

T3ED Help Index

The main developers:

- Lasse Kärkkäinen alias Nappel - 2nd of November 2004, Kotka / Finland
- Dev 3.2 and higher made by JimDiabolo aka GER JimD

This is Version 0.07pre1.1 Dev 4.4 of T3ED.

T3ED was originally written by Denis Auroux, 1998-99 (e-mail: auroux@math.polytechnique.fr).

Special thanks to Vitaly Kootin for some NFSHS file format information.

How To...

- [What's new in this version?](#)
- [Downloading the latest version of T3ED](#)
- [Frequently asked questions](#)
- [NFS3/HS Tracks Overview](#)
- [NFS3 Track File Formats](#)
- [NFS HS Track File Formats](#)
- [Modifying the document](#)
- [Adding track polygons](#)
- [Quickly fine-tuning opponent AI line float & speed values](#)
- [INI File Tutorial](#)
- [Selecting elements](#)

Commands

- [File menu](#)
- [Edit menu](#)
- [View menu](#)
- [Edit-Modes menu](#)
- [Edit-Tools menu](#)
- [Whole Track menu](#)
- [Help menu](#)
- [Toolbar](#)
- [Scroll bars](#)
- [Status bar](#)

What's new in this version??

New in Version 0.07pre1.1 Plus Dev 4.4 :

- Zoom In/Out now works when Track View windows is active.
- Zoom In/Out Track View only (using ALT Key) is now much faster.
- Additional safety check when starting Track View.
- Light properties has now a Combo box for easy selection of track glows (values taken from Tr(N).ini) or special effects.
- Light properties has a Raytrace Light button. Maximum Light Distance will be intensity * 0.1.
- Light/Sound properties will be shown as four bytes. Byte 2 must be zero, to be visible in game. If not, only Raytracer will use it.
- Delete Light/Sound object from block properties.
- Shadow Raytracer can now calculate lights:
 - A imaginary line from each light to vertice points will be calculated. Only when nothing is in the way the light shines there.
 - Light color and intensity will be taken from tr(N).ini
 - Multiplier of light distance can be set. Light Distance will be intensity * multiplier.
 - If red, green or blue of light is higher than the color part of the point it will be replaced.
 - Flashing Lights will be ignored.
 - Byte 2,3,4 of Light will be ignored.
- Light Raytracing for lanes added.
- Option to show PolyVRoad for (black) trackpolygons. It should always point forward. But now I can be changed with "Polygon Virtual Road Flags".
- Bugfix when replacing blocks and using undo.
- Multiple polygons can now be duplicated instead of only one.
- If showing "Virtual Road Boundary Polygons"
 - Polygons with "No ?? Neighbour" flag get line from middle pointing to the corresponding direction.
 - Polygons with "Extra Object Detection" flag get an "E".
- "Polygon Virtual Road Flags" You can now choose whether "Drive over behavior" should be set or remain as original.
- "Shadow ray tracer" the maximum end block has been corrected from 255 to 299.
- "Find polygon" Now you can specify how many percent of the hits should be selected.
- No delete for VRoad Points, T3ED would crash.
- "Texture" function disabled for special objects, T3ED would crash.
- "Virtual Road, heights, spdfa & spdra" now shows the VRoad information about shortcuts. Each VRoad point contains two possible numbers. If there is none both are -1.
- Fix for "Point properties" when only changing the coordinate.
- Updated "Virtual Road, heights, spdfa, spdra" with "HS_Extra" data.

- Updated Export / Import "Virtual Road, heights, spd, spd" with "HS_Extra" data.
- Improved "Adjust virtual road" with "Automatic lane detection".
- Virtual Road & AI Points not normal objects anymore. You have enable "Virtual Road Points" Edit-Mode to change them.
- In "NFSHS Texture Properties" you can enable the "Scroll effect". Scroll direction is set by "Rotation"
- "New/Duplicate" (Ctrl + N) usually splits track polygons to create a new. If it's the first or last of a row, it now adds one to the side. If you want the split function hold the "Alt" key.
- Holding ALT-key when pasting objects keeps the original position.
- Big changes to selecting function (LButtonDown): It is now possible to select a row of polygons or points. Select the first, then hold "Shift" and select the last. All on a line between them will be selected.
- New function: Edit-Tools > Add Lane.
- Interpolation of float values between two AI Points (using cosine) Select first (holding ALT) then second (CTRL + ALT) and start calculation with SHIFT + Q. By Default curve calculation is done, use ALT + SHIFT + Q for straight Interpolation.
- With "Clear / Modify All..." all float values of the speed files can be set to zero.
- With "Clear / Modify All...", float values that are zero can be recalculated (as straight or curve). An interpolation is made between the first and the last value that is not zero.
- Improved undo function:
 - Increased possible undos from 16 to 64.
 - Undo possible if deleting and adding blocks.
 - Undo points won't be deleted when saving track.
- Deleting all selected Polygons is now possible (not only the last).
- Deleting all selected Objects works, too.
- Animated objects data accepts neative numbers.
- TrackView starts with a bigger window.
- Show Block Direction added.
- Show / Hide track (black) Polygons added.
- Recalculate speed or position between two AI Points:
 - Enable "Edit Mode - Virtual Road Points".
 - Select Speed File for Viewing (Forward / Backward).
 - Select first AI Point with ALT pressed.
 - Select second AI Point with CTRL & ALT pressed.
 - Recalculate speed between two AI Points with SHIFT-S. If you want to recalculate the position (float values) between two selected AI points use SHIFT-Q.

New in Version 0.07pre1.1 Plus Dev 4.3 :

- Paste block to current vroad added.
- If you import a Vr_H_Spd.csv file with changed virtual road values, the track can be rearranged to that new virtual road. This can be used to change track shapes.
- Bug fix: Poly VR Flags will only be displayed for (black) track polygons. (Alt + enter)
- Bug fix for Pasting HS extra data (Number of lanes)

- New option: Set visibility minimum.
- New function: Copy & paste virtual road points. You can select what will be pasted. Position & vectors, Heights data, spdFa data, spdRa data or HS extra Data (Number of lanes)
- When converting (Global, extra & polygon) objects you won't lose shadow/color information anymore.
- Integrated Replay Camera Editor:
 - "Tr.cam" will be loaded / saved with the frd-file.
 - Enable viewing cameras by "View -> Show Replay Cameras"
 - When a camera is selected a light blue line from camera start to stop will be display along the virtual road.
 - Move, copy, and delete them like other objects.
 - When you press "Alt" while moving a camera T3ED will adjust start, stop and render.

New in Version 0.07pre1.1 Plus Dev 4.2 :

- Visibility:
 - When exporting visibility, the file will be saved in the track directory.
 - With "Show track based on visibility" you can now use the right mouse button to set visibility edges of the selected block. Control + right mouse toggles the block.
 - Better handling of visibility when you delete / insert a block.
- NFS3/4
 - Bugfix for deleting the last block of a track with less then 8 virtual road points.
 - Ability the change all texture of a block with its objects in one step.
 - Show more info about what is selected in the status bar. I did this for some testing purpose, I think it will stay.
 - When moving virtual road points manually, the „ColVroadVecs“? of track polygons will be recalculated.
 - You can change all texture of a block or an object in one step. (Edit ? Texture when in block or object mode.)
 - "File ? Changes all textures" to edit all texture numbers of the track,
 - Export a block to an off – file.
 - Paste a block as polygon (blue) object.
 - Rewritten for "Adjust road width" function.
 - Added an "Adjust road width" button to block properties.
- For NFS3:
 - T3ED loads now the nfs3 heights file "simhts.dat" and the speed files "speedsf.bin/speedsr.bin".
 - Adjust the "col.vroadHead.size" to allow paste/delete blocks on nfs3 tracks!
 - Changes to the texture dialog on nfs3 tracks:
 - It shows the texture id and the bitmap number in the qfs – file.
 - NFS3 Texture Properties dialog was added. It allows to change two sided & animated textures.
 - Import/Export the NFS3 textureblock.
- HS Only:

- Global object chunk 1 will be loaded and saved correctly. (Cross-) type 6 object are now visible.
- Type 6 objects can be deleted, duplicated and moved as global objects. To edit them, convert to an extra object, make the changes, and convert back to global.
- Since I don't understand all type6 data, the unknown 60 bytes can be exported & imported. I think it stores the weight, behavior and size.
- Shadows:
 - Clear / Set shadows of extra objects manually. You can also change the color and transparency of the objects texture there!
 - Point Properties with shading values.
 - Many changes to Nappe's ray tracer:
 - Extra object can get shadows.
 - Set sun and shadow color.
 - You can now select what object get new shadows
 - Select object that cast shadows
 - If you set transparency, these points can be ignored from ray tracer

New in Version 0.07pre1.1 Plus Dev 4.1 :

- Html help file (chm file) ist now used instead of old hlp format. Windows 10 doesn't support .hlp files anymore.
- "visibility.txt" files will now be stored in the track directory.
- Fix for deleting the last block.

New in Version 0.07pre1.1 Plus Dev 4.0 :

- A new paste mode has been added. "Replace only track texture information". It will replace all textures and properties from track polygones. Objects won,t be changed.
- A new view has been added "Show Track Based On Visibility". Invisible blocks are show in a light grey, and can,t be selected.
- Fix for crashing when selecting global objects and active trackpreview.
- Default trackpreview shows now two more blocks.
- If "Show Track Based On Visibility" is activated, trackpreview shows blocks that should be visible in game, up to 30 blocks. Global objects will always be shown in trackpreview.
- Visibility information can be changed for actual block by using [Alt] Page Up/Down keys. "Show Track Based On Visibility" must be enabled.
- Size of trackpreview window can be easier changed now.
- Use CTRL + Mouse Wheel to zoom in/out trackpreview without changing normal view scale.

New in Version 0.07pre1.1 Plus Dev 3.9 :

- Global NFS4 object manipulation added. Global object will now be shown, can be moved and converted in extra objects and back.
- It is now possible to delete any block from the track. Not only at the end.

- Paste block from Clipboard before selected block added. (Be careful, not very well tested !)

New in Version 0.07pre1.1 Plus Dev 3.8 :

- Frd file properties dialog added.
- Block properties dialog added.
- New function "Edit -> Find" for polygons added. You can search for polygons by texture, HS texture properties, texture animation data and polygon road flags (drive-over behavior).
- VRoad repoints will be handled as extra objects. Select "Edit Virtual Road Points" and you can move them like you want.
- For NFS4 only :
 - If available height.sim, spdfa.bin and spdra.bin will be loaded and saved with the tr.frd file. (Thanks to Christian Brandt & Lasse Karkkainen for their NFS4 Spdtools.)
 - Heights.sim, spdfa.bin & spdra.bin data will be copied and pasted together with block data.
 - Undo for Heights.sim, spdfa.bin & spdra.bin data added.
 - New dialog "Edit Tools > Virtual Road, heights, spdfa & spdra" (Ctrl+H) allows to edit that values. In block mode all data from actual block is shown. In object mode all data from selected VRoad repoints will be displayed if properties is selected from menu. Note : The "spdXA Speed" has only 4 values per block all others have usually 8. That's why speed values are shown twice, e.g. 0=1,2=3,4=5 and 6=7. Only the first (0,2,4,6) will be written to disk.

New in Version 0.07pre1.1 Plus Dev 3.7 :

- Textures of new created objects won't be animated by default anymore. This caused several HS crashes.
- Import (extra) objects from '.off' files. Object File Format contains no texture information, only structures can be imported. Because NFS only supports polygons with 4 vertices other will be transformed into 4 poly ones. Polygons with more than 8 vertices can't be imported.
- Export (extra) objects to '.off' files.
- The texture function for objects can now set NFS HS Texture properties.
- New "merge objects" function. Go to object mode, enable the merge/split button. When you select two (extra) objects the second will be merged into the first.
- Polygon & extra objects can be moved into another chunk. Turn off "Automatic Object Membership" and open properties. (ALT + Enter).
- Rotate for polygons (NFS3 compatible, rotates the texture without using HS Texture Properties.)
- Invert polygons function (CTRL + I), turns over front & back.
- Much more extra object properties are shown and can be changed. For example, the crosstype can be changed. That allows to make objects hitable. The collide effect specifies how they behave. Solid like trees/walls (1) or like signs (2).
- In the extra object properties dialog, the animation data can be exported & imported to a text file. And finally the length of the animation can be changed.

Bigger ones that the simple rotating donut on snowy are possible.

- New object can be created as animated objects.
- Show objects middle in status bar.

New in Version 0.07pre1.1 Plus Dev 3.6 :

- "Undo, but keep VRoad" added. The VRoad can be moved in block mode, when the track is moved. Now there a undo which brings back the track, but keeps the moved VRoad.
- Adjust road width fixed. Now you really can select the blocks which should be adjusted.
- Copy whole block to Clipboard
- Paste block as new additional block.
- Replace the currently selected block, with block from clipboard. Move and, or rotate the new block in old position.
- Moving blocks without smoothing by pressing control.
- Rotate whole blocks.
- The "Expand / Shrink", "Rotate", "Shadow Raytracer", "Add / Replace Block" & "Adjust Road Width" windows are keeping their settings for the next call.
- Polygon / extra object properties added (ALT+ENTER). If "Automatic Object Membership" is off, you can manually change the block where the object belongs to.
- Move distance can be manually entered.
- Show the refpoint position in status bar when block mode is active.
- The Block Neighbours function has now an option to connect points to near points of neighbour blocks.
- Line up x,y, (z) function for point mode added.

New in Version 0.07pre1.1 Plus Dev 3.5 :

- Copy Polygon-, Extra-, Light& Soundobjects to clipboard. You have to be in object mode and use Ctrl + C or "Edit -> Copy" from menu.
- Paste Polygon-, Extra-, Light& Soundobjects from clipboard to another block or even track ! Select object mode and use Ctrl + v or "Edit -> Paste" from menu.

New in Version 0.07pre1.1 Plus Dev 3.4 :

- Rotate (extra-) objects around their middle.
- New / Duplicate function has a new shortcut Ctrl + D or Ctrl + N. (Ctrl + V will be used for clipboard.).
- Copy position of a point to Windows clipboard (Ctrl + C).
- Paste clipboard postion to a point (Ctrl + V)

New in Version 0.07pre1.1 Plus Dev 3.3 :

- Change textures of trackpolygons, polygon & extra objects in one step. Choose the new lowest texture and all textures will be change with the same offset.
- Expand / Shrink (extra) objects is now possible. You can select each axis independent or connected. It,s selectable on which position the object will be

aligned. (Top, middle or bottom.)

- Fixed light & sound source properties. Values can be changed directly in T3ED now, for those who don't want to use the frd-tool.
- In point mode : Ctrl + Y copy X & Y position to memory, Shift + Y paste memory position to selected point(s).

New in Version 0.07pre1.1 Plus Dev 3.2 :

- Track shading tracer, by Lasse Kärkkäinen aka. Nappe1. It's one of the last thing he implemented before he gave up working on T3ED. All credits should go to him.
- Mousewheel can be used to zoom in / out.
- Compiled DirectX sdk sum2004 which is the latest working with VS6.
- Added options for LOD Generator and Track shading tracer. (Start Block, End Block, etc.)
- After moving objects, they will be stored in the Block which is closest to the new position.

Added in Version 0.07pre1.1 Plus Dev 3 :

- Copy / Paste for Z-coordinate (height) values of selected vertice
- Z-coordinate (height) gradient calculation on multiselected vertices.
- Z-coordinate average on multiselected vertices. (calculates average height of all selected vertices and sets all of them in that height.)
- new, more readable format on import / export visibility lists.
- Sound Source editing (move, copy, delete) works flawlessly at last
- animated textures can be now edited.
- new ways to view track on editing screen (highlight selected block, draw quads on dark green lines that have virtual road boundary flag set, Virtual Road Bitmap mode.)
- new algorithm generating virtual road data. (as option. old Denis' algorithm is still available too.)
- sound/light source data editing. (I have no idea how this works, but I can give a possibility to designer manually type those two bytes.)
- Generate LODS : LOW and MEDIUM Detail Mesh generation with 100% compliant NFS4 track file format vertex list sorting.
- 16 and 32 bit texture support on texture selector as well as in track view.
- Better control over Drive Over Behaviour flag set up. (everything now manually.)
- Additional runtime (not saved in track.) polygon flag for different purposes.
- Virtual Road Bitmap is automatically corrected on track load, as well as if polygon flags are changed.

Added in Version 0.07pre1.1: (experimental)

- Multiple selection in point, polygon and object modes: Press Control or Shift while clicking in order to add to the current selection.
- Multiple selection support for the Move XY and Move Z tools. Note that you must keep the Control or Shift key pressed when you click to move the selected items, otherwise the multiple selection is lost.

- Multiple selection support for the Polygon flags and Textures dialog boxes in polygon mode.
- Light sources are handled as objects: in object mode they can be selected, moved, duplicated and deleted.
- Texture Properties dialog box accessible from Textures dialog box (in NFSHS mode only).

Added in Version 0.06:

- Erase all function.
- New handling of track polygons; should fix a lot of bugs; adding new track sections should now work correctly (within the game engine's limitations); see the specific [help page](#) on this feature.
- Additions to the NFS HS file format specifications by Vitaly Kootin.

Added in Version 0.05:

- Support of NFS High Stakes track files (special thanks to Vitaly Kootin for some info about file formats)

Added in Version 0.04:

- Adjust road width command (to remove the uncrossable walls on either side of the track).
- Light sources are moved (when using Move tools in block or extended point modes).
- File format specs additions (in part by Vitaly Kootin).

Added in Version 0.03:

- [Track file format specifications](#).
- Delete command for polygons.
- Duplicate command for polygons (multi-resolution) (see the [FAQ](#) about adding new road sections).
- Support of road lane polygon textures.
- Polygon virtual road flags dialog box: allows control of object and wall collisions (see the [FAQ](#) for more details).
- Merge tool in Point mode.
- Split tool in Point mode.
- Completed help file.

Added in Version 0.02:

- Smoothing properties dialog box for Block mode and Extended Point mode.
- Extra smoothing option for the Move XY and Move Z tools in Block mode.
- Move XY and Move Z tools also work in Polygon mode.
- Delete command for objects.
- Duplicate command for objects.
- Creation of new objects (using the New/Duplicate command with no selected object).
- Texture browser and polygon texture selection.

- Help file (still incomplete).

Features still missing:

- Optional modeless texture display palette in Polygon mode
- Rotating tool for object mode.
- Improved handling of objects when track blocks are moved.
- Properties dialog boxes (all modes).
- Tool for banking the road.

Extra suggestions:

- object action after crash (road signs and some fences lie down when touched)
- "magnet" function to keep objects on the ground
- replay camera positions
- modifying texture affectation table
- opponent car AI data (optimal speed, etc...)
- conversion between NFS3 and NFSHS formats
- background bitmap

Downloading the latest version of T3ED

This is Version 0.07pre1.1 Plus Dev 4.4.0 of T3ED, first released on 2023/2/19

Downloads should be available at <https://www.nfsaddons.com>.

The last stable release is Version 0.06.

T3ED is Copyright (c) Denis Auroux, 1998-99 (e-mail : auroux@math.polytechnique.fr).

The latest version of T3ED is available from the official T3ED site at <http://www.math.polytechnique.fr/cmat/auroux/nfs/>

A mirror is available at <http://auroux.free.fr/nfs/>

Frequently Asked Questions

1. [General questions](#)
2. [Editing tips](#)
3. [Troubleshooting](#)

1. General questions

When does the final version of T3ED come out?

It all depends on how much time I can spare for working on T3ED. No further release is planned before October 1, 1999.

Does T3ED require any specific DLLs in order to work?

Some Microsoft DLLs are required by T3ED : notably MFC42.DLL and MSVCRT.DLL. These files are very widespread and it is extremely likely that they are already installed on your system. If not, they are quite easy to find as they are distributed with many recent programs, and they might even be available on Microsoft's web site.

Where do I find the track files I'm supposed to edit in T3ED?

The track files are located in the GAMEDATA\TRACKS\TRK00x subdirectories of your NFS3 installation directory. If there is no GAMEDATA\TRACKS subdirectory in your NFS3 directory, you must reinstall NFS3 in order to have the tracks loaded from the hard disk rather than from the CD-ROM.

In NFS High Stakes, track files are similarly located in the subdirectories of the DATA\TRACKS subdirectory of your NFS HS installation directory.

Can I have both edited tracks and the original tracks at the same time in NFS?

No. Unlike cars, tracks cannot be added to NFS, so any edited track has to replace an original track.

How can I revert back to the original tracks?

Simply copy the .FRD and .COL files from the GAMEDATA\TRACKS\TRK00x subdirectory of your NFS3 CD-ROM back into the corresponding subdirectory of your hard disk NFS3 installation directory. In order to be able to modify again these files with T3ED, you will need to remove the Read-only attribute set by Windows when they're copied from the CD-ROM.

In NFS High Stakes, proceed similarly with the DATA\TRACKS\... subdirectories; there is no .COL file, so only the .FRD file needs to be replaced.

How can I distribute a modified track?

Just create a ZIP file containing both the .FRD and .COL files. People will be able to install the modified tracks simply by unzipping that file in the correct directory. In NFS HS there is no .COL file, so you just need to distribute the .FRD file.

Is it possible to convert NFS2 tracks to NFS3?

This is an extremely complicated task which requires the development of specific tools. Vitaly Kootin (vitaly_necromancer@yahoo.com) is working on this.

Is it possible to convert NFS3 tracks to NFS HS and vice-versa?

Although this is much easier than conversion between NFS2 and NFS3, this feature is not yet offered in T3ED because of the large number of remaining bugs in the conversion code. It might become available in a later release.

Can I create a new track from scratch?

This is not possible with T3ED, since it is not possible to modify the table of track textures or the track length. However you can use the "Erase All" function in the Edit-Tools menu to remove nearly everything from an existing track, which allows you to start again almost from scratch.

2. Editing tips

How can I make an oval (or a high-speed ring, or a straightened track)?

First select Block mode in the Edit-Modes menu, and use the Smoothing properties dialog box (Alt-S) in order to activate the Extra-Smoothing option and to select a large value for the width of the smoothing zone (500 ought to be fine). Select the Move XY tool. By clicking on a track block and slightly dragging the mouse, you can remove all turns over a huge track section at once. Repeat this process two or three times on various track blocks in order to smoothen all the track. Next, select the Move Z tool, and repeat the process to remove the slopes.

In v0.06 there is a new easier way to make a circle track: simply use the "Erase All" function in the Edit-Tools menu to remove the track shape and road elevations.

Is it possible to modify textures or create new textures?

T3ED is not a texture editor. You need to download a texture converter such as the QFS Editing Suite (available at <http://www.math.polytechnique.fr/cmat/auroux/nfs/>), which will let you convert QFS files into plain Windows bitmaps that you can edit using your favorite software.

I moved an extra-object, and the cars pass through it as if didn't exist...

Collisions with extra-objects (displayed in green) are only detected over specific track polygons. In order to allow a moved object to be detected, you must modify the track polygon properties accordingly. First choose the High-resolution view, and switch to Polygon mode. Select the track polygon over which the object lies; use the Polygon Flags command in the Edit-Tools menu (or press Control+Enter), and check the "Extra-object collision detection" option. If the object lies across several polygons, repeat this operation for every polygon to enable object detection.

I moved a polygon object, and I get an invisible wall where it used to be...

