug       Print verbose debug info
 -t s | --time s      Wait s seconds
 --disable          Detach non-dntouch devices from X server
 --enable           Re-Attach non-dntouch devices to X server
 -h | --help        This help
```

If called manually with no parameters, dnmapping will map all touchdevice - monitor combinations in `/etc/opt/dn/dntouch/mappings` and add a "best guess" mapping for those devices not listed in the config file. It can also be used independently of the dntouch daemon to map the "raw" touch controller devices.

It is not harmful to list both dntouchX as well as the original device names in /etc/opt/dn/dntouch/mappings, as raw devices attached to dntouch-

provided devices will be automatically get deactivated by dntouch if the daemon is running.

A sample `/etc/opt/dn/dntouch/mappings` file running Wayland:

```
dntouch0 XWAYLAND1
dntouch1 XWAYLAND0
```

A sample `/etc/opt/dn/dntouch/mappings` file running Xorg:

```
dntouch0 DP-1
dntouch1 e DP-1
```

With version 2.3.3 the search algorithm for the touch controllers has been changed to improve the detection speed significantly. The order in which the touch controllers are found (and the device name wntouchX) can be changed by that if there was no `devices` file created.
Please check the device names used in the `mappings` file if the package is updated to version 2.3.3 or later.

Right-Click Emulation

Button emulation currently does not work via dnemulate3 when the Wayland graphics system is in use, since Wayland does not expose individual touch devices to user space via xinput.
Use the multi-touch features of the window manager to use Right-Click Emulation.

Restart

- stop the dntouch daemon

```
>> /etc/rc.d/init.d/dntouch stop
```

- start the dntouch daemon

```
>> /etc/rc.d/init.d/dntouch start
```

- restart the Wayland/Xorg server to restart the graphical interface

Note that restarting X is only necessary the first time after the `/etc/X11/xorg.conf.d/99-dntouch.conf` config file has been installed. For other changes in the configuration of dnconfig, it is sufficient to just restart dntouch by its start/stop script:

```
>> sudo /etc/rc.d/init.d/dntouch restart
```

which will be done automatically when saving a new configuration using the ddnconfig tool.

It may be necessary to call /opt/dn/dntouch/dnmapping again to make sure the autoconfiguration in multi-monitor setups is applied again, in case no fixed calibration has been saved using the dncalibrate tool

Troubleshooting

Pointers and screens do not match

If after restart the pointer moves out on the other screen,
you probably only must exchange or remove the `calibrationX` files in folder `/etc/opt/dn/dntouch`, or rerun /opt/dn/dntouch/bin/dncalibrate

Beep-on-Click Wav-File mixing

If the beep-on-click functionality of the driver shall be used, it might be necessary to create PCM sound devices that point to the desired PCM hardware, but feature mixing capabilities. Otherwise, no click sound can be played until the last has been fully completed.

After installation, the file `/etc/asound.conf` includes the file `/etc/alsa/pulse-default.conf` that looks like:

```
# PulseAudio plugin configuration
# $Id: pulse-default.conf,v 1.3 2008/03/09 15:50:49
lkundrak Exp $

# Let's create a virtual device "pulse" for mixer and
PCM

pcm.pulse {
       type pulse
       hint {
              description "PulseAudio Sound Server"
       }
}

ctl.pulse {
       type pulse
       hint {
              description "PulseAudio Sound Server"
       }
}

# Let's make it the default!

pcm.!default {
       type pulse
       hint {
              description "Default"
       }
}


ctl.!default {
       type pulse
       hint {
              description "Default"
       }
}
```

and some lines should be added to make it look like:

```
# PulseAudio plugin configuration
```

```
# $Id: pulse-default.conf,v 1.3 2008/03/09 15:50:49
lkundrak Exp $

# Let's create a virtual device "pulse" for mixer and
PCM

pcm.pulse {
       type pulse
       hint {
              description "PulseAudio Sound Server"
       }
}

ctl.pulse {
       type pulse
       hint {
              description "PulseAudio Sound Server"
       }
}

ctl.!default {
       type pulse
       hint {
              description "Default"
       }
}