Collisions with polygon objects are not handled by NFS; rather, the track polygons over which the object lies (and the ones beyond in the case of a fence or a wall) are marked as impassable for the cars. The procedure to change the polygon properties (passable or not, etc...) is the following: first chose the High-resolution view, and switch to Polygon mode. Select the track polygon, and use the Polygon Flags command in the Edit-Tools menu (or press Control+Enter); choose the relevant option in the "Drive-over behavior" list to change the passable status of a polygon. Also note that collisions with impassable polygons are handled only if all neighboring polygons have the "Wall collision detection" flag set; normally this flag is automatically set by T3ED when a polygon is made impassable. Also use the Adjust Road Width command in the Edit-Tools menu to adjust the virtual road width.

Is it possible to add extra road sections??

It is not possible to extend the number of track blocks in the track. However it is now possible to add road polygons to a track block. This is an advanced feature, not intended for beginner users. See the specific [help page](#) on this feature.

When I race in NFS, my car is stopped by an invisible wall even after making all polygons passable.

In addition to the passable/impassable flag for each track polygon, NFS also puts invisible walls on both sides of the track in order to delimitate the driveable area. A track polygon, even if marked passable, can be driven over only if it lies between the two boundaries. To adjust the track width, use the Adjust Road Width command in the Edit-Tools menu. Also see the page on See the specific [adding track polygons](#) page for more tips.

3. Troubleshooting

I get an error message when attempting to save a modified track. What is happening?

T3ED is not able to create the modified track files. First of all, check that you are not trying to save your track on the CD-ROM drive. Also, both the .FRD and .COL files must have the Read-only flag cleared; be especially careful with this if you have copied the files back from their CD-ROM version. The Read-only status of a file can be modified in its properties box, obtained by right-clicking on the file.

In some instances, the track files of NFS HS come with the Read-only flag set: you need to clear this flag before you can overwrite the tracks.

The textures displayed in the texture browser look very bad. What can I do?

Check that your Windows display is configured with a sufficient color depth. High-color or true-color modes (at least 16-bit depth, i.e. 65536 colors) are required in order for the texture browser to display textures correctly. In 256-color mode you will get garbled textures, even if you try working with the 8-bit textures.

I can't view the textures because the QFS file has the wrong format. What is happening?

Some of the QFS texture files have an abnormal compression format, and cannot be decoded by T3ED. This is in particular the case with the 8-bit mode textures for the Summit track. When this happens, you should try loading a different texture file. In particular the 16-bit mode texture files take longer to load, but do not suffer from this problem.

Why can't I use some of the tools?

Since this is still a preliminary version of T3ED, some features are not yet implemented. Also, most commands are only available in specific editing modes (e.g. texture browsing is only possible in polygon mode).

When I race in NFS, there are lots of display bugs. Why?

First of all you should be aware that display bugs are partly unavoidable, because the track file formats are not yet fully understood and some of the necessary information is not known. However, the following general tips can be helpful:

- Never create sharp bends in the track which would cause some of the track polygons to overlap with each other. If you accidentally create such a sharp turn, you should use the Undo command to restore a correct track structure, and modify the Smoothing properties in order to increase the amount of smoothing.
- Use block mode whenever possible rather than the extended point mode. Block mode is the only mode in which all the relevant virtual road information gets

- updated when you move things around.
- Avoid creating slopes that are too steep, or things will tend to disappear in front of you when you race the modified track.

When I race in NFS, my car is stopped by an invisible wall even after making all polygons passable.

See above, in *Editing tips* section.

It doesn't work, what can I do?

You can send me a precise description of your problem by e-mail (auroux@math.polytechnique.fr). Due to my limited available time, I will only help people who state clearly their problem and whose questions are not answered in this help file. If you get an error message, please state its exact contents. If you get suspicious behaviour, please indicate exactly what actions you performed and what happened.

NFS3/HS Tracks Overview

This help page describes the basic concepts involved in the structure of a NFS3/HS track. For technical information on the NFS3/HS file formats, please refer to the pages on [NFS3 Track File Formats](#) and [NFSHS Track File Formats](#).

Track structure

NFS3/HS tracks are subdivided into track blocks (ca. 200-300 of them), each of which consists of track vertices, polygons, objects and virtual road data. The corresponding data are mostly stored in a file with extension .FRD, and for NFS3, with a companion file with .COL extension.

Vertices

These are the points which make up the track. They can belong either to the main track structure, or to a "polygon/extra/global object" (see below).

Polygons

A track block consists of three series of polygons, corresponding to various resolutions (the low and medium resolutions are only used to save computing time when the car is still far away). To every polygon corresponds a texture, and the high-resolution series of polygons is also associated with virtual road information.

Objects

There are several types of objects:

- **Polygon objects** (displayed in blue in T3ED) make use of the same list of vertices as the main track structure. Polygon objects are fixed and cannot be collided with.
- **Extra objects** (displayed in green in T3ED) have separate vertex information structures compared to polygon objects, extra objects feature collision detection and can be animated as well.
- **Global objects** (displayed in teal in T3ED) are similar to extra objects, but they are not bound to a single track block and can be seen from any track block. In NFSHS they can also be used as track props with mass and a hitbox (what is dubbed as "complex behavior")
- **Light and sound sources:** objects used to place light glows and ambient sounds, respectively.

Virtual Road

The virtual road is all the information used by NFS3/HS in order to compute the behavior of the various cars (player, police, opponents or traffic). This includes, in particular, a sequence of "virtual nodes" stored either in the associated .COL file (for NFS3) or in the .FRD itself (for NFSHS); virtual forward and normal vectors associated to every high resolution track polygon; etc...

Textures

The track textures are stored in a companion file with .QFS extension.

- **For NFS3:** four files are provided for each track: the files whose names end with a '0' correspond to normal racing, while those which contain a '1' correspond to mirrored tracks. The files with a '_8' suffix in their name correspond to 256-color textures, while the others correspond to 65536-color textures.
- **For NFSHS:** there are either one (tr0.qfs) or two (tr0.qfs and trn0.qfs) texture files. tr0 is the texture file for the daytime track while trn0 is the texture file for the nighttime track.

NFS3 Track File Formats - V0.04, (c) Denis Auroux 1998- 99

This help page describes the structure of NFS3 track files. It is intended as reference information for other NFS3 developers; if you don't intend to program your own track editor, then the [NFS3 Tracks Overview](#) page is probably more appropriate.

Some of the information in this document was contributed by Vitaly Kootin.

DISCLAIMER: The information below is unofficial, and may be inexact by some aspects. It is only intended as reference information to help other NFS3 developers, and comes with no warranty. It may not be used for commercial purposes. It is forbidden to distribute a modified version of this document.

Contents:

- [Basic structures](#)
- [Track blocks](#)
- [Polygon blocks](#)
- [Extra-object blocks](#)
- [FRD files](#)
- [COL files](#)

I. Basic structures

1. FLOATPT structure (size: 12 bytes)

```
typedef struct FLOATPT {  
    float x,z,y;  
} FLOATPT;
```

This structure is used to store the three coordinates of a vertex. Each of the components is stored as a floating-point number (IEEE single-precision format). The x and y axes are horizontal, while the z axis is vertical and points upwards.

2. INTPT structure (size: 12 bytes)

```
typedef struct INTPT {  
    long x,z,y;  
} INTPT;
```

This is another way to store the three coordinates of a vertex (less frequently used than the FLOATPT structure). Each of the components is stored as a 32-bit integer, and must be divided by 2^{16} in order to obtain the corresponding floating-point value (in other words, the coordinates are 16.16 fixed precision numbers).

3. POLYGONDATA structure (size: 14 bytes)

```
typedef struct POLYGONDATA {  
    short vertex[4];  
    short texture;  
    short unknown1;  
    unsigned char flags;  
    unsigned char unknown2;  
} POLYGONDATA;
```

This structure describes a polygon. The `vertex` member refers to four entries in the vertex table associated with the polygon (either the vertex table of the TRKBLOCK structure in the case of a POLYBLOCK polygon, or that of the XOBJDATA structure in the case of an XOBJDATA polygon), describing the coordinates of the four polygon vertices. The `texture` member points to one of TEXTUREBLOCK structures in the FRD file and describes the polygon texture. The significance of the `unknown1` and `unknown2` members apparently has to do with animated textures (resp. first texture number and animation speed??) or with the special road lane textures. A non-zero value of `flags&0x04` indicates that the polygon has an animated texture, and a non-zero value of `flags&0x10` indicates that the polygon is two-sided.

II. Track blocks

1. TRKBLOCK structure (variable size)

```
typedef struct TRKBLOCK
{
    struct FLOATPT ptCentre;
    struct FLOATPT ptBounding[4];
    long nVertices;
    long nHiResVert,nLoResVert,nMedResVert;
    long nVerticesDup;
    long nObjectVert;
    struct FLOATPT vertices[nVertices];
    long shadingVertices[nVertices];
    struct NEIGHBORDATA nbdData[0x12C];
    long nStartPosition;
    long nPositions;
    long nPolygons,nVroad,nXobj,nPolyobj;
    long nSoundsrc,nLightsrc;
    struct POSITIONDATA posData[nPositions];
    struct POLYVROADDATA polyData[nPolygons];
    struct VROADDATA vroadData[nVroad];
    struct REF XOBJ xobj[nXobj];
    struct REFPOLYOBJ polyobj[nPolyobj];
    char padding[];
    struct SOUNDSRC soundsrc[nSoundsrc];
    struct LIGHTSRC lightsrc[nLightsrc];
} TRKBLOCK;
```

This structure describes a track block; it contains the track vertices, virtual road information, and miscellaneous reference tables, but does not include the track polygons nor the objects. The `ptCentre` member provides the coordinates of a central reference point around which the block is built. The `ptBounding` member provides the coordinates of four bounding points for the block vertices: these points delimit a horizontal rectangle inside which all the block's elements fit.

The `nVertices` and `nVerticesDup` values are always equal, and give the total number of vertices stored in the `vertices` member of the `TRKBLOCK` structure. Vertices 0 to `nObjectVert-1` are the vertices used by the polygon objects (see `OBJPOLYBLOCK` structure). Vertices `nObjectVert` to `nLoResVert-1` are used in chunks 0 and 1 of the `POLYBLOCK` structure, and correspond to the track vertices in low resolution. Vertices `nLoResVert` to `nMedResVert-1` are the additional vertices used in chunks 2 and 3 of the `POLYBLOCK` structure, and correspond to the track vertices in medium resolution (the low-resolution vertices are also used in medium resolution). Vertices `nMedResVert` to `nHiResVert-1` are the additional vertices used in chunks 4 and 5 of the `POLYBLOCK` structure, and correspond to the track vertices in high resolution (the low- and medium-resolution vertices are also used in high resolution). Finally, vertices `nHiResVert` to `nVertices-1` are used in chunk 6 of the `POLYBLOCK` structure, and correspond to the road lanes; after each road lane, a hole is usually left with four

unreferenced vertices. The `shadingVertices` member describes the shading applied to each vertex: for each vertex, the least significant byte is the blue component, the next byte is the green component, the next byte is the red component, and finally the most significant byte is the alpha value.

The `nStartPosition` member indicates where the block is located along the track, in units of "nodes" (see the `COLVROAD` structure in `COL` file). Normally each block corresponds to 8 node positions, except the last one: the number of nodes in the block is given by the `nPositions` member. The `nPolygons` member indicates the total number of high-resolution track polygons (i.e. the same as the size of chunk 4 of the `POLYBLOCK` structure). The `nXobj` member indicates the total number of non-animated (type 4) extra-objects stored in the four extra-object chunks associated with the track block. The `nPolyobj` member indicates the total number of objects (both polygon-objects and extra-objects) in object chunk 0 (this normally corresponds to length of `OBJPOLYBLOCK` chunk 0 in the `POLYBLOCK` structure, except if there are only extra-objects).

In addition, the `nPositions`, `nPolygons`, `nVroad`, `nXobj`, `nPolyobj`, `nSoundsrc` and `nLightsrc` members describe the number of entries in the various corresponding substructures (see below). As the `REFPOLYOBJ` structure exists in two variants (16-byte or 20-byte long depending on object type), extra padding is added in the `.FRD` file just after the `polyobj` substructure, in order to obtain a total size of $20 * nPolyobj$ in all cases for the `REFPOLYOBJ` table. Apart from this there is no padding anywhere.

2. NEIGHBORDATA structure (size: 4 bytes)

```
typedef struct NEIGHBORDATA {
    short block;
    short unused;
} NEIGHBORDATA;
```

This structure is used to list the block numbers corresponding to the neighbors of the currently described block, starting with the closest ones. There are always 0x12C such entries in the `TRKBLOCK` structure, but for obvious reasons most of these entries are unused and just contain -1 in the `block` member.

3. POSITIONDATA structure (size: 8 bytes)

```
typedef struct POSITIONDATA {
    short polygon;
    unsigned char nPolygons;
    char unknown;
    short extraNeighbor[2];
} POSITIONDATA;
```

This structure is used to indicate which track polygons are attached to a given node. The `polygon` member refers to an entry in the `POLYVROADDATA` structure (or in chunk 4 of the `POLYBLOCK` structure); all entries between this one and the one pointed to in the following `POSITIONDATA` structure correspond to the high-resolution track polygons attached to the same node. The `nPolygons` member indicates the number of

polygons attached to this node. The two `extraNeighbor` values are usually equal to -1, but can be used to specify up to two neighbouring nodes other than the previous node and the following node (e.g. if there are shortcuts).

4. POLYVROADDATA structure (size: 8 bytes)

```
typedef struct POLYVROADDATA {  
    unsigned char vroadEntry;  
    unsigned char flags;  
    unsigned char unknown[6];  
} POLYVROADDATA;
```

This structure describes the virtual road properties of a track polygon: each entry corresponds to a polygon in chunk 4 of the POLYBLOCK structure (high-resolution track polygons). The `vroadEntry` member refers to a VROADDATA structure which describes the virtual road vectors associated with the polygon.

The `flags` member describes the behavior of the polygon: a non-zero value of `flags&0x80` indicates that collisions with walls should be handled, i.e. that some of the neighboring polygons are not passable. A non-zero value of `flags&0x40` indicates that collisions with extra-objects should be handled, i.e. that some extra-objects might be present over the polygon. The next two bits in `flags` are unused. Finally, `flags&0x0f` describes the behavior of the polygon when driven over: values of 0 and 14 correspond to a polygon over which cars cannot pass; a value of 2 corresponds to a polygon which emits gravel when driven over; values 4 and 11 correspond to emission of leaves, 5 and 13 to emission of dust, 9 and 15 to emission of snow. All other values correspond to plain passable polygons.

5. VROADDATA structure (size: 12 bytes)

```
typedef struct VROADDATA {  
    short xNorm, zNorm, yNorm;  
    short xForw, zForw, yForw;  
} VROADDATA;
```

This structure describes the virtual road vectors associated with a track polygon; several track polygons can share the same VROADDATA structure. The `xNorm, zNorm, yNorm` members describe the coordinates of the normal vector (normalized to length 2^{15}) while the `xForw, zForw, yForw` members contain the coordinates of the forward vector (normalized to length 2^{15}).

6. REFJOB structure (size: 20 bytes)

```
typedef struct REFJOB {  
    struct INTPT pt;  
    short unknown1;  
    short globalno;  
    short unknown2;  
    char crossindex;
```

```

    char unknown3;
} REF_XOBJ;

```

This structure lists the properties of an extra-object. There is one entry for each non-animated (type 4) extra-object, regardless of the chunk to which it belongs; there are no entries for animated objects. The `pt` member contains the coordinates of the object's reference point (see `XOBJDATA` structure). The `globalno` member is a unique sequence number characterizing the object among all of the track's extra-objects. The `crossindex` member indicates the position of the extra-object in the first `POLYOBJ` chunk; it is equal to 0 when the object does not belong to the first chunk.

7. REFPOLYOBJ structure (size: 16 or 20 bytes)

This structure, which lists the properties of an object, exists in two versions: the version for polygon objects is 16-bytes long and has the following structure:

```

typedef struct REFPOLYOBJ {
    short entriysize;
    char type;
    char no;
    struct INTPT pt;} REFPOLYOBJ;

```

For polygon objects, the `entriysize` member equals 16 and the `type` member equals 1. The `no` member is a serial number among the block's objects; it is apparently not used by NFS3. The used numbers are not consecutive because the four `POLYOBJ` chunks are interleaved and only the first chunk is described by the `REFPOLYOBJ` structures. The `pt` member contains the reference position of the object.

In the case of extra-objects, the structure is 20-bytes long:

```

typedef struct REFPOLYOBJ {
    short entriysize;
    char type;
    char no;
    struct INTPT pt;
    long crossindex;} REFPOLYOBJ;

```

The `entriysize` member is now equal to 20, and the `type` member equals 4 (there are no animated extra-objects in the first chunk). The added `crossindex` member indicates the number of the corresponding `REF_XOBJ` structure.

There is one `REFPOLYOBJ` structure (of either type) for every object (either polygon object or extra-object) in the first `POLYOBJ` chunk. The three other `POLYOBJ` chunks are not listed. Also note that, unlike the `POLYOBJ` chunk, this structure remains present even when there are only extra-objects. The various `REFPOLYOBJ` structures of a track block are stored consecutively in the `.FRD` file; the `.FRD` files contains padding just after the `REFPOLYOBJ` structures, in order to obtain a total size of $20 * n_{Polyobj}$ in all cases for the `REFPOLYOBJ` table.

8. SOUND_SRC structure (size: 16 bytes)

```
typedef struct SOUNDSRC {  
    struct INTPT retpoint;  
    long type;} SOUNDSRC;
```

This structure describes the position and type of a sound source (water stream, etc...).

9. LIGHTSRC structure (size: 16 bytes)

```
typedef struct LIGHTSRC {  
    struct INTPT retpoint;  
    long type;} LIGHTSRC;
```

This structure describes the position and type of a light source (special lighting effect).

III. Polygon blocks

1. POLYGONBLOCK structure (variable size)

```
typedef struct POLYGONBLOCK {  
    struct POLYGONCHUNK poly[7];  
    struct OBJPOLYBLOCK obj[4];} POLYGONBLOCK;
```

This structure lists the polygons making up a track block; the `vertex` members of the contained `POLYGONDATA` structures refer to the vertex table of the corresponding `TRKBLOCK`. The `POLYGONBLOCK` structure contains seven `POLYGONCHUNK` substructures listing the track polygons, while the polygon objects are stored in the four `OBJPOLYBLOCK` substructures. Polygon chunk 0 lists the low-resolution track polygons; chunk 2 lists the medium-resolution track polygons; chunk 4 lists the high-resolution track polygons; and chunk 6 lists the polygons making up the lines between road lanes. Chunks 1, 3 and 5 are unused in most cases; occasionally they contain polygons making up fences and other transparent stuff: chunk 1 corresponds to low resolution, chunk 3 to medium resolution, and chunk 5 to high resolution. Also note that the `OBJPOLYBLOCK` substructures are non-empty only if there are polygon objects; if there are only extra-objects in an object chunk, the `OBJPOLYBLOCK` structure is left empty (the first object chunk is anyway described in the `TRKBLOCK` structure).

2. POLYGONCHUNK structure (variable size)

```
typedef struct POLYGONCHUNK {  
    long sz,szdup;  
    struct POLYGONDATA poly[sz];  
} POLYGONCHUNK;
```

The `sz` member lists the number of polygons in the polygon chunk; if it equals zero, then only the `sz` member is stored and the structure is only 4 bytes long. If `sz` is non-zero, its value is duplicated in the `szdup` member. The `poly` member then lists the polygons themselves.

3. OBJPOLYBLOCK structure (variable size)

```
typedef struct OBJPOLYBLOCK {  
    long nPolygons;  
    long nObjects;  
    struct POLYOBJDATA obj[nObjects];  
} OBJPOLYBLOCK;
```

The `nPolygons` member contains the total number of polygons referenced in the various polygon objects of the `OBJPOLYBLOCK` structure. If it equals zero, then only the `nPolygons` member is stored and the structure is only 4 bytes long. This is in particular the case when all objects are extra-objects. If `nPolygons` is non-zero, the `nObjects` member indicates the total number of objects in the chunk (both polygon

objects and extra-objects), and the objects themselves are stored in the `obj` substructure.

4. POLYOBJDATA structure (variable size)

This structure exists in two different versions. In the case of extra-objects, the structure is 4-bytes long:

```
typedef struct POLYOBJDATA {  
    long type;  
} POLYOBJDATA;
```

The `type` member equals 4 for a basic extra-object, and 3 for an animated extra-object. Since extra-objects are stored in separate XOBJDATA structures, the POLYOBJDATA structure contains no other information in that case. For polygon objects, the structure contains more information:

```
typedef struct POLYOBJDATA {  
    long type;  
    long numpoly;  
    struct POLYGONDATA poly[numpoly];  
} POLYOBJDATA;
```

The `type` member equals 1 for a polygon object. The `numpoly` member lists the number of polygons in the object, and the polygons themselves are stored in the `poly` substructure.

IV. Extra-object blocks

1. XOBJBLOCK structure (variable size)

```
typedef struct XOBJBLOCK {
    long nobj;
    struct XOBJDATA obj[nobj];
} XOBJBLOCK;
```

This structure is used to store extra-objects. There are actually four XOBJBLOCKs for every track block, each corresponding to one of the four object chunks. The FRD file also contains an extra XOBJBLOCK devoted to global animated extra-objects.

2. XOBJDATA structure (variable size)

This structure is used to describe an extra-object; it exists in two different versions, one for static objects (type 4) and one for animated objects (type 3). In the case of static objects, the structure is the following:

```
typedef struct XOBJDATA {
    long type;
    long crossno;
    long unknown;
    struct FLOATPT ptRef;
    long unknown2;
    long nVertices;
    struct FLOATPT vert[nVertices];
    long shadingVertices[nVertices];
    long nPolygons;
    struct POLYGONDATA polyData[nPolygons];
} XOBJDATA;
```

The `type` member equals 4 for static extra-objects. The `crossno` member references the `REFXOBJ` structure corresponding to the object in the `TRKBLOCK` structure of the appropriate track block. The `ptRef` member contains the coordinates of the object's reference point. The `nVertices` value is the number of vertices contained in the extra-object; the coordinates of these vertices are stored in the `vert` member. All coordinates are *relative to the reference point*. The `shadingVertices` member describes the shading applied to each vertex: the least significant byte is the blue component, the next byte is the green component, the next byte is the red component, and finally the most significant byte is the alpha value. The `nPolygon` value is the number of polygons making up the extra-object, and the polygons themselves are stored in the `polyData` member.

For animated extra-objects, the `XOBJDATA` structure becomes:

```
typedef struct XOBJDATA {
    long type;
    long crossno;
    long unknown;
```



```

short unknown2[9];
char type3,objno;
short nAnimLength,unknown3;
struct ANIMDATA animData[nAnimLength];
long nVertices;
struct FLOATPT vert[nVertices];
long shadingVertices[nVertices];
long nPolygons;
struct POLYGONDATA polyData[nPolygons];
} XOBJDATA;

```

The `type` member equals 3 for animated extra-objects. The `type3` member also equals 3, while the `objno` member contains the object's sequence number inside the track block. The `nAnimLength` member stores the length of the animation sequence; the successive object positions are stored in the ANIMDATA substructures contained in the `animData` member.

The `nVertices` value is the number of vertices in the object; the coordinates of these vertices are stored in the `vert` member. All coordinates are *relative to the reference points specified in the animation data*. The `nPolygon` value is the number of polygons making up the object, and the polygons themselves are stored in the `polyData` member.

3. ANIMDATA structure (size: 20 bytes)

```

typedef struct ANIMDATA {
    struct INTPT pt;
    float costheta,sintheta;
} ANIMDATA;

```

This structure describes the position of an animated extra-object. The `pt` member provides the position of the object's reference point (it is similar to the `ptRef` member of a static extra-object XOBJDATA structure, but in the case of an animated object the reference point moves during animation). The `costheta` and `sintheta` members describe the rotation to be applied to the object (this rotation always occurs in the XY plane, the Z coordinate is not affected).

V. FRD files

1. FRD file structure

```
typedef struct FRDFILE {
    char header[28];
    long nBlocks;
    struct TRKBLOCK trk[nBlocks+1];
    struct POLYGONBLOCK poly[nBlocks+1];
    struct XOBJBLOCK xobj[4*(nBlocks+1)+1];
    long nTextures;
    struct TEXTUREBLOCK texture[nTextures];
} FRDFILE;
```

FRD files start with a 28-byte header, followed by the number of the last track block. The actual number of blocks is `nBlocks+1` rather than `nBlocks`. The first `TRKBLOCK` structure is at offset 32 in the file; the `TRKBLOCK` structures are followed by the `POLYGONBLOCK` structures (one per track block), and the `XOBJBLOCK` structures (four per track block, plus one for global objects). These data are followed by the number `nTextures` of track textures, and by `TEXTUREBLOCK` structures which describe how these textures are obtained from the bitmaps stored in the QFS files.

2. TEXTUREBLOCK structure (size: 47 bytes)

```
typedef struct TEXTUREBLOCK {
    short width,height;
    long unknown;
    float corners[4][2];
    long unknown2;
    char islane;
    short texture;
} TEXTUREBLOCK;
```

This structure stores the properties of a track texture; the polygon textures described in `POLYCONDATA` structures always refer to one of these structures rather than directly to a QFS bitmap. The `width` and `height` members indicate the size of the bitmap in pixels. The `corners` member is a series of four planar coordinates, used to describe the positions of the four texture corners with respect to the QFS bitmap; these coordinates are often larger than 1, which makes it possible to tile several copies of a QFS bitmap into a single track texture. The `islane` member equals 1 if this texture is used to display road lanes (in which case it is *not* connected to a QFS file texture), and 0 in all other cases. Finally, the `texture` member indicates the relevant entry in the QFS file (except in the case of road lane textures where it represents the number of the relevant predefined texture).

The textures used for road lanes (i.e. those for which `islane` equals 1) correspond to 14 standard predefined textures rather than QFS textures: texture 0 is a yellow solid line, texture 1 is a yellow dotted line, texture 2 is a double yellow solid line, textures 3 and 4 are double yellow lines (one solid line with one dashed line). Textures 5 to 9 are

identical to textures 0 to 4 but are white instead of yellow. Textures 10 and 11 are solid tire marks, 12 and 13 are lighter tire marks.

WARNING? the TEXTUREBLOCK structure causes byte-alignment problems with the default options of most compilers, as its members are not aligned on even addresses; beware of this when writing track editing software.

VI. COL files

1. COL file structure

```
typedef struct COLFILE {
    char collID[4];
    long version;
    long fileLength;
    long nBlocks;
    long xbTable[nBlocks];
    struct XBCOLBLOCK xb[nBlocks];
} COLFILE;
```

The COL file has a structure comparable to the COL files of NFS2, but nearly all XBCOLBLOCK structures have changed. The only really useful information in COL files is the virtual road information, which is not redundant with what can be found in FRD files; the other structures are used to describe animated global objects, and their interest is no longer clear now that extra-objects can be animated too... however it turns out that some of the NFS3 tracks have their global objects described in the COL file rather than in the FRD file, so the description below is still necessary.

The `collID` member contains the characters 'COLL'; the `version` member equals 11. The `fileLength` member contains the length of the COL file. The file contents are stored in XBCOLBLOCK structures, which serve various purposes depending on their XBID identifier. The offset of each XBCOLBLOCK structure in the COL file is stored in the `xbTable` member; these offsets are *relative to the beginning of the XB table*, so one must add 16 in order to get an absolute offset in the COL file.

The number of blocks is either 2 if there are no global objects in the COL file, or 4 or 5 if there are global objects. The first block is used for a texture table, providing the conversion between QFS bitmaps and animated COL object textures; it is always present, even when there are no animated COL objects, and its contents are not related in any way with the texture table stored in the FRD file. The next blocks are only present if there are animated COL objects: first a block containing the 3D structures of the COL objects, followed by one or two blocks containing the objects themselves. The last block, which is always present, contains important virtual road information.

2. XBCOLBLOCK structure (variable size)

```
typedef struct XBCOLBLOCK {
    long size;
    short xbid;
    short nrec;
    struct (COLTEXTUREINFO|
            COLSTRUCT3D|
            COBJECT|
            COLVROAD) data[nrec];
} XBCOLBLOCK;
```

The `size` member contains the total length in bytes of the XBCOLBLOCK. The `xbid` member is an identifier which describes the meaning of the block's contents; the `nrec` member contains the number of data records in the XBCOLBLOCK. The meaning of these data records depends on the XBID value.

The following XBCOLBLOCK types are legal in NFS3 COL files (in their order of apparition in the COL file):

a) COL texture information (XBID 2) (8-byte data records)

```
typedef struct COLTEXTUREINFO {
    short texture;
    short unknown1;
    short unknown2;
    short unknown3;
} COLTEXTUREINFO;
```

These records store the description of a COL texture in terms of a QFS bitmap. The `texture` member refers to an entry in the QFS file.

b) COL 3D-structure information (XBID 8) (variable size records)

Before describing a COL 3D-structure record, we start with two auxiliary structures (vertices and polygons):

```
typedef struct COLVERTEX {
    struct FLOATPT pt;
    long unknown;
} COLVERTEX;
```

This structure describes a vertex of a COL 3D structure. The `pt` member contains coordinates relative to the reference position stored in the COL object information block.

```
typedef struct COLPOLYGON {
    short texture;
    char vert[4];
} COLPOLYGON;
```

This structure describes a polygon of a COL 3D structure. The `texture` member refers to an element in the COL texture information block, while the `v` member contains the numbers of the four vertices making up the polygon.

```
typedef struct COLSTRUCT3D {
    long size;
    short nVert,nPoly;
    struct COLVERTEX vertex[nVert];
    struct COLPOLYGON polygon[nPoly];
} COLSTRUCT3D;
```

The `size` member stores the size of the data record; the `nVert` and `nPoly` members contain the numbers of vertices and polygons respectively. The vertices and polygons themselves are stored in the `vertices` and `polygon` members.

c) COL object information (XBID 7 or 18) (variable size records)

These data records contain the description of a COL object; however the 3D structures (vertices and polygons) themselves are stored in the COL 3D structure block. There are two versions depending on whether the object is static or animated. For static objects the structure is the following:

```
typedef struct COBJECT {
    short size;
    char type;
    char struct3D;
    struct INTPT ptRef;
} COBJECT;
```

For static objects, the `size` member is equal to 16 and the `type` member is 1. The `struct3D` member refers to the relevant data record in the previous XBCOLBLOCK (3D structure information), and the `ptRef` member contains the object's reference position.

For animated objects, the structure becomes

```
typedef struct COBJECT {
    short size;
    char type;
    char struct3D;
    short animLength;
    short unknown;
    struct ANIMDATA animData[animLength];
} COBJECT;
```

For animated objects, the `size` member contains the size of the record, and the `type` member is 3. The `struct3D` member refers to the relevant data record in the previous XBCOLBLOCK (3D structure information). The `animLength` member contains the length of the object's animation, and the successive object positions are stored in the `animData` member (the ANIMDATA structures are the same ones as in the description of animated FRD extra-objects).

d) COL virtual-road information (XBID 15) (36-byte data records)

The following structure is used to store a vector in a COL file:

```
typedef struct COLVECTOR {
    signed char x,z,y,unused;
} COLVECTOR;
```

The three vector components are normalized so the vector has length 128.

```
typedef struct COLVROAD {  
    struct INTPT refPt;  
    long unknown;  
    struct COLVECTOR normal, forward, right;  
    long leftWall, rightWall;  
} COLVROAD;
```

There is one COL virtual-road data record for each track node; there are usually 8 track nodes (and therefore 8 COLVROAD records) for each FRD track block (see the `nStartPos` and `nPositions` members in TRKBLOCK structures).

The `refPt` member stores the coordinates of the node's reference point (a point along the road). The `normal` vector points upwards, the `forward` vector points forward, the `right` vector points to the right.

The `leftWall` and `rightWall` members indicate the distance between the node's reference point and the left and right track walls (this distance is counted along the direction pointed by the `right` vector); as cars can't cross these track walls, it is necessary to ensure that all passable road polygons lie inbetween.

NFS HS Track File Formats - V0.03, (c) Denis Auroux and Vitaly Kootin 1999

This help page describes the structure of NFS High Stakes track files.

DISCLAIMER: The information below is unofficial, and may be inexact by some aspects. It is only intended as reference information to help other NFS developers, and comes with no warranty. It may not be used for commercial purposes. It is forbidden to distribute a modified version of this document.

Contents:

- [Basic structures](#)
- [Block headers](#)
- [Track block data](#)
- [FRD file](#)

I. Basic structures

1. FLOATPT structure (size: 12 bytes)

```
typedef struct FLOATPT {  
    float x,z,y;  
} FLOATPT;
```

This structure is used to store the three coordinates of a vertex. Each of the components is stored as a floating-point number (IEEE single-precision format). The x and y axes are horizontal, while the z axis is vertical and points upwards.

2. INTPT structure (size: 12 bytes)

```
typedef struct INTPT {  
    long x,z,y;  
} INTPT;
```

This is another way to store the three coordinates of a vertex (less frequently used than the FLOATPT structure). Each of the components is stored as a 32-bit integer, and must be divided by 2^{16} in order to obtain the corresponding floating-point value (in other words, the coordinates are 16.16 fixed precision numbers).

3. POLYGONDATA structure (size: 13 bytes)

```
typedef struct POLYGONDATA {  
    short vertex[4];  
    short texture;  
    short flags;  
    unsigned char animInfo;  
} POLYGONDATA;
```

This structure describes a polygon. The `vertex` member refers to four entries in the vertex table associated with the polygon (either the vertex table of the TRKBLOCK structure in the case of a POLYBLOCK polygon, or that of the XOBJDATA structure in the case of an XOBJDATA polygon), describing the coordinates of the four polygon vertices. The `texture` member directly references a texture in the QFS file.

If `texture` is greater than 2047, it is to be interpreted as a road lane texture (dotted or solid line,

yellow or white): after subtracting 2048 to the texture number one gets the code of the road lane texture (see NFS3). Otherwise, the texture number directly points to an entry in the QFS file. However, some of the textures in the QFS file are to be skipped, namely those which contain after their local palette a comment (i.e. a piece starting with header ID 0x6F) whose text is the string "<mirrored>"; the correct texture is therefore the `texture`-th bitmap of the QFS file which does not have the "<mirrored>" comment.

Unlike NFS3 where the first vertex of a track polygon is forward left, followed by forward right, backward right and backward left, the NFS HS ordering is forward right, forward left, backward left, backward right (the mirror image).

The flags are as follows: `(flags>>2)&3` indicates the multiple of 90° by which the texture should be rotated (0 for no rotation, 1 for 90°, 2 for 180°, 3 for 270°); a non-zero value of `flags&0x10` indicates that the texture is horizontally flipped; a non-zero value of `flags&0x20` indicates that the texture is vertically flipped. The value of `(flags>>6)&7` indicates the scaling factor: 0 is no scaling; 1 means that the texture is tiled twice horizontally; 2 that the texture is tiled twice vertically; 3 indicates 4x horizontal tiling, 4 indicates 4x vertical tiling. Finally, a non-zero value of `flags&0x8000` indicates that the polygon is one-sided.

The texture is animated if the `animInfo` member is non-zero; in that case the LSB part indicates the length of the animation, and the MSB part indicates the periodicity.

4. ARRAYPTR structure (size: 8 bytes)

```
typedef struct ARRAYPTR {
    long nObjects;
    void* pObjects;
} ARRAYPTR;
```

This structure is used to describe a collection of objects: the `nObjects` member contains the number of objects, while the `pObjects` member merely contains a 32-bit space to be filled with a pointer to the objects themselves (which are stored elsewhere in the file).

II. Block headers

1. VROADBLOCK structure (size: 84 bytes)

This structure stores the virtual road data (road position, width, orientation) for a track position.

```
typedef struct VROADBLOCK {
    struct FLOATPT nodePt;
    struct FLOATPT normal, forward, right;
    float leftWall, rightWall;
    float unknown[2];
    short extraNeighbors[2];
    long unknown[4]; // first 2 are lane widths, last 2 are flags
} VROADBLOCK;
```

There is one virtual-road block record for each track node; there are usually 8 track nodes (and therefore 8 VROADBLOCK records) for each FRD track block.

The `refPt` member stores the coordinates of the node's reference point (a point along the road). The `normal` vector points upwards, the `forward` vector points forward, the `right` vector points to the right; their components are now three floating-point numbers, with normalization to unit length (note that these values are multiples of 1/128). The `leftWall` and `rightWall` members indicate the distance between the node's reference point and the left and right track walls (this distance is counted along the direction pointed by the `right` vector).

The two `extraNeighbor` values are usually equal to -1, but can be used to specify up to two neighbouring nodes other than the previous node and the following node (e.g. if there are shortcuts).

2. TRKBLOCK structure (size: 1512 bytes)

```
typedef struct TRKBLOCK
{
    long nPolygons[11];
    POLYGONDATA* pPolygons[11];
    long nVertices;
    long nHiResVert, nLoResVert, nMedResVert;
    long nVerticesDup;
    long nObjectVert;
    FLOATPT* vertices;
    long* shadingVertices;
    struct FLOATPT ptCentre;
    struct FLOATPT ptBounding[4];
    struct NEIGHBORDATA nbdData[300];
    struct ARRAYPTR xobj[4];
    long nVroad;
    struct FLOATPT minPt, maxPt;
    POLYVROADDATA* pVroad;
```

```

    long nPositions;
    struct ARRAYPTR refxobj;
    struct ARRAYPTR refobj0;
    struct ARRAYPTR soundsrc;
    struct ARRAYPTR lightsrc;
    long neighbors[8];
} TRKBLOCK;

```

This structure describes a track block; it describes how many items of each type the block contains, and provides pointers to these data. Note that in order to speed up file access the data structure stored in the FRD file contains 32-bit memory pointers; these values are to be ignored and replaced by return values of subsequent `malloc()` calls at load time.

The structure starts with eleven polygon chunks: seven for the track structure, and four for the track's polygon objects. Polygon chunk 0 lists the low-resolution track polygons; chunk 2 lists the medium-resolution track polygons; chunk 4 lists the high-resolution track polygons; and chunk 6 lists the polygons making up the lines between road lanes. Chunks 1, 3 and 5 contain polygons making up fences and other transparent stuff: chunk 1 corresponds to low resolution, chunk 3 to medium resolution, and chunk 5 to high resolution. Chunks 7, 8, 9 and 10 correspond to the polygon objects. The `nPolygons` member contains the number of polygons in each chunk, while the `pPolygons` member provides room for pointer storage.

The `nVertices` and `nVerticesDup` values are always equal, and give the total number of vertices to be stored in the `vertices` member of the TRKBLOCK structure. Vertices 0 to `nObjectVert-1` are the vertices used by the polygon objects (chunks 7,8,9,10). Vertices `nObjectVert` to `nLoResVert-1` are used in polygon chunks 0 and 1, and correspond to the track vertices in low resolution. Vertices `nLoResVert` to `nMedResVert-1` are the additional vertices used in chunks 2 and 3, and correspond to the track vertices in medium resolution (the low-resolution vertices are also used in medium resolution). Vertices `nMedResVert` to `nHiResVert-1` are the additional vertices used in chunks 4 and 5, and correspond to the track vertices in high resolution (the low- and medium-resolution vertices are also used in high resolution). Finally, vertices `nHiResVert` to `nVertices-1` are used in chunk 6, and correspond to the road lanes. The `shadingVertices` member describes the shading applied to each vertex: for each vertex, the least significant byte is the blue component, the next byte is the green component, the next byte is the red component, and finally the most significant byte is the alpha value. The structure contains two pointers used to access the vertex coordinate table and the vertex shading table.

The `ptCentre` member provides the coordinates of a central reference point around which the block is built. The `ptBounding` member provides the coordinates of four bounding points for the block vertices: these points delimit a horizontal rectangle inside which all the block's elements fit.

The four `xobj` array-pointers indicate how many extra-objects are contained in each of the four extra-object chunks associated with the track block, and provide the corresponding pointer space.

The `nVroad` member indicates how many `POLYVROADDATA` structures are used in the block to describe the virtual road parameters of the track polygons; the `pVroad` member provides pointer space.

Normally each block corresponds to 8 node positions, except the last one: the number of nodes in the block is given by the `nPositions` member. The node positions refer to the `VROADBLOCK` records at the beginning of the FRD file.

The four array-pointers `refxobj`, `refobj0`, `soundsrc` and `lightsrc` are used to access the various corresponding substructures of types `REFXOBJ`, `REFPOLYOBJ`, `SOUNDSRC` and `LIGHTSRC` respectively (see below). The number of `REFXOBJ` entries is the number of non-animated (type 4) extra-objects stored in the four extra-object chunks associated with the track block. The number of `REFPOLYOBJ` entries is the total number of objects (both polygon-objects and extra-objects) in object chunk 0.

The `neighbors` member lists blocks with which the current block is in direct contact. Usually these are the previous block and the following block. The unused entries contain -1.

The `minPt` and `maxPt` members, whose z coordinates are unused, store in their x components respectively the minimum and maximum x coordinates of any vertex belonging to a non-type14 high-resolution road polygon (i.e. a polygon for which there is a `POLYVROADDATA` entry). Similarly the y components of `minPt` and `maxPt` contain the minimum and maximum y coordinates of any vertex belonging to a non-type14 high resolution road polygon. These two members are used by the `POLYVROADDATA` entries.

3. NEIGHBORDATA structure (size: 4 bytes)

```
typedef struct NEIGHBORDATA {  
    short block;  
    short unused;  
} NEIGHBORDATA;
```

This structure is used to list the block numbers corresponding to the neighbors of the currently described block, starting with the closest ones. There are always 300 such entries in the `TRKBLOCK` structure, but for obvious reasons most of these entries are unused and just contain -1 in the `block` member.

III. Track block data

1. TRKBLOCKDATA structure (size: variable)

This structure stores the various data blocks referenced by the TRKBLOCK structure (above).

```
typedef struct TRKBLOCKDATA {
    struct FLOATPT vertices[];
    long shadingVertices[];
    struct POLYVROADDDATA vroad[];
    struct REFPOBJ refxobj[];
    struct REFPOLYOBJ refobj0[];
    char padding[];
    struct SOUNDSRC soundsrc[];
    struct LIGHTSRC lightsrc[];
    struct POLYGONDATA polygons[11][];
    struct XOBJCHUNK xobj[4];
} TRKBLOCKDATA;
```

The number of `vertices` and `shadingVertices` entries corresponds to the `nVertices` member of the TRKBLOCK structure. These contain respectively the coordinates and shading RGBA values of the track vertices.

The number of POLYVROADDDATA structures (see below) is given by the `nVroad` member of the TRKBLOCK structure.

The number of REFPOBJ, REFPOLYOBJ, SOUNDSRC and LIGHTSRC structures (see below) is given by the corresponding array-pointers in the TRKBLOCK structure.

The REFPOLYOBJ structures have variable size (16 or 20 bytes); the .FRD file contains padding just after the REFPOLYOBJ structures, in order to obtain a total size of $20 * nPolyobj$ in all cases for the REFPOLYOBJ table.

The `polygons` member contains 11 series of POLYGONDATA structures, making up the 11 polygon chunks of the track block. The length of each chunk is given at the beginning of the TRKBLOCK structure. These structures contain all polygons both for the track structure and polygon objects.

The four `xobj` members contain the block's extra-objects (4 chunks). The number of XOBJHEAD and XOBJDATA structures for each chunk are the number of objects of each `xobj` chunk, stored in the `xobj` array-pointers of the TRKBLOCK structure.

2. POLYVROADDDATA structure (size: 24 bytes)

```
typedef struct POLYVROADDDATA {
    unsigned char minY, maxY, minX, maxX;
    char isEdge[4];
    char flags;
    char unknown;
```

```

    short polygon;
    short xNorm, zNorm, yNorm;
    short xForw, zForw, yForw;
} POLYVROADDDATA;

```

This structure describes the virtual road properties of a track polygon (i.e. a polygon in chunk 4). There is one entry for each driveable polygon (non-driveable polygons of type 14 do not have an entry). The polygon's position in chunk 4 is given by the `polygon` member. The `xNorm`, `zNorm`, `yNorm` members describe the coordinates of the normal vector (normalized to length 2^{15}) while the `xForw`, `zForw`, `yForw` members contain the coordinates of the forward vector (normalized to length 2^{15}).

The `minY`, `maxY`, `minX`, `maxX` members give bounds on the x and y coordinates of the four vertices making up the polygon, and are used by NFSHS to determine which polygon the car hits as it moves. These values are scaled relatively to the `minPt` and `maxPt` members of the corresponding `TRKBLOCK` structure, by a linear transformation: 0 stands for the `minPt` value, and 255 stands for the `maxPt` value. For example, denoting by `v[i]` the four vertices of the polygon,

```
minX = 255*(min(v[i].x, i=0..3) - minPt.x) / (maxPt.x - minPt.x)
```

The `isEdge` array indicates along which of the sides of the polygon there are neighboring track polygons: the first value corresponds to the front edge, followed by the left edge, the back edge, and the right edge. These values are 0xFF if there is no adjacent track polygon along this edge, 0 if there is one.

The `flags` member describes the behavior of the polygon: a non-zero value of `flags&0x80` indicates that collisions with walls should be handled, i.e. that some of the neighboring polygons are not passable. A non-zero value of `flags&0x40` indicates that collisions with extra-objects should be handled, i.e. that some extra-objects might be present over the polygon. `Flags&0x20` seems to be set for the first row of polygons in a track block. Finally, `flags&0x0f` describes the behavior of the polygon when driven over: a value of 0 corresponds to a polygon over which cars cannot pass (14 is unused in NFSHS, see above); a value of 2 corresponds to a polygon which emits gravel when driven over; values 4 and 11 correspond to emission of leaves, 5 and 13 to emission of dust, 9 and 15 to emission of snow. All other values correspond to plain passable polygons.

3. REFXOBJ structure (size: 20 bytes)

```

typedef struct REFXOBJ {
    struct INTPT pt;
    short unknown;
    short globalno;
    char unknown[4];
} REFXOBJ;

```

This structure lists the properties of an extra-object. There is one entry for each non-animated (type 4) extra-object, regardless of the chunk to which it belongs; there are no entries for animated objects. The `pt` member contains the coordinates of the

object's reference point (see XOBJDATA structure). The `globalno` member is a unique sequence number characterizing the object among all of the track's extra-objects. There are also entries for type 6 objects; however these objects are not stored among the track block's extra-objects, so type 6 entries should mostly be ignored.