pcm.dmixer0 {
       type plug
       slave.pcm "dmix:0,0"
}

pcm.dmixer1 {
       type plug
       slave.pcm "dmix:1,0"
}

pcm.!default {
       type plug
       slave.pcm "dmix:1,0"
       hint {
              description "Default"
       }
}
```

Now the two devices '`dmixer0`' and '`dmixer1`' can be used for output to the two available sound cards (internal and USB).
Please note that you might need to adapt this to your system setup and soundcards available and/or desired for output.

The Touch Driver in Detail

Architecture:

The touch controllers are exposed as evdev devices in `/dev/input/eventX`. By default, they are recognized as mouse by the kernel and all events are routed to a `/dev/input/mouseX` device which is also automatically used by the Wayland graphics server. Some newer touchscreen devices may not be recognized at all by the X-Server though, because of an unknown product/vendor id, these will only be functional with a version of dntouch that knows and translates their events to mouse clicks or touch events.

The raw input events generated by the original device files do not allow any
calibration and advanced processing support, therefore dntouch inserts an additional abstraction
layer by means of the developed driver. As stated by the requirement, this abstraction layer is completely in user-space and implemented with a daemon program called **dntouch**. dntouch will scan all devices in `/dev/input/` for supported touchscreen controllers. Whenever it finds a matching input device it will open it and request for exclusive access. This will stop further processing of the incoming events by the kernel and conversion to mouse events. To allow Xorg to use the device, dntouch will create a new evdev device named `dntouchX` (the internal evdev name of a device can be seen using the evtest tool). When using Wayland instead of Xorg, these internal devices names are not exposed via xinput, but can be shown using “sudo libinput list-devices”. Additionally, dntouch will create a link in `/dev/input/dntouchX` to the newly created `eventX` device so that it can be referred directly in configuration files.

An accompanying user space tool called **dnmapping** is used to map monitors and touch devices correctly, using the calibration interface of dntouch, so the former manual screen calibration becomes optional in many cases. Button emulation is currently not supported since Wayland does not support xinput access to raw touchscreen devices.

The dntouch daemon is controlled by command line options which are parsed from variables in `/etc/opt/dn/dntouch/options`, which are evaluated and assembled into one large commandline by `/etc/init.d/dntouch start` on system boot time.

Mapping Tool

To allow automatic configuration changes and mappings of newly attached devices, the dnmapping tool is called by udev each time a new supported device is detected. Additionally, the dntouch daemon gets to re-enumerate its devices by receiving signal 1 (SIGHUP), when called by udev. In the reverse scenario, whenever dntouch needs to refresh its emulated device list, dnmapping will be called with options `--dntouch` and `--disable`, so only the dntouch devices are attached to screens by Xorg, and the original "raw" devices are detached from the input queue in order to avoid conflicting mouse/touch protocol data (version 2.3.2: device disabling is not needed anymore in Wayland).

Calibration Tool

To allow calibration, the dntouch driver exposes a calibration interface through a socket for each device discovered located in `/var/run/dntouch/dncalibX`.
Access rights to the calibration interface can be controlled by setting the desired access rights to these special files.

The calibration tool dncalibrate will use these sockets to communicate with the driver and set new calibration data. Once opened the dntouch driver will send all incoming events to the calibration socket unfiltered instead of sending them to the output event device. dncalibrate will use the data to calculate a new calibration and will send this new data to the dntouch driver using the following commands:

```
> CALIBRATE <min_x> <max_x> <min_y> <max_y>
> SCREEN_DATA <x> <width> <y> <height>
> DESKTOP_DATA <x> <width> < y> <height>
```

Once new calibration data is sent, the dntouch driver will generate output events again using this

calibration data. To persist the new calibration data, the dncalibrate tool needs to send the following command:

```
> SAVE
```

If the connection to the dntouch driver is disconnected before sending the save command, the original calibration is restored by the dntouch driver. The dntouch driver will store the calibration data as text file in

`/etc/opt/dn/dntouch/calibrationX`, containing the min_x, max_x, min_y and max_y values on individual lines.

The calibration tool can be started as follows:

```
>> /opt/dn/dntouch/bin/dncalibrate --screen=0
>> /opt/dn/dntouch/bin/dncalibrate --screen=1
```

The calibration tool has a draw mode to verify the calibration. This can be started with the following Parameter:

```
>> /opt/dn/dntouch/bin/dncalibrate --test
```

This mode can also be used with the `--screen` parameter.

Additional Configuration Files

As each login display manager uses its own startup script before the user logs in, and also a user's profile after login may change screen geometry and resolution, additional scripts take care of calling dnmapping again after such events. They are automatically installed by the dntouch-udev package.

`/etc/gdm/Init/Default /etc/X11/xdm/Xsetup /etc/kde4/kdm/Xsetup /etc/kde5/kdm/Xsetup`: Configuration files for display managers gdm, xdm, kde4 or kde5

`/etc/X11/xinit/xinitrc.d/zz-dntouch.sh`: Executed during session startup after login.

`/etc/xdg/autostart/zz-dntouch.desktop`: Executed after other user-defined autostartup-scripts.

If **no window manager** if used, the startup script might not be called. In this case, the mapping is not applied. Add a call of `/etc/X11/xinit/xinitrc.d/zz-dntouch.sh` to the default start script e.g. `~/.xinitrc`

Driver Version History

- 2.3.2 – Initial release on Rocky Linux 8 with Wayland instead of Xorg
- 2.3.3 – Support for D2xxx Displays, Mapping Tool supports Xorg

Diebold Nixdorf Systems GmbH
D-33094 Paderborn