4. REFPOLYOBJ structure (size: 16 or 20 bytes)

This structure, which lists the properties of an object, exists in two versions: the version for polygon objects is 16-bytes long and has the following structure:

```
typedef struct REFPOLYOBJ {
    short head;
    char type;
    char no;
    struct INTPT pt;
} REFPOLYOBJ;
```

For polygon objects, the `head` member equals 4 and the `type` member equals 1. The `no` member is a serial number among the block's objects; it is apparently not used by NFS. The `pt` member contains the reference position of the object.

In the case of non-animated extra-objects, the structure is 20-bytes long:

```
typedef struct REFPOLYOBJ {
    short head;
    char type;
    char no;
    struct INTPT pt;
    char crossindex; // ?
    char unknown[3];
} REFPOLYOBJ;
```

The `head` member is still equal to 4, and the `type` member equals 2 or 4. The `crossindex` member refers to the corresponding REFPOBJ entry (?)

There is one REFPOLYOBJ structure (of either type) for every object (either polygon object or extra-object) in the first object chunk. The three other chunks are not listed. The various REFPOLYOBJ structures of a track block are stored consecutively in the .FRD file; the .FRD file contains padding just after the REFPOLYOBJ structures, in order to obtain a total size of $20 * n_{Polyobj}$ in all cases for the REFPOLYOBJ table.

There are also variants for type 3 and 6 extra-objects; type 6 extra-objects have 16 bytes like for polygon objects; type 3 objects contain animation data (see XOBJDATA below) which quickly fills up the space for REFPOLYOBJ structures, so that most of the table is actually lost. In any case, this table is not used by NFS, so it can safely be erased.

5. SOUNDSRC structure (size: 16 bytes)

```
typedef struct SOUNDSRC {
    struct INTPT refpoint;
```



```

    long type;
} SOUNDSRC;

```

This structure describes the position and type of a sound source (water stream, etc...).

6. LIGHTSRC structure (size: 16 bytes)

```

typedef struct LIGHTSRC {
    struct INTPT retpoint;
    long type;
} LIGHTSRC;

```

This structure describes the position and type of a light source (special lighting effect).

7. XOBJCHUNK structure (variable size)

This structure stores the data of an extra-object chunk (4 chunks per track block, plus two more chunks at the end of the file for global objects). It has the following structure:

```

typedef struct XOBJCHUNK {
    struct XOBJHEAD xobj[];
    struct XOBJDATA xobjdata[];
} XOBJCHUNK;

```

The number of XOBJHEAD and XOBJDATA structures are given in the TRKBLOCK header structure.

8. XOBJHEAD structure (52 bytes long)

This structure is used to describe an extra-object; it exists in different versions, depending on the object's type: static object (type 2 or 4), animated object (type 3), static global object (type 1), complex-behavior global object (type 6). The structure is the following (52 bytes long):

```

typedef struct XOBJHEAD {
    long type;
    long crossno;
    long unknown;
    struct FLOATPT ptRef;
    long dataSize;
    void *data;
    long nVertices;
    FLOATPT* vertices;
    long* shadingVertices;
    long nPolygons;
    POLYGONDATA* polygons;
} XOBJHEAD;

```

The `type` member equals 2 or 4 for static extra-objects, 3 for animated objects, 1 for static global objects, 6 for complex-behavior objects. The only difference between type 2 and type 4 objects is that type 2 objects do not appear in the REFEXOBJ table (they

cannot be collided with). Type 3 objects do not appear in the REFXOBJ table either; they are non-interactive animated objects, i.e. they follow a prescribed trajectory that can't be affected by cars. Type 1 objects are static objects like type 2 objects, but do not belong to any track block (they're used in the final xobj chunks of the FRD file). Finally, type 6 objects are static objects which can be knocked over by the cars, resulting in a complex behavior: e.g. the crates in Route Adonf, cones in raceway tracks, etc...; these objects are stored in the final xobj chunks of the FRD file, although they are referred to in the REFXOBJ tables of the individual track blocks (they're not present in the xobj data of the block but postponed to the end of the file instead).

The `crossno` member is only used for objects which appear in the REFXOBJ table, i.e. type 4 static objects (and type 6 global objects); it then references the entry in the REFXOBJ structure corresponding to the object.

The `ptRef` member contains the coordinates of the object's reference point (only in the case of type 2, 4 or 6; for type 3 a sequence of animation positions is used instead, and type 1 objects have global coordinates).

The `dataSize` member contains the length in bytes of the object's type-specific extra data, only non-zero for type 3 and 6 objects; the data is to be accessed via the `data` pointer member. For type 3 objects this data contains the animation's successive positions, and for type 6 objects this data is always 72 bytes long and stores the object's position and orientation.

The `nVertices` value is the number of vertices contained in the extra-object; the coordinates of these vertices are to be stored in the `vertices` member (pointer space only). All coordinates are *relative to the reference point*, except for type 1 global objects where these are absolute coordinates. The `shadingVertices` member is pointer space to store the shading applied to each vertex. The `nPolygon` value is the number of polygons making up the extra-object, and the polygons themselves are to be accessed via the `polygons` pointer.

9. XOBJDATA structure (variable size)

The XOBJDATA structure is:

```
typedef struct XOBJDATA {
    struct SPECIFICDATA data;
    struct FLOATPT vertices[];
    long shadingVertices[];
    struct POLYGONDATA polygons[];
} XOBJDATA;
```

The sizes of these tables are contained in the corresponding XOBJHEAD structure (`nVertices` and `nPolygons` members). All coordinates are *relative to the reference point* (except for type 1 objects).

The `data` member is empty for type 1, 2 or 4 objects (static objects). In the case of an animated (type 3) object, the specific data structure is:

```
typedef struct SPECIFICDATA {
    short head;
    char type;
    char no;
    short animLength;
    short unknown;
    struct ANIMDATA animData[animLength];
} SPECIFICDATA;
```

The extra information describing the animation looks much like a REFPOLYOBJ structure: `head` equals 4, `type` equals 3, `no` is an object number. The `animLength` member indicates how many ANIMDATA structures are stored.

In the case of a type 6 object (last chunk only), it becomes:

```
typedef struct SPECIFICDATA {
    struct FLOATPT ptRef;
    float weight;
    struct FLOATPT Xdir,Ydir,Zdir;
    struct FLOATPT dimensions;
    long unknown;
    short unknown2,unknown3;
} SPECIFICDATA;
```

The `ptRef` field contains the object's reference point (it is used instead of the one given in the XOBJHEAD); the `weight` member contains the object's weight. The three vectors `Xdir`, `Ydir`, `Zdir` indicate the orientation of the object and contain respectively the world coordinates of the X, Y and Z axes of the object's coordinate frame. These vectors are mutually orthogonal and have unit length. The components of the `dimensions` member provide information on the object's size along the various axes. The value of `unknown2` is related to the object's shape (cube, pyramid, ...)

9. ANIMDATA structure (size: 20 bytes)

```
typedef struct ANIMDATA {
    struct INTPT pt;
    float costheta,sintheta;
} ANIMDATA;
```

This structure describes the position of an animated extra-object. The `pt` member provides the position of the object's reference point (it is similar to the `ptRef` member of a static extra-object XOBJDATA structure, but in the case of an animated object the reference point moves during animation). The `costheta` and `sintheta` members describe the rotation to be applied to the object (this rotation always occurs in the XY plane, the Z coordinate is not affected).

IV. FRD file structure

```
typedef struct FRDFILE {
    char header[28];
    long nLastBlock;
    long nNodes;
    struct VROADBLOCK vroad[nNodes];
    struct TRKBLOCK trk[nLastBlock+1];
    struct TRKBLOCKDATA data[nLastBlock+1];
    struct GLOBALXOBJ global_xobj[2];
    // anything else ?
} FRDFILE;
```

FRD files start with a 28-byte header, followed by the number of the last track block. The actual number of blocks is `nLastBlock+1`. Besides the data corresponding to blocks (as described above), the final part of the FRD file is devoted to global (animated) objects. The following structure is used:

```
typedef struct GLOBALXOBJ {
    long nObj;
    struct XOBJHEAD xobj[nObj];
    struct XOBJDATA xobjdata[nObj];
} GLOBALXOBJ;
```

This is the typical place where animated objects are stored; object types are widely variable in these two final XOBJ chunks. The first chunk can contain objects of type 1 or 3; the second one can contain objects of type 6.

Modifying the Document

To edit a track, you need to use a combination of a selection mode and an editing tool. The available selection modes are listed in the [Edit-Modes menu](#), and the available editing tools are listed in the [Edit-Tools menu](#). In order to facilitate the editing process, the view parameters can be adjusted by using the commands in the [View menu](#).

Adding track polygons

The procedure to add track polygons has been changed since previous versions, and is now slightly less flexible but is now almost bug-free (now it should work!):

1. Switch to high-resolution view and point mode; select the Split/Merge tool.
2. Split the vertices around the polygons you want to duplicate.
3. Switch to polygon mode; select the polygon you want to duplicate, and use the Edit-Tools Duplicate to split it into as many pieces as necessary. In order to avoid bugs, it is no longer possible to move a polygon from a track node (a "row") to another one.
4. Switch back to point mode; move and merge the vertices to fit the intended disposition.
5. Switch back to polygon mode; set the passable/non-passable properties for each newly created polygon, if they differ from those of the originally duplicated polygon; similarly for the textures. In the case of a High Stakes track, don't forget to properly check the "no forward/backward neighbor" buttons in the Polygon Properties dialog box if necessary.
6. Switch to half-resolution view, and modify its contents if necessary (polygon deletion and duplication work like in versions up to 0.05).
7. Repeat the previous step in low-resolution view.

NOTE: the structure of the track polygons imposed by the NFS game engine is extremely strict. You should always ensure the following:


- each of the 8 track nodes within each track block should correspond to a single row of polygons, ordered from left to right and adjacent to each other along a common edge. Starting with v0.06, T3ED ensures this automatically in most cases, but you should be careful when importing tracks edited with older versions.
- each track polygon should be facing upwards and with the correct orientation. Once again, this is automatic in version 0.06 and above, but you should be careful when importing tracks edited with older versions.
- the track polygon type (Polygon Flags dialog box) should be set to 0 for non-passable polygons lying in the middle of the track, and to 14 instead for non-passable polygons lying to the side of the track; the polygons with type not equal to 14 should form a single row for each track node.
- each passable polygon should have the collision detection flags set correctly; in NFSHS the absence of passable neighbor also has to be indicated (in the Polygon Flags dialog box for forward and backward neighbors; automatically handled for left and right neighbors).
- the road width should be adjusted in order to avoid invisible walls.

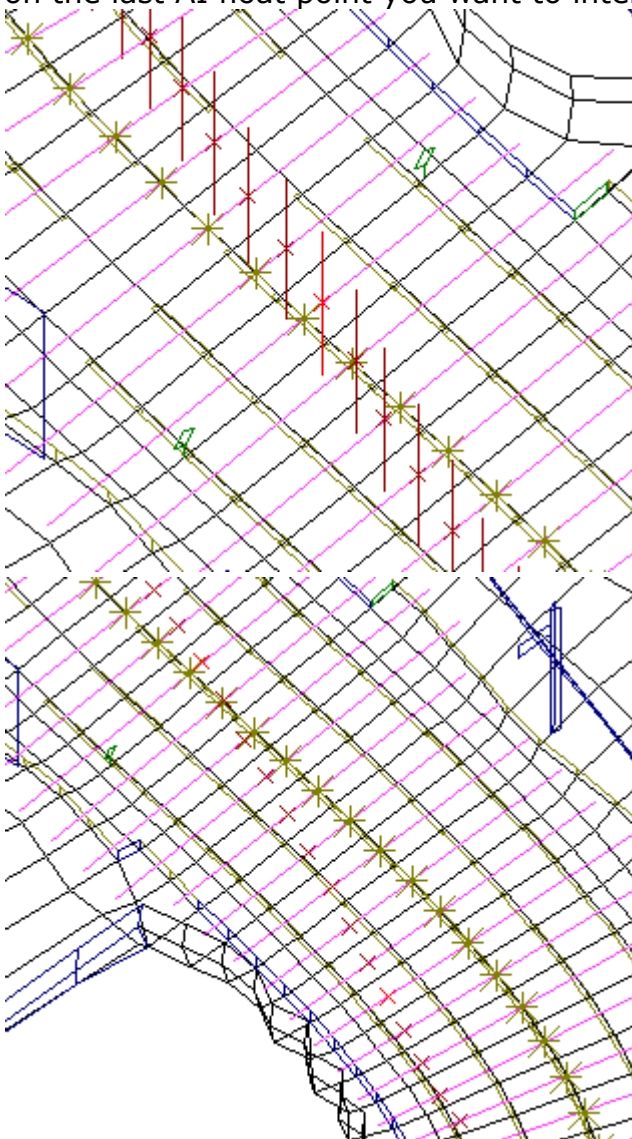
If any of the above fails, the track is likely to contain invisible walls and other bugs; remember, adding track polygons is a complicated process, so be sure to check carefully what you do.

A final note: if you want to create a track which branches into two separate ways, it is still necessary to keep track polygons between the two parts of the track (remember each track node should be a single row of adjacent polygons); you can either give them the proper textures, or even in some cases make them transparent (select a proper empty texture).

Quickly fine-tuning opponent AI line float & speed values

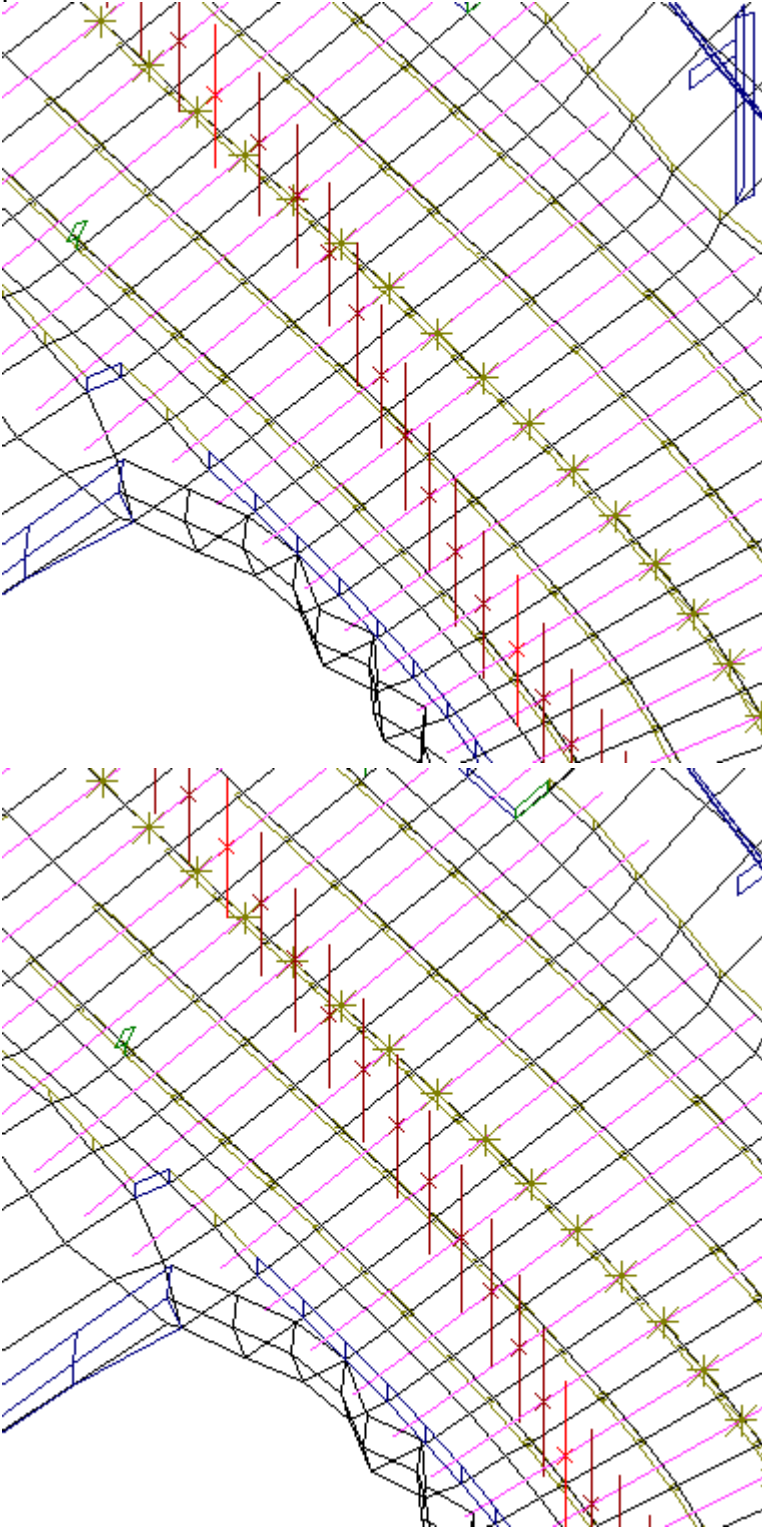
If you want to quickly fine-tune opponent AI lines, you can use the interpolation function for AI line float points and speeds. To do so:

- Change to Virtual Road Point mode . You will see the AI line points (dark yellow asterisks) along side their opponent AI line float values (dark red Xs and lines)
- Hold Alt and click on the first AI float line point. Then hold Ctrl and Alt and click on the last AI float point you want to interpolate all the way with the first.



- Press Shift + Q to interpolate the AI points between the first and last points you selected as a curve; if you want a straighter line between the two points, press Shift + Alt + Q. If you want to recalculate the speeds between points as well,

press Shift + S.



INI file Tutorial

by John Arnström (a.k.a Zacabeb) - original author; AJ_Lethal - a few revisions

1. [The INI files](#)
 - [1.1 What's in the INI files?](#)
 - [1.2 What do I need to edit the INI files?](#)
 - [1.3 What can I do by editing the INI files?](#)
2. [Variables in the INI file](#)
 - [2.1 Horizon Strip \[strip\]](#)
 - [2.2 Clouds \[clouds\]](#)
 - [2.3 Lightning \[lightning\]](#)
 - [2.4 Distance Fog \[fog\]](#)
 - [2.5 Weather \[weather\]](#)
 - [2.6 Ambient Light and Car Shadows \[light\]](#)
 - [2.7 Special Effects \[special\]](#)
 - [2.8 Sun \[sun\]](#)
 - [2.9 Track Glows \[track glows\]](#)

1. The INI files

1.1 What's in the INI files?

The INI files contain weather and light specifications for the tracks in NFS:HS. They originate from the HRZ files found in NFS3, but with the enhancements made in NFS:HS they have been augmented with new features.

There are four INI files; one for each of the time/weather combinations available. The names of the files correspond to the conditions, as follows:

TR.INI	Daytime
TRW.INI	Daytime with weather turned on
TRN.INI	Nighttime
TRNW.INI	Nighttime with weather turned on

1.2 What do I need to edit the INI files?

All you need to edit the INI files is a text editor; by default Windows will use Notepad for editing INI files. If you have selected a different text editor, make sure you save the file in ASCII format or it will appear corrupted to the game and crash it.

1.3 What can I do by editing the INI files?

By editing the INI files, you can change the weather, light and sky color among other things. You can add thunder, rain or snow, change the position and size of the sun, alter the fog and edit the color and brightness of the light sources on the track. With careful editing you can change the atmosphere of a track only by changing values in the INI files - creating a misty summer morning or a chilling night scene. By editing the INI files you can add the final touch to the tracks you make - letting them appear just the way you want.

2. Variables in the INI file

2.1 Horizon Strip [strip]

The strip contains information on the horizon and its properties.

hrzShouldDrawBlack=b

This is a boolean which specifies whether the screen should be filled with black before each frame is drawn, in which case the value is 1, if not the value is 0. When set to 1, no pixmap or horizon is drawn. This is useful if the track does not contain any horizon or pixmap (see below).

This feature is not used on any of the original tracks in NFS:HS, the ones from NFS3 included. In NFS2 however, it was used on Monolithic Studios, and in NFS3 for the PlayStation it was used on the Easter Egg track Caverns. There is also a corresponding function in the .FRD files (where the track model is kept), which specifies the ShouldDrawBlack variable individually for each slice of the track. This is used in the Temple on the track Lost Canyons.

hrzMirror=b

This is a boolean and tells if the pixmap (see below) should be mirrored in the two halves of the horizon (1) or if it should be repeated with the same orientation in the two halves (0). For prominent horizons such as hills, it is often best to make a pixmap which can be repeated seamlessly in the two halves and set hrzMirror to 0, otherwise the mirroring of the two halves will cause a Rorschach effect at the seam. When you want there to be a scene only in the one end of the horizon, draw it in one end of your pixmap set hrzMirror to 1.

hrzHasPmx=b

This specifies whether the track has a pixmap, i.e. a background landscape. This is a boolean value set to 1 if a horizon pixmap is used, and a 0 if not. This can be used to leave out the pixmap if none is needed, such as on a track set on an island in the middle of the ocean.

hrzBottomYOff=y

This specifies the offset for the pixmap and clouds relative to the logical horizon (the actual middle of the 3D world itself). Normally this offset has a negative value, placing it below the logical horizon. In most cases, a value at around -1000 to -1500 is used.

hrzGouraudHeight=h

This is a single? precision floating point value indicating the height of the Gouraud shaded region of the sky and ground.

hrzGouraudMiddle=y

This is a single? precision floating point value indicating the position of the center of the Gouraud shaded region of the sky and ground.

hrzPmxTop=y

This value specifies how high in the 3D world the top of the pixmap is relative to the horizon offset, hrzBottomYOff. The pixmap will appear closer or further away depending on the radius of the horizon; hrzRadius, so you will probably want to adjust the latter before you adjust hrzPmxTop. Commonly the value is at around 1500.0-2000.0 for a horizon with a radius at around 2000.0 and a hrzBottomYOff at -1000.

hrzPmxBottom=y

This specifies the bottom of the pixmap in the same manner as hrzPmxTop specifies the top. Commonly the bottom of the pixmap is slightly above the physical horizon and lying at 500-1000, placing the bottom of the pixmap near the logical horizon. When this is the case, the pixmap's bottom should be a bit below the horizon offset with a margin of about 50 units, to hide the seam where the sky and ground meet at the logical horizon.

hrzSunColor=[r,g,b]

This specifies the RGB color for the sky at the position of the horizon. The RGB values are expressed in normal 24-bit format and can each range from 0-255, where 0 is dark and 255 is bright.

hrzOppositeSunColor=[r,g,b]

This specifies the RGB color of the sky at the point opposite of the sun.

hrzSkyTopColor=[r,g,b]

This specifies the RGB color at the top of the sky. This will gradually blend with the hrzSunColor and hrzOppositeSunColor values towards the horizon, above the top of the Gouraud shaded area the sky will be a solid color.

hrzEarthTopColor=[r,g,b]

This specifies the RGB color at the top of the ground visible outside the track area.

hrzEarthBotColor=[r,g,b]

This specifies the RGB color at the bottom of the ground visible outside the track area.

hrzfRadius=r

This specifies the radius of the horizon, and affects the appearance of the pixmap, clouds and shading. Common values range from 1500.0-2500.0.

hrzAngle=?

This specifies the rotation around the horizontal axis, Theta (or ?), of the horizon in a counter-clockwise direction and is expressed in revolutions. A revolution is a full 360° rotation. To convert degrees to revolutions, use the following formula:

$$r=d/360$$

where d is the angle in degrees and r the result expressed in revolutions. Round the result off to 2 decimals.

Note that you cannot use this formula directly in the INI file; you must calculate the result and insert it 'as is'.

2.2 Clouds [clouds]

cloudType=t

This is an integer and indicates the type of clouds to be applied. A value of 0 means no clouds will appear, a value of 1 means the clouds will be blended with the sky and finally a value of 2 means they will be blended as additive.

cloudBright=b

This is an integer and specifies the intensity, or brightness of the clouds. The value can range from 0 to 255, where 0 is dark and 255 bright.

cloudVariance=v

This is an integer describing the variance of the clouds, i.e. the contrast between the bright and dark areas. Values may range from 0 to 255, where 0 means no variance and 255 full variance.

cloudNimbus=n

This is a fixed point variable and tells if the clouds produce a Nimbus effect in thunderstorms. The Nimbus effect is simply the refraction of light within the clouds which make them light up with the flashes. Values may range from 0.00 to 1.00, where 0.00 means there is no Nimbus effect at all and 1.00 there should be a strong Nimbus effect.

cloudFog=f

This is a boolean value which simply tells whether the fog on the track should affect the clouds, in which case it is set to 1; if not it is set to 0.

cloudVisualHeight=h

windSpeed=s

This is an integer specifying the windspeed. The windspeed affects the drifting of the clouds, as well as special effects such as rain and smoke. A value of 0 means the clouds and effects do not drift. For slow drift, use values around 20, for faster drifts use values between 60 and 90.

windDir=a

This is an integer describing the direction of the wind (in a counter-clockwise direction), again affecting both the clouds and special effects. Unlike the other angles in the INI file, this is expressed in degrees and can thus range from 0 to 359.

cloudYOffset=200.0

cloudHeight=500.0

cloudUscale=1.0

2.3 Lightning [lightning]

These two values describe if lightning should occur and if so, how often. When lightningChance is set to a value above 0, there is lightningChance percent chance lightning will strike in lightningOffTicks time.

If you want frequent flashes, set the percentage of lightningChance reasonably high and lightningOffTicks low. If you want occasional flashes, set both lightningChance and lightningOffTicks low. Note that if you set the lightningChance too high, flashes will occur with a distinct recurrence which might appear unnatural.

lightningChance=c

This is an integer and tells, in percent, how great the chance is a flash will occur in the period specified in lightningOffTicks. Normally, keep this value below 80 for a naturally random recurrence and adjust the lightningOffTicks value to determine the periodicity of the flashes. If you don't want any lightning set this to 0.

lightningOffTicks=o

This specifies the periodicity of the lightning in ticks.

2.4 Distance Fog [fog]

Fog comes in two forms; global and dynamic. The global fog affects the entire track in all areas where no dynamic fog zone has been defined. Dynamic fog zones let you add specific fog to a region of the track, and also fade smoothly between the different fogs.

fogColor=[r,g,b]

This specifies the color of the global fog in 24-bit RGB. Each value is an integer and may range from 0 to 255 where 0 is dark and 255 bright. The scenery is blended towards this color by the amount specified in fogDensity.

fogDensity=d

This specifies the density of the fog in percent, ranging from 0 for no fog at all up to 100 for completely opaque fog.

fogHorizon=h

This is a fixed point value and tells if the horizon should get affected by the fog, ranging from 0.00 for no effect up to 1.00 for full effect. Normally keep this at around 0.70-0.80.

fogNumRegions=n

This value is an integer and specifies how many fog regions have been defined, thus it should equal the number of dynamic fog regions in your INI file.

[fog region n]

- **startSlice=ss**
- **s_color=[sr,sg,sb]**
- **s_density=sd**
- **centerSlice=cs**
- **c_color=[cr,cg,cb]**
- **c_density=cd**
- **endSlice=es**
- **e_color=[er,eg,eb]**
- **e_density=ed**

These are dynamic fog regions which specify a fog unique to a certain part of the track. fog region specifies the number of the region in the list and should start with 0 for the first region, followed by 1,2,3... etc for each subsequent fog region. startSlice, centerSlice and endSlice specify three positions along the track for the fog region. At each position the fog will have the properties specified in the *_color and *_density variables below the respective Slice entry. Between these positions, the game will gradually blend the nearest two values to fade between the different fogs.

The positions along the track specified in the Slice variables correspond to the rows of polygons making up the track. To determine the desired slice, you can open the track in T3ed and select Segment mode. Then move to the block you want and select it. In the lower right corner of the window you will see the number of the block. To determine the number of the first slice of that block, multiply the block number by 8. This will give you a preliminary value to use which you can fine tune later.

Note that where two fog regions join, they should both have the same color and density at the joining ends. Otherwise the fog will instantly switch from the one value to the other as you drive inbetween the regions. Similarly, where either end of a fog region does not join another region, they should have the same values as the global fog specified in fogColor and fogDensity to avoid switching.

If you need to cover a large section of the track with an even fog area, use two fog regions where the values at the opposing ends are the same as their adjacent fogs, and the four values in between have the values you want for the large fog area. Then set the centerSlice of both fog regions 10-50 slices in from the Slices at the opposing ends, so to create a transition region at either end. Finally set the two adjacent Slices of the two fog regions to any value in between the two centerSlices to make them join and complete the fog area.

2.5 Weather [weather]

type=t

This specifies the type of weather to be used, where 0 means no weather, 1 means rain and 2 means snow.

startSlice=s

startSlice specifies an optional slice where the weather should start and can be used if you want to restrict the weather to a certain region of the track. Again, the Slice value refers to the slices of the track. Look above in the explanation of fog regions for

information on how to determine the desired Slice value. If you want the weather to cover the entire track, set this variable and endSlice to 0.

endSlice=e

This variable is connected to the startSlice and specifies the slice at which the weather should end.

fade=f

When you have restricted the weather to a certain part of the track using startSlice and endSlice, this specifies a number of slices at either ends of the weather region across which the weather should be faded, that is gradually become denser as you drive into it. Reasonable values range from 5 for a quick transition up to 50 for a smooth transition.

stayTimeOn=t

Together with the other Time variables, this lets you choose to have the weather come and go periodically. stayTimeOn specifies how long the weather should remain active before the break, and is counted in ticks.

fadeTimeOff=t

When you have chosen to have the weather come and go periodically in the stayTimeOn and StayTimeOff variables, this specifies a time during which the weather should fade out. Again, this is measured in ticks.

stayTimeOff=t

Like the stayTimeOn variable, this is used for making the weather break periodically and specifies how long the break should last before the weather comes back on. If you want the weather to remain constant, set this variable to 0.

fadeTimeOn=t

While the fadeTimeOff variable specifies a time over which the weather should fade out, this specifies a time during which the weather should fade back in after the break. As is the case with the other Time values, this is measured in ticks.

density=100

This specifies the density of the weather in percent. 0 means there will be no downpour, and 100 that the downpour will be as dense as the game permits.

2.6 Ambient Light and Car Shadows [light]

- **AmbientRed=r**
- **AmbientGreen=g**
- **AmbientBlue=b**

These specify the ambient light for the track and are measured in percent. At a value of 0, the track will be completely dark, while at a value of 100 it will be as bright as has been specified in the internal shading values of the track file itself. If you want to

make the track dark, choose values which lie at around 20 or more, anything less will make the track unplayable when the headlights have been busted. If you want the track to be fully lit, set the values to 100 or near.

- **lightTheta=?**
- **lightRho=?**

These two values are used to determine the source of the incoming light for the car shading. Both have two decimals and specify the angle around the horizontal axis (Theta) and vertical axis (Rho). Both are expressed in revolutions. lightTheta expresses the horizontal angle in a counter-clockwise direction, where 0.00 equals East, 0.25 equals North, 0.50 equals West and 0.75 equals South. lightRho expresses the vertical angle ranging from 0.00 for Zenith where the sun is placed at the top of the sky, and 0.39 where it lies at the horizon. Normally set lightTheta to the value opposite that of the angleTheta of the Sun (see below) and lightRho to the same value as the angleRho for the Sun.

carshadow=[i,r,g,b]

This specifies the intensity and color of the car shadow. i tells the intensity of the shadow and may range from 0 (none) to 255 (fully opaque). r,g,b specify the color in normal 24-bit RGB format, thus each value can range from 0 (dark) to 255 (bright). For most cases, set i to around 100, and each of r,g,b to 255.

2.7 Special Effects [special]

These variables specify certain special effects which can be used on the tracks.

meteor=b

This is a boolean and tells if there should be 'shooting stars' on the track, in which case it should be set to 1, otherwise it should be set to 0.

flare=[r,g,b]

When the track features a sun, this specifies the intensity of the sun and lens flares which appear when the sun is in view. Each of r,g,b can range from 0 to 255 and specify the intensity of the respective color, where 0 means no flare and 255 full intensity. If you don't want a flare to appear, for instance if you have a clouded sky with only a vague sun or a moon at night, set each of r,g,b to 0. For an intense flare, set each of r,g,b to 255.

- **fireworks=f**
- **fireworksAngle=?**

This lets you add fireworks to the track, but it is a restricted effect which depends on the date the game was installed. If the game was installed before the 4th of July, fireworks will only have appeared on the 4th of July, if it was installed later than the 4th of July fireworks will always appear when this option is set.

fireworks tells whether fireworks should appear. A value of 0 means no fireworks. fireworksAngle expresses the Rho or vertical angle of the fireworks and is expressed in revolutions, where 0.00 places them at the top of the sky and 0.39 at the horizon.

However, the Theta or horizontal angle of the fireworks cannot be specified; they will always appear opposite the Sun.

2.8 Sun [sun]

This contains information on the sun of the track, however the sun may also be a moon (!).

hasSun=b

This is a boolean and indicates whether the track should have a sun (1) or not (0). If this variable is left out on tracks made from the NFS3 tracks, the sun will appear ??? where???

- **angleTheta=?**
- **angleRho=?**

These specify the position of the sun on the horizon in revolutions. angleTheta expresses the horizontal angle and may range from 0.00 to 1.00. angleRho expresses the vertical angle and may range from 0.00 to 0.39. To make the sun appear within the view, normally set the angleRho between 0.10 and 0.30, however it might be obscured by the pixmap if the latter is high, and the inFront variable (see below) has been set to 0.

radius=r

This is a decimal value expressing the radius of the sun. For a normal sun, set this to around 200.0, for a big sun set it to around 300.0, and for an oversize novelty sun set it to 500.0 - or a bit above :).

rotates=b

This boolean tells if the sun should rotate around its own axis as you turn, in which case it should be set to 1. If your sun is a moon (!), for obvious reasons you might not want to use this, so then set it to 0.

additive=b

This boolean tells if the sun should be pasted on top of the sky (0) or blended with it (1).

inFront=b

A boolean which specifies whether the sun should appear in front of the pixmap (1) or behind it (0).

2.9 Track Glows [track glows]

The track glows contain the specifications of the track's glows, which are the light sources seen on lamp posts and similar. The values specified in the track glows were previously common for all the tracks, but since the inclusion of specific variables in the INI files their appearance can be edited. Normally you will find most of the glows to have the same values on all the original tracks. Not all of the glows are used, on many tracks only two or three are.

glow*=[i,r,g,b], f, t, ?, d

These arrays specify the color and behaviour of the glows. A total of 32, named glow0 through to glow31 appear in the original files, however not all of them are used on every track.

[i,r,g,b] specify the intensity and RGB color of the light, and can all range from 0 to 255.

f is a boolean and tells if the glow is solid (0) or flashes (1), in which case t tells how long the glow should wait in between changing its state. At a low t value, such as 2 or 3, the glow will flash very quickly. I would not recommend using values lower than 5 as the resulting strobe effect can appear disturbing. At a slightly higher value, such as 6 or 7, the flashing is slow enough to be suitable for objects such as flashing stoplights or similar.

d is a fixed point value with two decimals, specifying the diameter of the glow itself. An diameter of 0 means the glow will not appear, which is useful for placing light sources without any glow with T3ed's Light Ray Tracer.

There are also three unknown variables which only appear in some of the INI files, and always with the same values (0,0), T0. Experimenting with these values does not seem to have any noticeable effect other than in some cases eliminating the respective glow. Since these variables are not present in many of the INI files and when they appear always have the same consistent values, it can be assumed they do not have any proper use in the game. These variables might be part of features which were left out or never put in. They are however redundant and can be left out.

Selecting elements

To select an element in T3ed, you simply click on it. Depending of the element, you might have multiple ways of selecting them.

Track blocks

You can only select one track block at a time.

Track objects

You can select multiple objects by holding Ctrl or Shift and clicking on each one.

Points

- You can select multiple separate points by holding Ctrl and clicking on each one.
- You can select multiple points in a row by holding Shift, clicking on the first point, then click on the last point you want to select of the row.
- Selecting points will also give you a measure from the first to the last point you've selected. This measure is used for some tools such as [Enter move distance](#).

Polygons

- You can select multiple separate polygons by holding Ctrl and clicking on each one.
- You can select multiple polygons in a row by holding Shift, clicking on the first polygon, then click on the last polygon you want to select of the row.

Virtual Road points

- You can select multiple Virtual Road points by holding Ctrl or Shift and clicking on each one.
- You can select multiple Virtual Road float value points by holding Ctrl + Alt and clicking on each one.

File menu commands

The File menu offers the following commands:


Open	Opens an existing track file.
Save	Saves the opened track using the same file name.
Save As	Saves the opened track to a specified file name.
Recent files	List of the 4 most recent files opened
Properties	Shows properties of the currently opened .frd file
Set speedfile shift amount	Select amount of shifting for data in the speedfiles upon saving. Default is 0.
Exit	Exits T3ED.

Open command (File menu)

Use this command to open an existing track file. A [dialog box](#) will appear to let you choose the file to open.

IMPORTANT (NFS3): *Although you are prompted only for a .FRD track file, T3ED also simultaneously loads a companion file with .COL extension, which contains miscellaneous track data. If the .COL file is not available, T3ED will be unable to open the track file.*

Shortcuts

Toolbar	Keys
	Ctrl + O

File Open dialog box

The following options allow you to specify which file to open:

- **File Name** Type or select the filename you want to open. This box lists files with the extension you select in the List Files of Type box.
- **List Files of Type**
Select the type of file you want to open:
 - *.FRD: NFS3/HS track files.
- **Drives**
Select the drive in which T3ED stores the file that you want to open.
- **Directories**
Select the directory in which T3ED stores the file that you want to open.
- **Network...**
Choose this button to connect to a network location, assigning it a new drive letter.


IMPORTANT (NFS3): *Although you are prompted only for a .FRD track file, T3ED also simultaneously loads a companion file with .COL extension, which contains miscellaneous track data. If the .COL file is not available, T3ED will be unable to open the track file.*

Save command (File menu)

Use this command to save the edited track to its current name and directory. If you want to change the name and directory of a track before you save it, choose the [Save As command](#).

IMPORTANT (NFS3): *Besides the .FRD track file, T3ED also simultaneously saves a companion file with .COL extension, which contains miscellaneous track data. If you want to distribute the saved track, you must provide both the .FRD file and the .COL file. Also, both these files must be writeable in order for the save operation to be successful.*

Shortcuts

Toolbar	Keys
	Ctrl + S

Save As command (File menu)

Use this command to save and rename the edited track. T3ED displays the [Save As dialog box](#) so you can name your track file.

To save a track with its existing name and directory, use the [Save command](#).

IMPORTANT (NFS3): *Besides the .FRD track file, T3ED also simultaneously saves a companion file with .COL extension, which contains miscellaneous track data. If you want to distribute the saved track, you must provide both the .FRD file and the .COL file. Also, both these files must be writeable in order for the save operation to be successful.*

File Save As dialog box

The following options allow you to specify the name and location of the file you're about to save:

- **File Name**
Type a new filename to save a track with a different name. T3ED adds the extension you specify in the Save File As Type box.
- **Drives**
Select the drive in which you want to store the document.
- **Directories**
Select the directory in which you want to store the document.
- **Network...**
Choose this button to connect to a network location, assigning it a new drive letter.

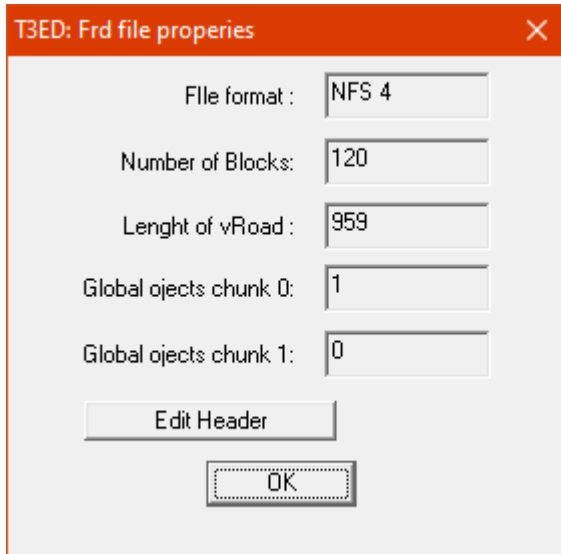
IMPORTANT (NFS3): *Besides the .FRD track file, T3ED also simultaneously saves a companion file with .COL extension, which contains miscellaneous track data. If you want to distribute the saved track, you must provide both the .FRD file and the .COL file. Also, both these files must be writeable in order for the save operation to be successful.*

Recent files (File menu)

Lists the 4 most recent files you opened. You can open them by clicking on any of them or typing its corresponding number in the keyboard.

Properties command (File menu)

Displays properties of the .FRD file





Edit Header

Opens a dialog to edit the .FRD file header. **Only use this option if you know what you're doing.**

Exit command (File menu)

Use this command to exit T3ED. T3ED prompts you to save the edited track if changes have been made since the last time it was saved.

Shortcuts

Window	Keys
click on 	Alt + F4
-or- double click on 	

Edit menu commands


The Edit menu offers the following commands:

Undo	Undoes the last action
Undo, but keep VRoad	Same as Undo but it won't undo changes made to the Virtual Road
Copy	<p>Copies the selected element to the clipboard. Elements that can be copied are:</p> <ul style="list-style-type: none">• Track blocks• Track objects (including light and sound sources)• Replay cameras• Point positions• Virtual Road points
Cut	Cuts the selected element to the clipboard. Only works for track blocks.
Paste	<p>Pastes the element stored in the clipboard. Options depend of the element being pasted.</p> <ul style="list-style-type: none">• Ctrl + Alt + V pastes the element in with same coordinates it was copied from.
Find	Finds and selects polygons based on the criteria you select. Only works in Polygon mode.
Merge	
Delete	Deletes the selected element.
Copy Z	Copies Z position of selected point. Only works in Point mode.
Paste	Pastes position of copied point. Only works in Point mode
Paste mode	Changes paste mode of Copy/Paste Z command
Paste X+Y	Pastes X and Y positions of copied point. Only works in Point mode.
Paste Z	Pastes Z position of copied point. Only works in Point mode.

Undo command (Edit menu)

Use this command to reverse the last editing action, if possible. You can undo up to the last 64 actions you performed.

Shortcuts

Toolbar	Keys
	Ctrl + Z -or- Alt + Backspace

Paste command (Edit menu)

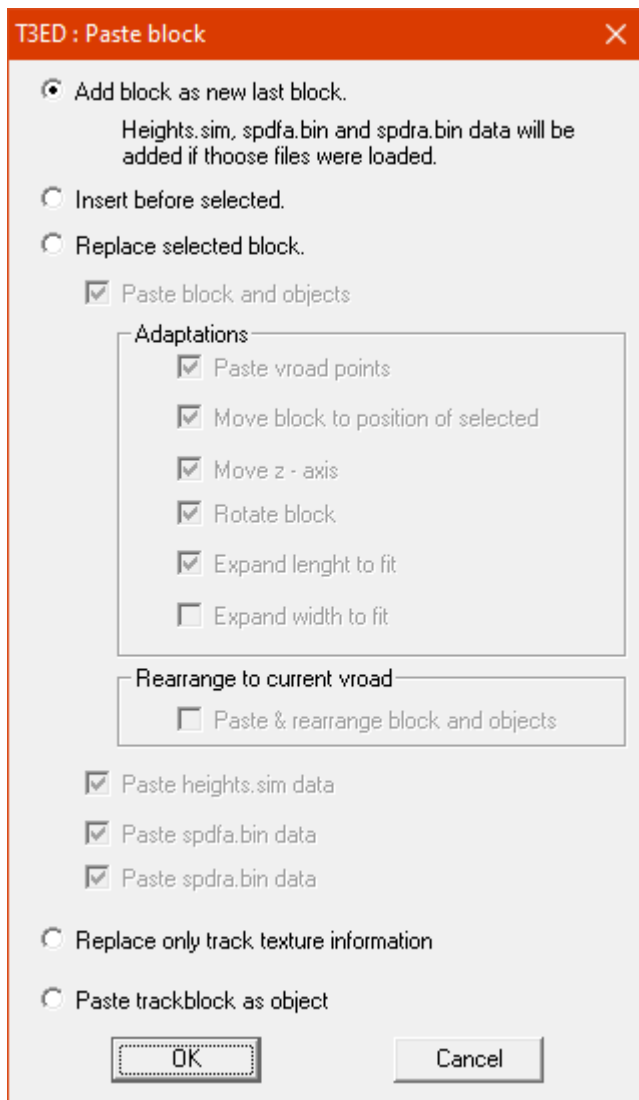
Use this command to paste the element you copied to the clipboard. Options depend on the type of element you copied.

Supported elements are:

- [Track blocks](#)
- [Points](#)
- [Objects](#)
- [VRoad points](#)

Track blocks

The following dialog will appear when you paste a track block:



The screenshot shows a dialog box titled "T3ED : Paste block" with a close button (X) in the top right corner. The dialog contains several options for pasting a track block:

- ☒ Add block as new last block.
Heights.sim, spdfa.bin and spdra.bin data will be added if those files were loaded.
- ☐ Insert before selected.
- ☐ Replace selected block.
- ☒ Paste block and objects
 - Adaptations
 - ☒ Paste vroad points
 - ☒ Move block to position of selected
 - ☒ Move z - axis
 - ☒ Rotate block
 - ☒ Expand lenght to fit
 - ☐ Expand width to fit
 - Rearrange to current vroad
 - ☐ Paste & rearrange block and objects
- ☒ Paste heights.sim data
- ☒ Paste spdfa.bin data
- ☒ Paste spdra.bin data
- ☐ Replace only track texture information
- ☐ Paste trackblock as object

At the bottom, there are "OK" and "Cancel" buttons.

Add block a new last block

Pastes the copied block as the last block of the track. Heights, spdfa and spdra data will be added as well if those files were loaded.

You'll have to move the track blocks nearby to see the changes.

Insert block before selected

Pastes the copied block before the currently selected block. You'll have to move the track blocks nearby to see the changes.

Replace selected block

Pastes the copied block replacing the currently selected block.

Paste block and objects

Pastes the copied block with its objects.

Adaptations

You can choose the following adaptations upon pasting the block:

- Paste VRoad points
- Move block to the position of selected
- Move z-axis
- Rotate block
- Expand length to fit
- Expand width to fit

Rearrange to current VRoad

- Paste & rearrange block and objects

Replace only track texture information

Only pastes the track texture settings from the copied block.

Paste track block as object

Pastes the copied track block as a polygon object.

Remarks

- Holding Alt while pasting will paste the block with the same coordinates it was copied from.
- Keep on mind you'll have to move and merge the pasted track block points with its neighbours after pasting it.

Points

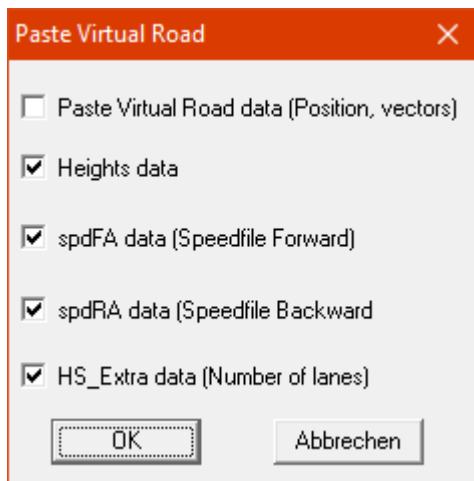
Pastes the copied point's coordinates into the selected point.

Objects

Pastes the copied object in the currently selected block. Holding Alt while pasting will paste the object with the same coordinates it was copied from.

VRoad points

The following dialog with the following options to toggle will appear when you paste a VRoad point:



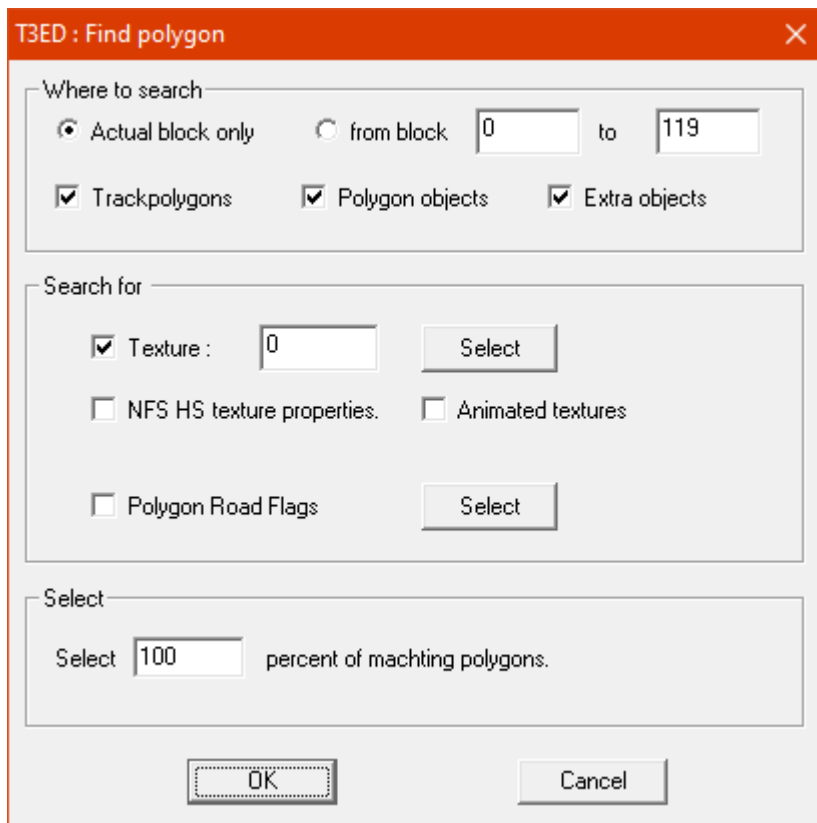
- Paste Virtual Road data (Position, vectors)
- Heights data
- spdFA data (Speedfile Forward)
- spdRA data (Speedfile Backward)
- HS_Extra data (Number of lanes)

Shortcuts

Keys
Ctrl (+ Alt) + V

Find polygon command (Edit menu)

Use this command to find and select polygons based on the criteria you select. Only works in Polygon mode.



The image shows a screenshot of the 'T3ED : Find polygon' dialog box. The dialog has a title bar with the text 'T3ED : Find polygon' and a close button (X). It is divided into three main sections: 'Where to search', 'Search for', and 'Select'. In the 'Where to search' section, there are two radio buttons: 'Actual block only' (selected) and 'from block' (with input fields for '0' and '119'). Below these are three checked checkboxes: 'Track polygons', 'Polygon objects', and 'Extra objects'. The 'Search for' section contains three groups of options: 'Texture : 0' with a 'Select' button, 'NFS HS texture properties.' and 'Animated textures' (both unchecked), and 'Polygon Road Flags' (unchecked) with a 'Select' button. The 'Select' section has a label 'Select' followed by an input field containing '100' and the text 'percent of machting polygons.' (note the typo). At the bottom are 'OK' and 'Cancel' buttons.

T3ED : Find polygon

Where to search

☒ Actual block only ☐ from block to

☒ Track polygons ☒ Polygon objects ☒ Extra objects

Search for

☒ Texture :

☐ NFS HS texture properties. ☐ Animated textures

☐ Polygon Road Flags

Select

Select percent of machting polygons.

Where to search

Actual block only

Searches in currently selected block only.

From block

Searches in the range of the blocks you specify in the fields (left field is starting block and right field is end block).

Trackpolygons/Polygon objects/Extra objects

Toggles search for track polygons, polygon objects or extra objects.

Search for

Texture

If enabled, searches the texture you specify in the field to the right. You can also pick it with the [Texture](#) dialog if you click on the Select button.

NFS3/HS Textures properties

If enabled, it will search for the selected texture properties you specified with the Select > Texture > Properties dialog. It will also enable automatically upon selecting texture properties beforehand.

Animated textures

If enabled, it will search for animated textures. It will also enable automatically upon enabling the Animated flag with the Select > Texture > Properties dialog beforehand.

Polygon Flags

If enabled, it will search for the selected [polygon flags](#) you specified with the Select button to the right. It will also enable automatically upon selecting polygon flags beforehand.

Select

Select % of matching polygons

Selects random percentage of polygons specified in the field.

Shortcuts


Keys
Ctrl + F

Delete command (Edit menu)

This command can be used to delete the currently element. It can be used for the following elements:

- Track blocks
- Track objects (including light and sound sources)
- Replay cameras
- Polygons
 - The last track polygon of a track block cannot be deleted.
 - Deleting the last polygon of an object automatically deletes the object.
- Point positions

Shortcuts

Toolbar	Keys
	Del

View menu commands

The View menu offers the following commands:

Toolbar	Shows or hides the toolbar.
Status Bar	Shows or hides the status bar.
Zoom In	Zoom in.
Zoom Out	Zoom out.
Translation mode	Sets the translation mode: the scrollbars can be used to scroll the track view.
Rotation mode	Sets the rotation mode: the scrollbars control the observer's viewpoint.
Resolution	Choose one of the three possible display resolutions.
Show Trackblocks & Fences	Shows or hides track (black) and fence (dark yellow) polygons.
Show Objects	Shows or hides the track objects.
Show Replay Cameras	Shows or hides replay cameras. They will be shown as magenta camera icons.
Show Lanes	Shows or hides the road lanes.
Show Trackview	<p>Opens up the 3D track preview window. You will be prompted to load the .qfs texture file of the file if you haven't done so through the Textures dialog.</p> <ul style="list-style-type: none">• Zoom In/Out Trackview: Zooms in/out only in the 3D track preview.
Show Virtual Road Bitmap Data Mode	Shows or hides the Virtual Road data of track polygons as color-coded polygons, it also shows AI opponent and traffic lines, alongside road boundaries.
Show Virtual Road Width and Node Points	<p>Shows or hides the Virtual Road nodes alongside their width.</p> <ul style="list-style-type: none">• Show speedfile for viewing: Choose the speedfile for forward (spdfa.bin) or reverse (spdra.bin) directions.
Show Block Direction	Shows or hides direction of block.
Show Track Based on Visibility	Toggles view where the current block and the visible blocks from it defined in the visibility file are highlighted while the rest are greyed out.
Show Current Block	Toggles view where currently selected block is highlighted with thicker lines.

Highlighted	
Show Polygon VRoad Flags	Shows or hides track polygon's Virtual Road Flags, alongside their direction.
Show Polygon Flags Color-Coded	Shows track polygon flags with color-coded lines and icons. <ul style="list-style-type: none"> • Show Poly Flag for Color Coding: select the poly flag types you want highlighted.
Re-center	Center view on the current selection
Refresh	Redraw screen

Toolbar command (View menu)

Use this command to display and hide the Toolbar, which includes buttons for some of the most common commands in T3ED. A check mark appears next to the menu item when the Toolbar is displayed.

See [Toolbar](#) for help on using the toolbar.

Status Bar command (View menu)


Use this command to display and hide the Status Bar, which describes the action to be executed by the selected menu item or depressed toolbar button, as well as basic status information. A check mark appears next to the menu item when the Status Bar is displayed.

See [Status Bar](#) for help on using the status bar.

Zoom In command (View menu)

This command zooms in the document view by a factor of two. A total of 9 zoom levels are available.


Shortcuts

Toolbar	Keys
	Mousewheel up

Zoom Out command (View menu)

This command zooms out the document view by a factor of two. A total of 9 zoom levels are available.


Shortcuts

Toolbar	Keys
	Mousewheel down

Translation Mode command (View menu)

This command selects translation mode. In this mode, the scroll bars behave in their usual way (i.e. they scroll the track view).

Shortcuts

Toolbar	Keys
	F9

Rotation Mode command (View menu)

This command selects rotation mode. In this mode, the scroll bars control the observer's view point rather than scrolling the view as usual. The horizontal scroll bar rotates the observer around the track along a horizontal circle, while the vertical scroll bar raises or lowers the viewpoint.

Tips

- As T3ED is slightly slower in rotation mode than in translation mode, rotation mode should not remain selected as the default working mode when not necessary.
- Raising the viewpoint to its top-most position can be very useful for optimal control of the Move XY tool. Conversely, lowering the viewpoint to the bottom-most position allows optimal control of the Move Z tool, but since it tends to produce a display containing lots of overlapping polygons, it is sometimes best to work with a slightly raised viewpoint.

Shortcuts

Toolbar Keys



Alt +
F9

Resolution submenu (View menu)

The Resolution submenu of the view menu can be used to select one of the three available resolutions for viewing the track polygons. The highest resolution is used in the game whenever the car becomes close enough, while the two other resolutions are used to save computing time when the car is still far away.

The lowest resolutions allow faster redrawing of the T3ED screen, but since some track vertices are not visible in these modes it is not recommended to work with them when moving track blocks (the highest resolution should always be used in order to ensure that the track structure is not messed up after a block is moved). Use of the low resolutions is also not recommended because it then becomes impossible to select a vertex which only belongs to a high-resolution polygon.

However, operations such as modifying polygon textures or deleting polygons must be performed in all three resolutions, and therefore require switching between the three available modes.

Full	Selects full resolution.
Half	Selects intermediate resolution.
Low	Selects low resolution.

Full Resolution command (View > Resolution menu)

This command selects the full resolution mode for viewing the track. Although this is the slowest mode, it is the only one in which all track vertices are displayed, and it is therefore strongly recommended to work in this mode when using the editing tools.

Operations such as modifying polygon textures or deleting polygons must be performed in all three available resolutions.

Shortcuts

Keys
Alt + 1

Half Resolution command (View > Resolution menu)

This command selects the half resolution mode for viewing the track. This mode is faster than the full resolution mode, but does not allow correct control of how the editing process affects the track structure.

Operations such as modifying polygon textures or deleting polygons must be performed in all three available resolutions.

Shortcuts

Keys
Alt + 2

Low Resolution command (View > Resolution menu)

This command selects the low resolution mode for viewing the track. This mode is the fastest, but does not allow correct control of how the editing process affects the track structure.

Operations such as modifying polygon textures or deleting polygons must be performed in all three available resolutions.

Shortcuts

Keys
Alt + 3

Show Objects command (View menu)

This command enables or disables display of the track objects. The object editing commands become unavailable when objects are not displayed.

NFS3/HS tracks contain several kinds of objects:

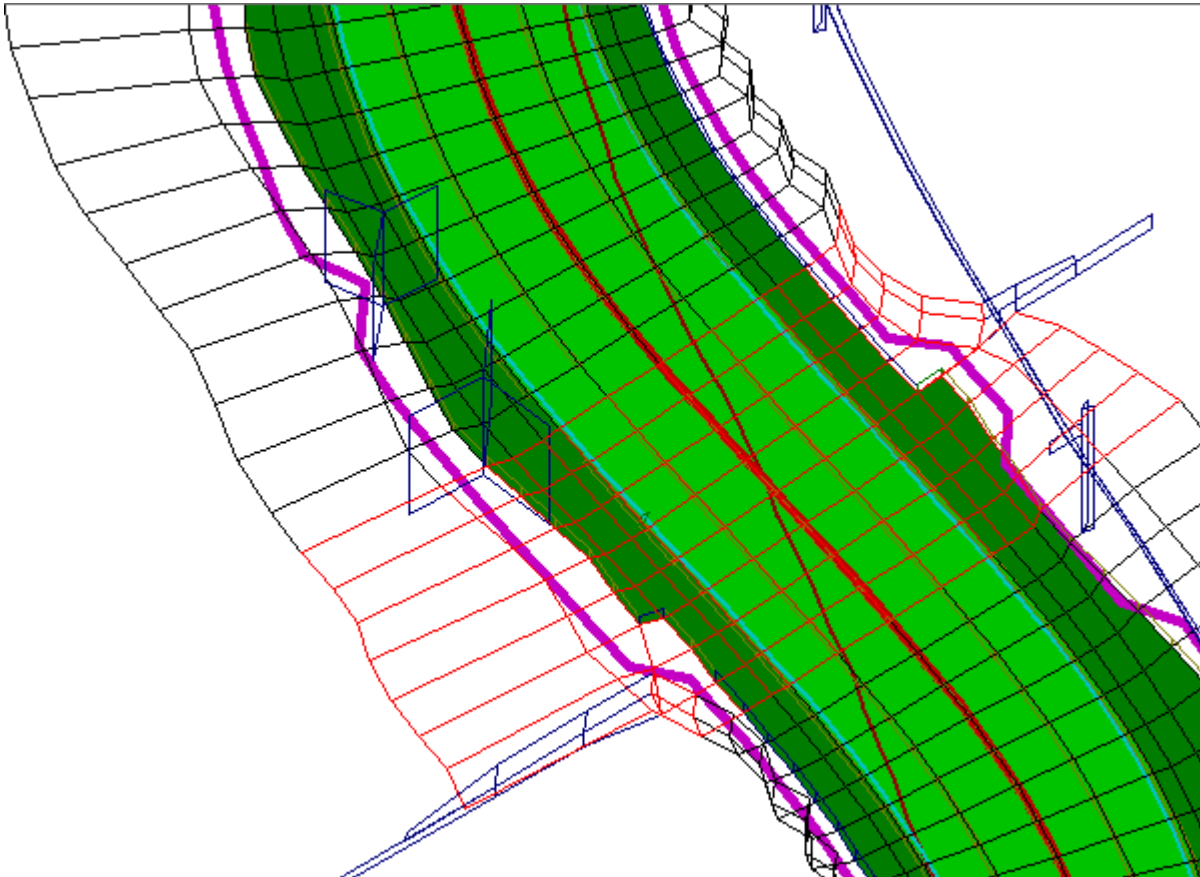
- **Polygon objects:** displayed in blue, are directly incorporated into the main track structure and are immovable.
- **Extra objects:** displayed in green, are the only ones which allow the detection of a collision in the game, with the effect of either stopping the car (e.g. a tree) or moving the object (e.g. a road sign).
- **Global objects:** displayed in teal, are usually animated objects (like the plane in Hometown); in NFSHS they are also used as objects cars can interact with (like cones or phone booths)
 - In order to edit global objects, you should convert them to extra objects first via the object's Properties dialog, then convert them back to global objects when you finish editing them.
- **Light sources:** displayed as dark yellow Xs, they are basically the light flares displayed in the track, usually in the form of street lighting. In NFSHS they can also be used as special effects (smoke, steam, etc).
- **Sound sources:** displayed as magenta speakers, they are used to deploy ambient sounds.

Show Lanes command (View menu)

This command enables or disables display of the polygons making up the lines between road lanes. The road lanes are displayed in dark yellow.

Show Virtual Road Bitmap Data Mode command (View menu)

Shows or hides the Virtual Road data of track polygons as color-coded polygons, it also shows AI opponent and traffic lines, alongside road boundaries.

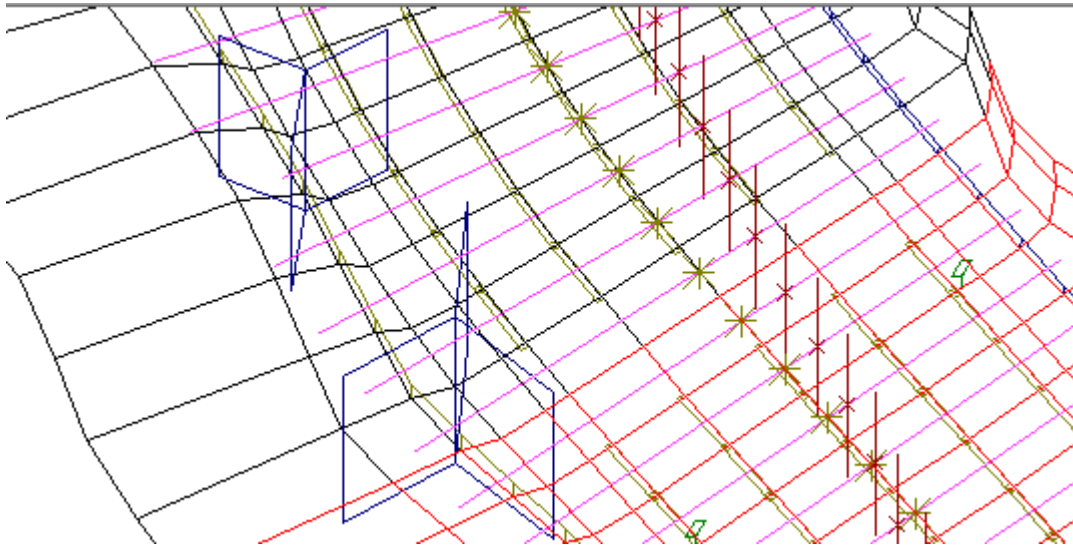


Legend

- **Green polygons:** drivable road-polygons
- **Dark green polygons:** drivable non-road polygons
- **Thick magenta lines:** VRoad width
- **Thick red line:** VRoad center
- **Dark red line:** Opponent AI lines
- **Cyan lines:** left/right lanes width

Show Virtual Road Width and Node Points command (View menu)

Shows or hides the Virtual Road nodes alongside their width.



Legend

- **Dark yellow asterisks:** Virtual Road points
- **Magenta lines:** Virtual Road points' width
- **Dark red X/lines:** Opponent AI lines and speed

Submenus

Show speedfile for viewing

Choose the speedfile for forward (spdfa.bin) or reverse (spdra.bin) directions.

Show Polygon Flags Color-Coded command (View menu)

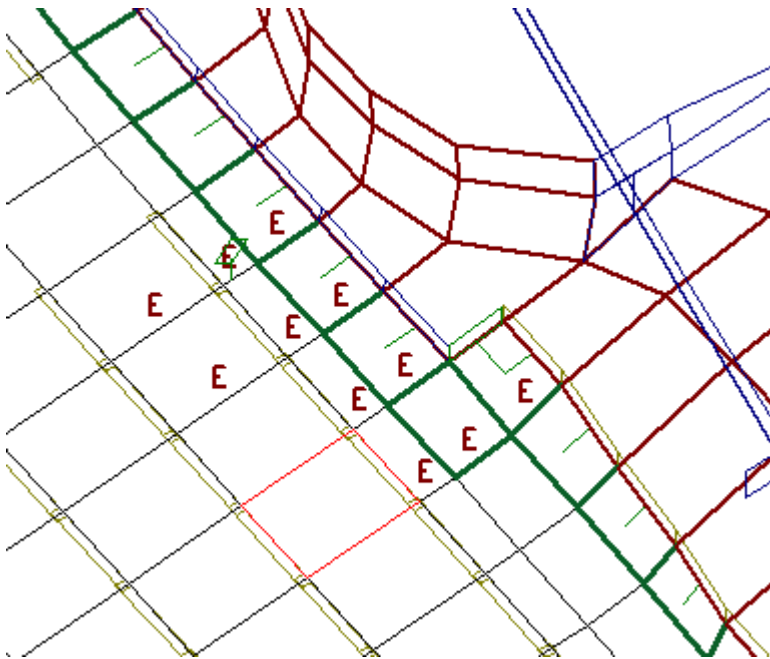
Shows track polygon flags with color-coded lines and icons. Flags displayed vary on options selected in the Show Poly Flag for Color Coding submenu

Submenus

Show Poly Flag for Color Coding

Toggles options for viewing polygon flags:

- **Virtual Road Boundary Polygons:** displays only polygon flags of road polygons.
- **Virtual Driving Road Edge Polygons:** displays only polygon flags of non-road polygons.
- **Non-passable polygons:** toggles highlighting of non-drivable polygons.



Legend

- **Thick forest green lines:** "Wall collision detection" flag
- **Dark red "E"s:** extra object collision detection
- **Green lines:** "No neighbor" flags
- **Thick dark red lines:** "Non-passable (outer)" flags

Re-center command (View menu)

This command recenters the view by placing the currently selected element in the middle of the T3ED window.

Shortcuts

Keys
Home

Refresh command (View menu)

This command causes redrawing of the T3ED window, and should be used in the event of a display problem.

Edit-Modes menu commands


The Edit-Modes menu offers the following commands:

Block	Selects block mode (the selection is a whole track block).
Point	Selects point mode (the selection is a single point).
Xtended Point	Selects extended point mode (the selection is a single point, with automatic smoothing).
Object	Selects object mode (the selection is an object).
Polygon	Selects polygon mode (the selection is a polygon).
Virtual Road Points	Selects Virtual Road points mode (the selection is a VRoad point). It will also enable the Show Virtual Road Width and Points option of the View menu
Smoothing	Modify the smoothing parameters for block mode or extended point mode.
Automatic object membership	If enabled, objects will change the block they belong to automatically upon moving them.
Select only from current block	If enabled, you will only able to select objects from the currently selected block; other blocks will be also disabled (greyed out)
Extra Smoothing	Enable or disable extra smoothing for the move tools in block mode.

Block Mode command (Edit-Modes menu)

This command selects T3ED's block mode. In this mode, the selected item is a track block, which can be moved around to reshape of the track. In order to avoid breaking the track when a block is moved, smoothing is automatically applied when using the Move XY and Move Z tools in block mode; the smoothing parameters can be adjusted by using the Smoothing command (Edit-Modes menu).


Shortcuts

Toolbar	Keys
	Ctrl + 1

Point Mode command (Edit-Modes menu)

This command selects T3ED's point mode. In this mode, the selected item is a single point, thus allowing precise control of the position of every piece of the track. Points belonging to different structures (objects, track blocks...) can sometimes be glued together: in that case, moving a point simultaneously affects all the other points attached to it. The Merge/split tool can be used to detach or attach points freely to each other.


Shortcuts

Toolbar	Keys
	Ctrl + 2

Xtended Point Mode command (Edit-Modes menu)

This command selects T3ED's extended point mode. The difference with point mode is that, in extended point mode, the Move XY and Move Z tools use smoothing in order to preserve the track structure. This mode is therefore appropriate for reshaping the track at a scale smaller than that of a block but without having to take care of every single point.

Shortcuts


Toolbar	Keys
	Ctrl + 3

Object Mode command (Edit-Modes menu)

This command selects T3ED's object mode. In this mode, the selected item is an object (either a polygon object or an extra-object). Object mode can be used only if objects are displayed (see Show Objects command in View menu). Objects can be moved, duplicated or deleted, and even created from scratch.

In any case, keep in mind that collision detection is only possible with extra objects.


Shortcuts

Toolbar	Keys
	Ctrl + 4

Polygon Mode command (Edit-Modes menu)

This command selects T3ED's polygon mode. In this mode, the selected item is a polygon, belonging either to the track itself or to an object. Polygons can be moved, duplicated or deleted. The texture applied to a polygon can be modified by using the Texture command (Edit-Tools menu). Also, when the selected polygon is a high-resolution track polygon, the Polygon flags command (Edit-Tools menu) can be used to modify its behavior when driven over by a car.

Shortcuts

Toolbar	Keys
	Ctrl + 5

Smoothing Properties command (Edit-Modes menu)

This command displays the Smoothing Properties dialog box, which can be used to control the smoothing parameters used by the Move XY and Move Z tools. This command is only available in block mode or in extended-point mode.

Smoothing Properties dialog box - Extended point mode

In extended point mode, moving a point affects the whole surrounding area in the following manner: all points which lie close enough to the selected point are moved together with it; the shape of the track is then adjusted over the "smoothing zone" in order to fit with the applied movement. The dialog box allows setting the respective sizes of these two zones by changing the "Width of zone to move" and "Width of smoothing zone" parameters. These parameters are expressed in arbitrary distance units; the length of a track block is about 50 units. The first distance is by default 0, i.e. smoothing begins immediately away from the selected point, which is the most useful setting in usual circumstances.

Smoothing Properties dialog box - Block mode

In block mode, T3ED can use two different smoothing methods: normal smoothing or "extra-smoothing". In normal smoothing mode, a certain portion of the track (by default slightly more than a block) is moved as is, without any deformation; the shape of the track over the smoothing zone is then adjusted in order to fit with the applied movement. Existing turns in the track are preserved by this adjustment: normal smoothing is therefore appropriate for adding new turns without removing the existing ones.

In extra-smoothing mode, the shape of the track is smoothened using polynomial interpolation, without retaining the original turns. All existing turns within the smoothing zone are removed, and replaced by a single bend which connects the new position of the moved block with the rest of the track. Extra-smoothing is therefore appropriate for deeper reshaping operations, such as replacing a sequence of sharp turns by a single wider turn, or even straightening a track portion.

In both cases, the size of the smoothing zone can be modified by using the "Width of smoothing zone" control: the width of the smoothing zone on each side of the selected block is expressed in units of track nodes (one track block corresponds to 8 track nodes). In normal smoothing mode, the size of the central zone which is moved without deformation can be modified using the "Width of zone to move" control (this parameter is not used in extra-smoothing mode where the smoothing always occurs over the whole smoothing zone). Also, note that smoothing only occurs along the selected movement directions: i.e., the Move XY tool does not act on the track altitudes, while the Move Z tool does not affect the XY coordinates.

Remarks

1. When moving track blocks, be very careful not to create a sharp turn in which the track polygons would overlap with each other; this would cause display problems in NFS. Once the track structure is disorganized, it is impossible to reobtain a normal structure without moving every single point back in place; the only reasonable option is therefore to immediately use the Undo command in order to cancel the changes. In order to avoid this situation, it is advisable to increase the width of the smoothing zone, especially when moving a track block over a long distance. Enabling extra-smoothing is also helpful in many cases. Note that the default smoothing parameters at T3ED startup are only appropriate for moving track blocks over relatively short distances.
2. When extra-smoothing is enabled, selecting a very large value for the "Width of smoothing zone" parameter makes it possible to very easily straighten the track (e.g. in order to make an oval). When the maximum value of 500 is chosen, moving a single track block immediately smoothen a huge track portion, so that moving two or three suitably chosen track blocks is normally sufficient to obtain a track without any sharp turn (oval, speed ring, etc...). Simply remember that, as the Move XY tool does not affect the altitudes, it is also necessary to repeat the process with the Move Z tool in order to obtain a flat track.

Shortcuts

Keys
Alt + S

Extra Smoothing command (Edit-Modes menu)

This command enables or disables the extra-smoothing option of the Move XY and Move Z tools in block mode. This command is only available in block mode.

In normal smoothing mode, a certain portion of the track is moved as is, without any deformation; the shape of the track over the smoothing zone is then adjusted in order to fit with the applied movement. Existing turns in the track are preserved by this adjustment: normal smoothing is therefore appropriate for adding new turns without removing the existing ones.

In extra-smoothing mode, the shape of the track is smoothened using polynomial interpolation, without retaining the original turns. All existing turns within the smoothing zone are removed, and replaced by a single bend which connects the new position of the moved block with the rest of the track. Extra-smoothing is therefore appropriate for deeper reshaping operations, such as replacing a sequence of sharp turns by a single wider turn, or even straightening a track portion.

Edit-Tools menu commands

The Edit-Tools menu offers the following commands:

Enter move distance	<p>Opens up the Enter move distance dialog, where you can move an element by inputting new coordinates to be moved by. Can be used on the following elements:</p> <ul style="list-style-type: none">• Track blocks• Track objects• Polygons• Points
Move XY	<p>Selects the Move XY tool: drag the mouse in order to move the selection along horizontal directions.</p>
Move Z	<p>Selects the Move Z tool: drag the mouse in order to move the selection along the vertical direction.</p>
Drop down on track	<p>Drops selected element to the nearest track polygon. Can be used on the following elements:</p> <ul style="list-style-type: none">• Track objects• Lane polygons• Points
Expand / Shrink	<p>Opens up the Expand / Shrink dialog where you can scale up/down the dimensions of an element. Can be used on the following elements:</p> <ul style="list-style-type: none">• Track blocks• Track objects
Invert	<p>Flips a polygon orientation. Can only be used in Polygon mode.</p>
Merge/Split	<p>Selects the Merge/Split tool: control-left-click in order to merge vertices or adjacent polygons?; right-click in order to split vertices (<i>This command has not yet been implemented for polygons</i>).</p>
New/Duplicate	<p>Create or duplicate an object or a polygon.</p>
Rotate	<p>Rotates the selected element. Can be used on the following elements:</p> <ul style="list-style-type: none">• Track blocks• Track objects• Track polygons<ul style="list-style-type: none">◦ You can rotate the texture orientation of a polygon by pressing Ctrl + Alt + R

Add Fence	Adds a fence polygon at the center of the track block
Add Lane	Adds a lane polygon at the center of the track block
Adjust road width	Opens a dialog to adjust the virtual road width to remove the invisible walls on either side of the track.
Virtual Road, heights, spdfa & spdra.	Opens up a dialog where you can edit the Virtual Road data point properties of the currently selected block such as AI speeds and lanes.
Polygon flags	Modify the virtual road properties of a track polygon (collision detection, passability...)
Properties	Display the properties of currently selected element.
Texture	View and modify the texture of the currently selected element. Can be used on the following elements: <ul style="list-style-type: none"> • Track blocks • Track objects • Track polygons
Visibility	Manages the track blocks visibility. <ul style="list-style-type: none"> • Set visibility: opens a dialog to set track visibility for track blocks • Export/Import visibility of track blocks to/from visibility.txt • Increase or decrease visibility of currently selected track blocks for forward and backward directions
Block Neighbours	Opens up a dialog to set up blocks connected to the currently selected block.
Export	Exports various kind of data to files <ul style="list-style-type: none"> • Track vertices to Vertices.txt • Texture values to Textures.txt • Export object (to .off file) <ul style="list-style-type: none"> ◦ Objects can be exported centered or with their positions in-track • NFS3 Texturebooks • Virtual Road, heights, spdfa and spdra data
Import	Imports various kind of data from files <ul style="list-style-type: none"> • Texture values from Textures.txt • Import object (from .off file) • NFS3 Texturebooks • Virtual Road, heights, spdfa and spdra data
Generate LODs	Experimental feature, generates LODs of track blocks

Enter move distance command (Edit-Tools menu)

Opens up the Enter move distance dialog, where you can move an element by inputting new coordinates to be moved by. Can be used on the following elements:

- Track blocks
- Track objects
- Polygons
- Points

The screenshot shows a dialog box titled "T3ED : Enter move distance" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Distance :** A group box containing three input fields for X, Y, and Z coordinates, each with a value of 0.
- Use from last move:** A button above three input fields, each with a value of 0.
- Use last distance:** A button above three input fields, each with a value of 0.
- No (extra-) smoothing:** A checkbox that is currently unchecked.
- Start Block :** An input field with a value of 0.
- End Block :** An input field with a value of 0.
- OK** and **Cancel** buttons at the bottom.

Distance

X/Y/Z

Coordinates you want to move your element by.

Others

Use from last move

Uses the coordinates from the last time you moved an element with the [Move X/Y](#) or [Move Z](#) tools or the the last time you used the Enter move distance command.

Use last distance

Uses the coordinates from the distance measured from the most recent first and last points you have selected.

No (extra) smoothing

For track blocks only. If enabled, it will disable smoothing for the resulting move command.

Start/End block

For track blocks only. Sets range of blocks to apply the tool to.

Shortcuts

Keys
Alt + M

Move XY command (Edit-Tools menu)

This command selects or deselects the Move XY editing tool. This tool can be used to move items in the horizontal (XY) directions. It is available in all editing modes (but must be reselected every time the editing mode is changed). When the Move XY tool is selected, an item can be moved by clicking on it and dragging the mouse while keeping the left button pressed.

In Point, Object and Polygon modes the behavior is straightforward. In Extended point and Block modes, smoothing is automatically applied in order to make the editing process easier. Also note that in Polygon mode the neighbouring polygons are modified so that moving the selected polygon does not disconnect it from the rest of the structure.

Remarks:

- In Block mode, two different smoothing methods are available: see [Extra-Smoothing command](#) for more information.
- In Object mode, moving an object along the track over a sufficiently long distance automatically moves the object from one track block to another. On the contrary, a vertex or a polygon keeps belonging to the same track block, no matter how far it is moved.

Note about moving track blocks:

When moving track blocks, be very careful not to create a sharp turn in which the track polygons would overlap with each other; this would cause display problems in NFS. Once the track structure is disorganized, it is impossible to reobtain a normal structure without moving every single point back in place? the only reasonable option is therefore to immediately use the Undo command in order to cancel the changes. In order to avoid this situation, it is advisable to increase the width of the smoothing zone, especially when moving a track block over a long distance. Enabling extra-smoothing is also helpful in many cases. Note that the default smoothing parameters at T3ED startup are only appropriate for moving track blocks over relatively short distances.

Notes about moving objects:

- Collisions with extra-objects (displayed in green) are only detected over specific track polygons. In order to allow a moved object to be detected, you must modify the track polygon properties accordingly. First choose the High-resolution view, and switch to Polygon mode. Select the track polygon over which the object lies; use the Polygon Flags command in the Edit-Tools menu, and check the "Extra-object collision detection" option. If the object lies across several polygons, repeat this operation for every polygon to enable object detection.
- Collisions with polygon objects are not handled by NFS; rather, the track polygons over which the object lies (or beyond) are marked as impassable for the cars. The procedure to change the polygon properties (passable or not, etc...) is the following: first chose the High-resolution view, and switch to Polygon mode. Select the track polygon, and use the Polygon Flags command in the Edit-Tools menu; choose the relevant option in the "Drive-over behavior" list to change the passable status of a polygon. Also note that collisions with impassable polygons are handled only if all neighboring polygons have the "Wall collision detection" flag set; normally this flag is automatically set by T3ED when a polygon is made impassable.

Shortcuts

Toolbar




Move Z command (Edit-Tools menu)

This command selects or deselects the Move Z editing tool. This tool can be used to move items in the vertical direction. It is available in all editing modes (but must be reselected every time the editing mode is changed). When the Move Z tool is selected, an item can be moved by clicking on it and dragging the mouse along the vertical axis while keeping the left button pressed.

In Point, Object and Polygon modes the behavior is straightforward. In Extended point and Block modes, smoothing is automatically applied in order to make the editing process easier. Also note that in Polygon mode the neighbouring polygons are modified so that moving the selected polygon does not disconnect it from the rest of the structure.

Remark:

In Block mode, two different smoothing methods are available: see [Extra-Smoothing command](#) for more information.

Shortcuts

Toolbar

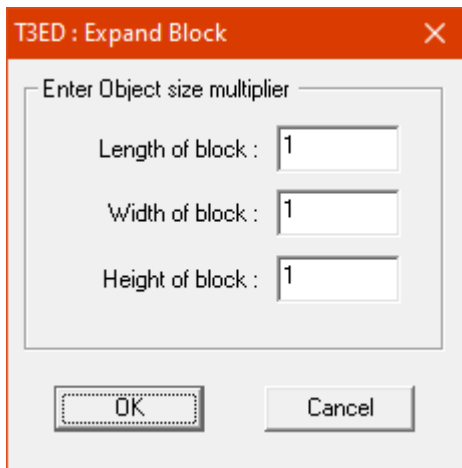



Expand / Shrink command (Edit-Tools menu)

Opens up the Expand / Shrink dialog where you can scale up/down the dimensions of an element. Can be used on the following elements:

- Track blocks
- Track objects

Track blocks



Enter Object size multiplier

Length/Width/Height of block

Scaling multipliers for the block's length, width and height. Values above 1 will scale it up, values below 1 will scale it down.

Track objects



Enter Object size multiplier

X/Y/Z-axis

Scaling multipliers for the object's X/Y/Z dimensions. Values above 1 will scale it up, values below 1 will scale it down.

Connect axis

If enabled, will lock the axis values to be the same as the X-axis in order to scale the object proportionally.

Alignment

Top/Middle/Bottom

Selects the pivot in which the object will scale towards.

Shortcuts

Keys
Ctrl + E

Merge/Split command (Edit-Tools menu)

This command selects or deselects the Merge/Split editing tool. This tool is only available in Point mode or in Polygon mode. ***It is currently only implemented for Point mode.***

This tool can be used to merge vertices together, or on the contrary to split a vertex in order to separate the various polygons which share it.

Merge function (Point mode):

This tool can be used to glue several vertices together in order to ensure that the polygons containing them are suitably connected. After this operation is performed, the structure will remain connected until the vertex is split back into pieces using the Split function: the merged vertices become, for all practical purposes, a single point. To merge two vertices, first select a point (using the left mouse button as usual), then press the Control key and click on another point (using the left mouse button): this point is then moved to the position of the first point, and the two points become merged.

Split function (Point mode):

This tool can be used to split a vertex, in order to disconnect the various polygons to which it belongs. This makes it possible to move some of these polygons independently of the others. In order to split a point, simply click on it with the right mouse button. The various polygon vertices which were merged then become separated (each created vertex is moved in the general direction of the center of the corresponding polygon). It is possible to use the Merge function in order to partially re-connect the split structure.

Remark:

Whenever possible, the Split function attempts to preserve the multi-resolution structure of the track blocks. Namely, when T3ED detects that two track polygons both involving the selected point and belonging to different resolutions actually contain each other, then these two track polygons remain connected in order to facilitate the track structure editing process.

Merge function (Polygon mode):

not yet implemented.

Shortcuts

Toolbar




New/Duplicate command (Edit-Tools menu)

This command can be used to duplicate the currently selected object or polygon. It is only available in Object and Polygon modes.

In **Polygon mode**, the New/Duplicate command is only available when a polygon is selected. It then has the effect of duplicating the selected polygon, preserving all its properties (texture, flags...). The newly created copy is slightly shifted, and is disconnected from all existing polygons so it can be easily moved. In order to facilitate the addition of new track portions, the multiple resolutions are automatically handled: namely, when a high-resolution track polygon is duplicated, copies of it are also created in medium and low resolutions; when a medium-resolution track polygon is duplicated, a copy of it is also created in low resolution. Therefore, track polygons always ought to be created in high resolution, before their medium and low-resolution versions are merged two by two in order to obtain a smaller number of polygons in medium and low-resolution. Also note the following consequence: the created polygons share the same vertices, so moving the high-resolution version also affects the two lower-resolution polygons created simultaneously, but any change of texture will need to be repeated three times.


Note: The correct procedure to add more polygons to the track structure is the following:

1. Switch to high-resolution view and polygon mode.
2. Select a track polygon reasonably similar to the ones you want to add, *in the same track block*.
3. Duplicate it as many times as needed using the Duplicate tool.
 - Duplicating the first of the last polygon of a row will add a new polygon before/after it instead of splitting it. If you want to split it like a polygon that's in the middle, hold Alt while duplicating.
4. Move all the new polygons around until their general disposition is satisfying. Beware of not flipping a polygon upside down, because the two sides of a polygon are not at all equivalent.
5. Switch to point mode and select the Split/Merge tool.
6. Merge the vertices of the new polygons with the existing frame by successively clicking on an existing vertex and control-clicking on all new vertices to be attached to it.
7. Switch to half-resolution view; in order to speed up the in-game display, merge pairs of adjacent polygons to form larger polygons (since the Merge tool for polygons has not yet been implemented in this version of T3ED, this is not yet possible: an alternative is to delete some unimportant polygons from the half-resolution view).
8. Repeat step 7 in low-resolution view.

In **Object mode**, the New/Duplicate command is always available. If an object is selected, then this object is duplicated. The newly created object has exactly the same properties as the original object, and lies at exactly the same place; the new copy will therefore have to be moved elsewhere.

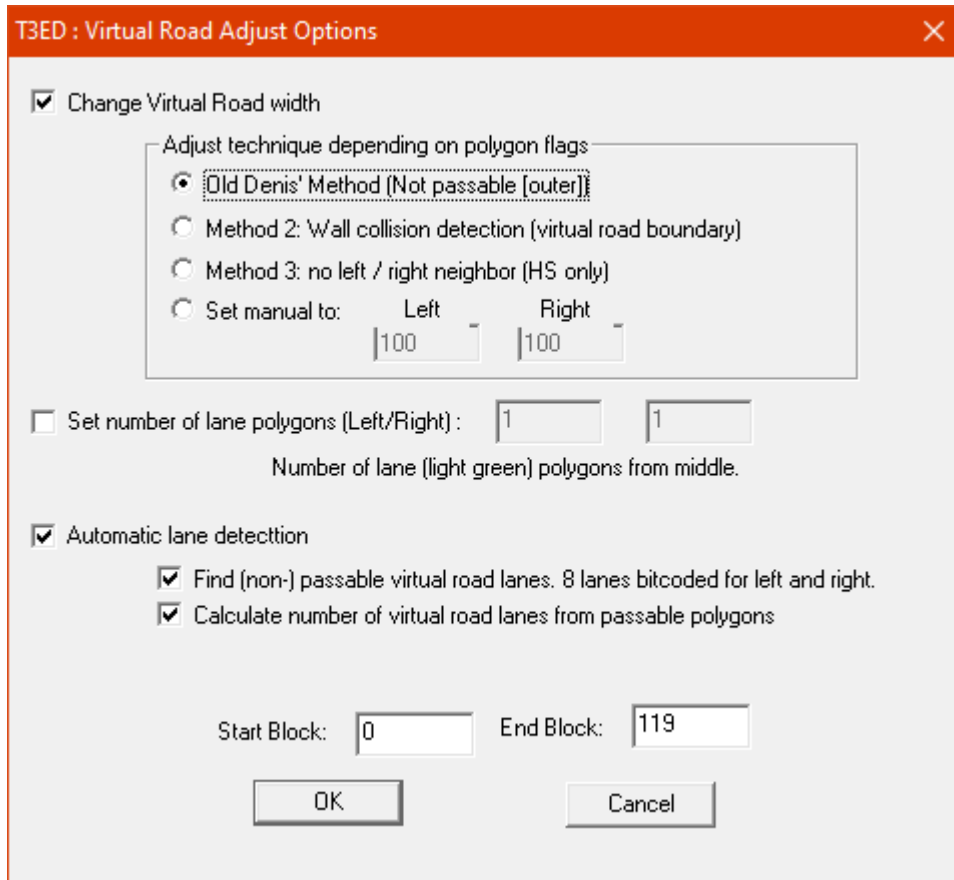
If no object is selected, the New/Duplicate command attempts to create a new object in the currently selected block. This object initially consists of a single rectangle; more polygons can later be added by using the Duplicate command in Polygon mode together with the Merge tool in Point mode. Before creating a new object, T3ED displays a dialog box which lets you choose the object type, as well as the texture of the created object polygon.

Shortcuts

Toolbar	Keys
	Ctrl (+Alt) + D -or- Ctrl (+Alt) + N

Adjust Road Width command (Edit-Tools menu)

This command is used to adjust the width of the virtual road in order to get rid of the invisible walls restraining the car's movements on either side of the road (the track polygons also need to be made passable using the Polygon Flags command). When you use this command, this dialog appears:



The screenshot shows a dialog box titled "T3ED : Virtual Road Adjust Options" with a close button (X) in the top right corner. The dialog contains several options and input fields:

- ☒ Change Virtual Road width
 - Adjust technique depending on polygon flags
 - ☒ Old Denis' Method (Not passable [outer])
 - ☐ Method 2: Wall collision detection (virtual road boundary)
 - ☐ Method 3: no left / right neighbor (HS only)
 - ☐ Set manual to: Left: Right:
- ☐ Set number of lane polygons (Left/Right) :
Number of lane (light green) polygons from middle.
- ☒ Automatic lane detection
 - ☒ Find (non-) passable virtual road lanes. 8 lanes bitcoded for left and right.
 - ☒ Calculate number of virtual road lanes from passable polygons
- Start Block: End Block:
- OK Cancel

Change Virtual Road Width

If enabled, it will adjust the road width with the selected method below

- Old Denis' Method: method used in old T3ed versions, will set VRoad width bound by polygons flagged as "non-passable (outer)"
- Wall collision detection: uses wall collision polygon flags to set VRoad width
- No left/right neighbor: for NFSHS tracks only; will use "no neighbor" polygon flags to set VRoad width.
- Set manual: sets VRoad width based on left/right unit values you provide.

Set number of lane polygons (left/right)

If enabled, it will set lane polygons (shown as light green if Bitmap VRoad mode is enabled) from the VRoad center based on the values you provide.

Automatic lane detection

If enabled, T3ed will attempt to detect lanes automatically. It also has two toggleable options:

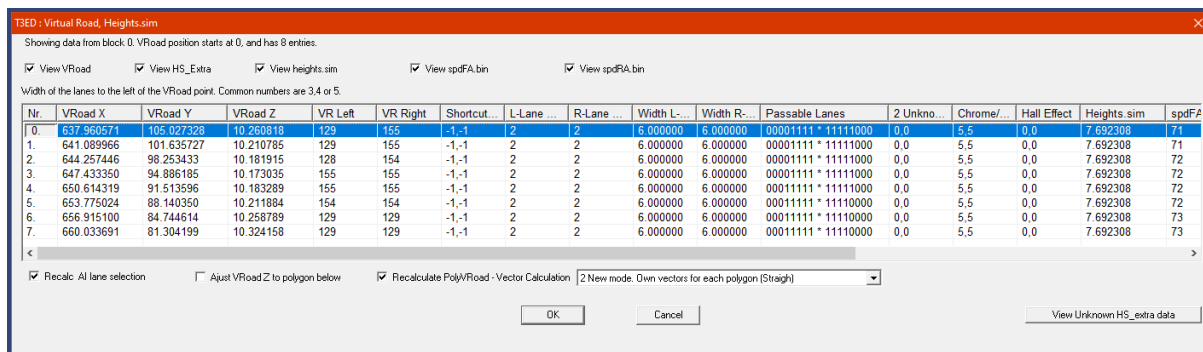
- Find (non-)passable virtual road lanes
- Calculate number of virtual road lanes from passable polygons.

Start & End Blocks

Determines the range of blocks in which the command will be applied. Default is 0 (first block) for Start Block and total block amount (as seen in File menu > Properties) - 1 (last block) for End Block.

Virtual Road, Heights.sim, etc. (Edit-Tools menu)

Opens up a dialog where you can edit the Virtual Road data point properties of the currently selected block such as AI speeds and lanes.



View VRoad/HS_Extra/heights.sim/spdFA.bin/spdRA.bin

Toggle the types of data displayed below

- **VRoadX/Y/Z:** coordinates of the VRoad point.
- **VR Left/Right:** left/right edges of the Virtual Road (measured from the VRoad point). Can also be changed via [Adjust Virtual Road](#).
- **Shortcut to:** shortcut from the VRoad point to others. Two VRoad point numbers can be specified. If no shortcut is specified, it should be set to -1,-1.
- **L-Lane/R-Lane Polys:** number of lanes to the left/right of the VRoad point.
- **Width L-Polies/R-Polies:** width of the lanes to the left/right of the VRoad point. Common values ranges from 3 to 5.
- **Passable Lanes:** displays lanes used by opponents in bits
 - 1 corresponds to a passable lane, 0 to a non-passable lane.
 - * is the VRoad point
 - Exactly 8 numbers must to be specified for left and right
- **2 Unknown Bytes:** unknown bytes. Usually both are 0.
- **Chrome/Weather:** chrome (reflections) effect for cars in left/right lanes from the VRoad point. Usual values are:
 - 0: no chrome
 - 2: low chrome, not affected by weather (usually seen in tunnels)
 - 4: regular chrome, affected by weather
- **Hall effect:** hall (echo) effect for audio in left/right lanes from the VRoad point. Usual values are:
 - 0: disabled
 - 3-4: enabled
 - 6: underwater tunnel (like Aquatica's)
- **Heights.sim:** value in the Heights.sim file of the VRoad point.
- **spdFA/spdRA Speed:** maximum speed (in m/s) AI opponents can reach for that VRoad point in forward/backward direction.

- **spdFA/spdRA AI Lane selection:** sets opponents preferred lanes in forward/backward direction. Usual values are:
 - 96 for left of VRoad point
 - 112 for right of VRoad point
 - Values must be divisible by 16
- **spdFA/spdRA float:** indicates points of ideal line in forward/backward direction. AI Lane Selection takes higher priority than this.

Recalc AI Lane selection

If enabled, recalculates AI Lane selection values.

Adjust VRoad Z to polygon below

If enabled, it will drop VRoad points to the nearest road polygon below.

Recalculate PolyVRoad - Vector Calculation

If enabled, recalculates polygon Virtual Road flags with the specified settings from the drop down:

- Old mode (per polygon)
- One block vector, based on VRoad points for all polygons
- New mode. Own vectors for each polygon (straight)

View UnknownHS_extra data

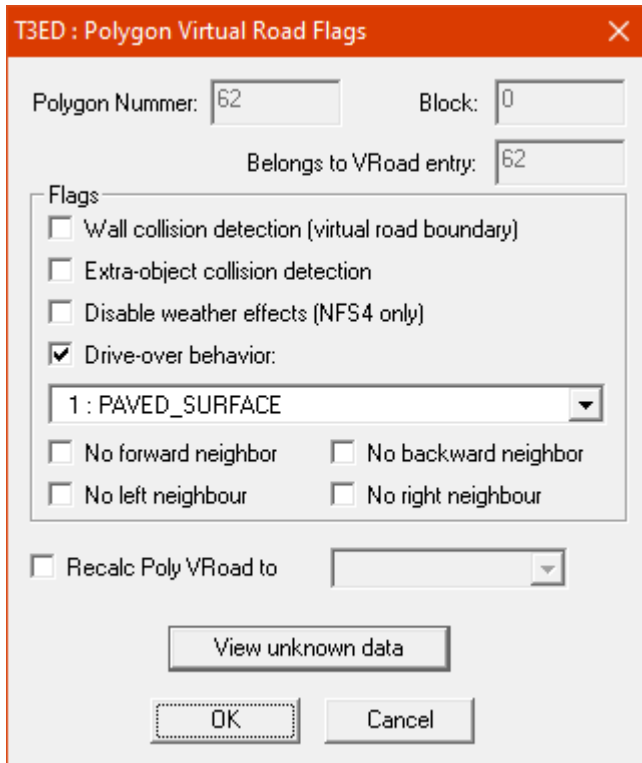
Opens a dialog with the unknown data of the VRoad points. **Only use it if you know what you're doing.**

Shortcuts

Keys
Ctrl + H

Polygon Flags command (Edit-Tools menu)

This command can be used to modify the behavior of a high-resolution track polygon when a car drives over it. It is only available in Polygon mode and when a high-resolution track polygon is selected. T3ED then displays the Polygon Virtual Road Flags dialog box, which lets you modify the following settings:



T3ED : Polygon Virtual Road Flags

Polygon Number: 62 Block: 0

Belongs to VRoad entry: 62

Flags

- ☐ Wall collision detection (virtual road boundary)
- ☐ Extra-object collision detection
- ☐ Disable weather effects (NFS4 only)
- ☒ Drive-over behavior:
1 : PAVED_SURFACE
- ☐ No forward neighbor
- ☐ No backward neighbor
- ☐ No left neighbour
- ☐ No right neighbour

☐ Recalc Poly VRoad to

View unknown data

OK Cancel

Flags

- **Wall collision detection (virtual road boundary):** this option must be checked if the polygon is passable and any of the neighboring track polygons is not passable. Otherwise, NFS3/HS is unable to detect any collision with an invisible wall and the car will be free to move into a neighboring impassable polygon.
- **Extra-object collision detection:** this option must be checked if the polygon is passable and if any extra-object lies nearby. Otherwise, NFS3/HS is unable to detect any collision with an extra-object and the car will be free to move across trees and other obstacles.
- **Disable weather effects (NFS4 only):** disables effects such as wet roads in the selected polygons, which is seen in covered-up areas like tunnels.
- **Drive-over behavior:** Choose an item in the list in order to describe how cars should behave when they pass over the polygon. Values 0 and 14 correspond to impassable polygons, while all others correspond to passable polygons. This list

also lets you choose whether grass, snow, dust... should be projected when a car passes.

- **NOTE:** If you wish to make passable polygons which previously weren't passable, it is likely that you will have to adjust the width of the virtual road in order to get rid of the invisible walls restraining the car's movements on either side of the road: for this, use the Adjust Road Width command in the Edit-Tools menu.
- **No neighbor flags:** indicates if the polygon is besides a wall polygon in order to apply collisions correctly. NFS3 only has no-left and no-right neighbour flags; NFSHS has no-forward and no-backward flags in addition.

Recalc Poly VRoad to

Sets VRoad polygon direction

View unknown data

View additional polygon data that hasn't been documented yet. **Only use it if you know what you're doing.**

Shortcuts

Keys

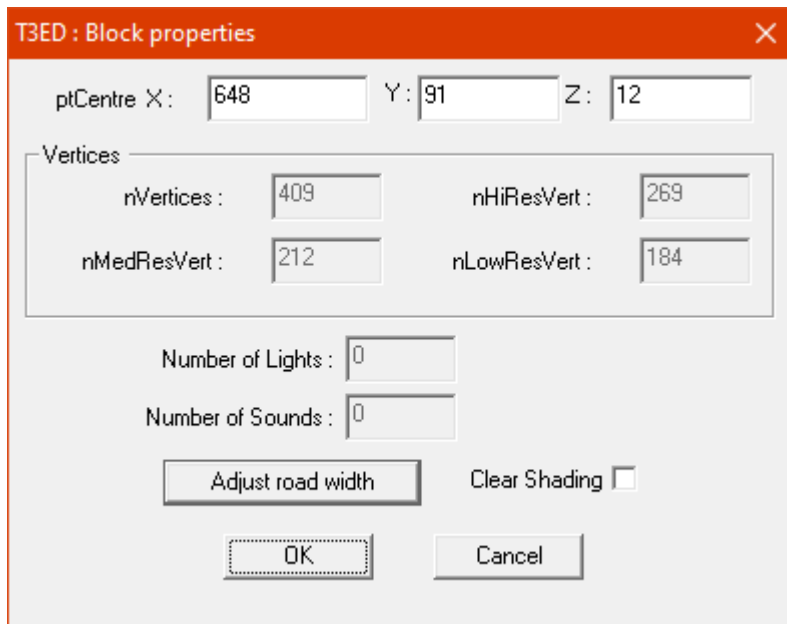
Ctrl + Enter

Properties command (Edit-Tools menu)

Opens up a dialog showing the properties of the currently selected element.

- [Track block properties](#)
- [Point properties](#)
- [Polygon object properties](#)
- [Extra/global object properties](#)
- [Light/sound source properties](#)
- [Replay cameras](#)
- [Polygon properties](#)
- [Virtual Road point properties](#)
- [Shortcuts](#)

Track block properties



T3ED: Block properties

ptCentre X: 648 Y: 91 Z: 12

Vertices

nVertices: 409 nHiResVert: 269

nMedResVert: 212 nLowResVert: 184

Number of Lights: 0

Number of Sounds: 0

Adjust road width Clear Shading ☐

OK Cancel

ptCentre

Displays the coordinates of the center of the track block

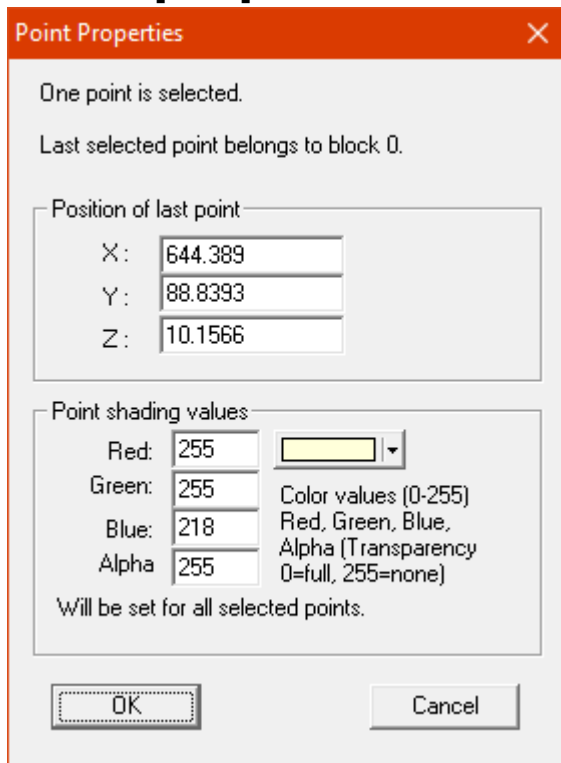
Adjust road width

Opens up the [Adjust Road Width Options dialog](#) for that track block

Clear Shading

Deletes the shading of all the track polygons

Point properties



The dialog box has an orange title bar with the text "Point Properties" and a close button (X). The main area is light gray and contains the following elements:

- Text: "One point is selected."
- Text: "Last selected point belongs to block 0."
- Section: "Position of last point" with three input fields:
 - X: 644.389
 - Y: 88.8393
 - Z: 10.1566
- Section: "Point shading values" with four input fields and a color swatch:
 - Red: 255
 - Green: 255
 - Blue: 218
 - Alpha: 255A color swatch showing yellow is next to the Red field.
- Text: "Color values (0-255)
Red, Green, Blue,
Alpha (Transparency
0=full, 255=none)"
- Text: "Will be set for all selected points."
- Buttons: "OK" and "Cancel"

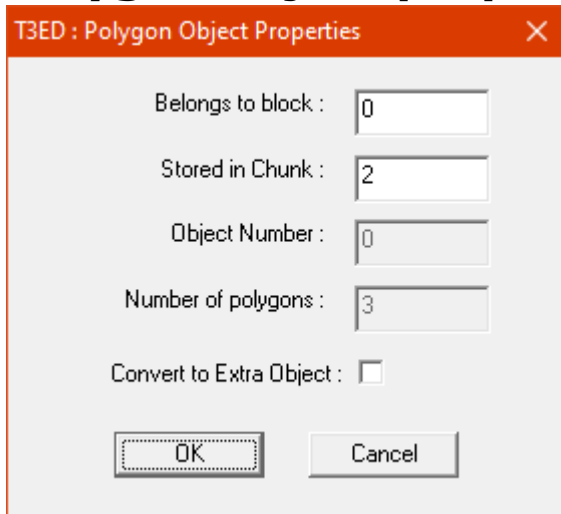
Position of last point

The coordinates of the selected point

Point shading values

Color and transparency values of the selected point(s), in RGBA format (with 0-255 scale)

Polygon object properties

The image shows a screenshot of a software dialog box titled "T3ED : Polygon Object Properties". The dialog has a light gray background and a red title bar with a close button (X) in the top right corner. Inside the dialog, there are five labeled input fields: "Belongs to block :" with a text box containing "0", "Stored in Chunk :" with a text box containing "2", "Object Number :" with a text box containing "0", and "Number of polygons :" with a text box containing "3". Below these fields is a checkbox labeled "Convert to Extra Object :" which is currently unchecked. At the bottom of the dialog are two buttons: "OK" and "Cancel".

T3ED : Polygon Object Properties

Belongs to block : 0

Stored in Chunk : 2

Object Number : 0

Number of polygons : 3

Convert to Extra Object : ☐

OK Cancel

Belongs to block

Block where the object belongs

Stored in Chunk

Each object is stored in a "chunk" of the object track data; chunks can determine how and where objects appear. For objects, the most commonly used chunks are:

- **Chunk 0 & 2:** used for regular polygon objects like buildings, trees and other sorts of static scenery outside of the driving road boundaries.
- **Chunk 3:** used for objects that only appear when the track is set to day with no weather (such as the wooden bridge light rays in Hometown)

Convert to Extra Object

Converts the selected polygon object to an extra object

Extra/global object properties

T3ED: Extra Object Properties

General

Object Refpoint position X: -432.924 Y: 339.946 Z: 1.87819

Change Refpoint ☐ X & Y = Middle | Z = Middle

Belongs to block: 75 Stored in chunk: 2

Object number: 0 Number of polygons: 28

Unknown: 0 Unknown 3: -244

Crosstype: 3 - Animated

Shading

0000=R:114 G:102 B:124 A:255

R: 114 G: 102 B: 124 A: 255 Set RGBA for all

Color values (0-255) Red, Green, Blue, Alpha (Transparency 0=full, 255=none)

HS Type 6

Type 6 position X: 0 Y: 0 Z: 0

Mass: 0 Byte 0-3

Hitbox X: 0 Y: 0 Z: 0 40-51

Type 6 Data: Export Import View

REF/OBJ Data

Unknown 1: 0

Unknown 2: 0

Collide effect: 0

0 - nothing, 1 - solid (trees, walls)
2 - Hit & fall (sions)

AnimData

Animation Memory: 348 Export

Animation Delay: 6 Import

Animationdata Nr: Num 0

Pos. X: -432.924 Pos. Y: 339.946 Pos. Z: 1.87819

Multi 1: 0 Multi 2: 0 Multi 3: 0 Multi 4: 16384

Convert to Polygon Object: ☐ Convert to Global Object: ☐

OK Cancel

General

Obj refpoint position:

Displays the object's coordinates; enable **Change refpoint** checkbox and set it to "Manual change" to change them

Belongs to block

Block where the object belongs

Stored in Chunk

Each object is stored in a "chunk" of the object track data; chunks can determine how and where objects appear. For objects, the most commonly used chunks are:

- **Chunk 0 & 2:** used for objects like signs and trees the player can collide with or global animated objects.
- **Chunk 1:** used for global objects with complex behavior, like cones.

Unknown & Unknown 3

Unknown data

Crosstype

The type of extra/global object

1. Global
2. Simple (NFSHS only)

3. Animated
4. Collision
5. unused
6. Complex behavior

REFXOBJ Data

Unknown 1 & 2

Unknown data

Collide effect

Selects collision type

0. no collision
1. solid (i.e. trees or walls)
2. Hit & fall (i.e. signs or gates)

AnimData

Animation Delay

Delay between animation frames in game tics (1/30th of a second), the shorter the delay between frames, the faster the animation.

Export/Import

Exports or imports animation data of object to/from a text file.

AnimationDataNr

Animation frame to edit

Pos. X/Y/Z

Coordinates of the object in the selected animation frame

Multi 1-4

XYZW rotations of the object, units are in degrees and they are represented as quaternions but with signed integer numbers, so you'll have to do a couple of conversions detailed below:

- You can use this 3D Rotation Calculator (<https://www.andre-gaschler.com/rotationconverter/>) to convert Euler XYZ rotation angles to quaternions. Just make sure both input and output units are set to degrees.
- Copy each of the quaternion values and multiply them by 16384, input the resulting numbers in the corresponding fields
- You cannot input negative values manually in the Multi fields but you can copy and paste them
- Euler XYZ rotation axis work as the following:
 - X will rotate the object up and down

- Y will rotate the object left and right
- Z will tilt the object left and right

Please note in order to apply changes for a AnimationDataNr Pos or Multi values, you'll have to select another one after changing them.

Shading

- The drop down list contains a list of the points of the object
- Below there's a color selector and RGBA (0-255) fields to determine the selected point color and transparency
- The **Set RGBA for all** button sets the chosen color for all points in the object

HS Type 6

As denoted, this is exclusive of NFSHS; Type 6 global objects do have mass and a hitbox; they can be interacted with by vehicles.

Mass

Weight of the object in kg

Export/Import

Exports or imports the Type 6 data of the object to/from a file

View

View the data of the Type 6 object directly. **Only use this option if you know what you're doing**

Light/sound source properties

The screenshot shows a dialog box titled "T3ED : Light and Sound Source Properties" with a close button (X) in the top right corner. Inside the dialog, there is a text box with the instruction: "Set light/special effect source data here. Numbers from 0 to 31 are usually lights, that can be selected by the combobox. Look at Tr(N).ini, [track glows] section. Higher numbers are special effects." Below this, there is a label "Belongs to block :" followed by a text box containing the value "63". Underneath is a combobox showing "glow 1, intensity 185". Below the combobox are four pairs of input fields: "Type : 1" and "Type (HEX) : 1", "Byte 2 : 0" and "Byte 2 (HEX) : 0", "Byte 3 : 255" and "Byte 3 (HEX) : FF", and "Byte 4 : 255" and "Byte 4 (HEX) : FF". To the left of the Byte 2 and Byte 3 fields is the text "1-255 = invisbile in NFS". Below these fields is a label "Light distance multiplier:" followed by a text box containing "0.1" and a button labeled "Raytrace Light". At the bottom, there is a section labeled "Position" containing three text boxes for "X:", "Y:", and "Z:" with values "-518.536", "15.3701", and "7.10045" respectively. At the very bottom are "OK" and "Cancel" buttons.

Belongs to block

Block where the object belongs

Type

- **Lights:** selects a track glow type from the ini file. You can also use the drop down above to preview its color and select it. You can see info on track glows [here](#)
- **Sounds:** selects a sound from the track's sound bank (tram#[N][W].bnk) file

Byte 2/3/4

Unknown data. **Don't touch it unless you know what you're doing.**

Raytrace light (light only)

Performs light raytracing on the adjacent track geometry using the light source and its [glow settings](#). Light radius can be adjusted with the Light distance multiplier field.

Position

Coordinates of the light object

Replay cameras

Will open the [Replay Camera Editor](#) for the selected camera.

Polygon properties

Polygon Shading Colors

Point shading values

Red: 255

Green: 255

Blue: 255

Alpha: 255

Color values (0-255)
Red, Green, Blue,
Alpha (Transparency
0=full, 255=none)

Will be set for all selected polygons

OK Cancel

Point shading values

Color and transparency values of the selected polygons' points, in RGBA format (with 0-255 scale)

Virtual Road point properties

Will open up the Virtual Road, heights.sim, spdfa and spdra dialog for that VRoad point

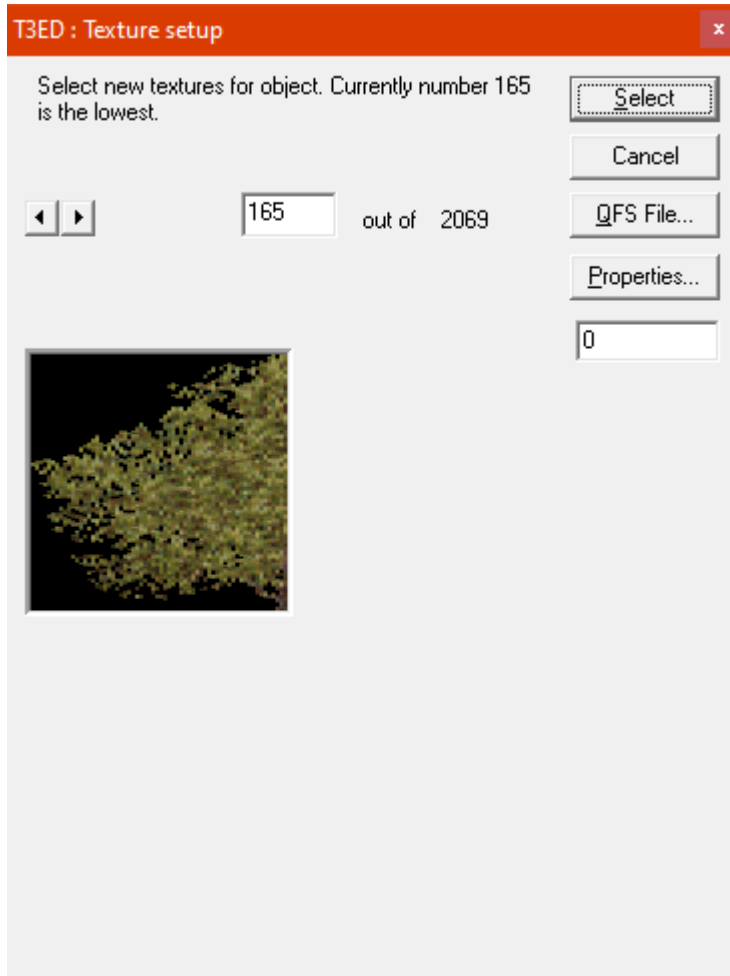
Shortcuts

Keys

Alt + Enter

Texture command (Edit-Tools menu)

- [Remarks](#)
- [Texture properties](#)
- [Shortcuts](#)



This command makes it possible to choose the texture of the currently selected polygon. It is available in Block, Object and Polygon modes. The first time this command is used after loading a track file, T3ED prompts for a QFS texture file to load; the QFS file-load dialog box lets you choose the appropriate texture file.

- **For NFS3:** there are usually four texture files in each track directory: the ones with names ending in `_8` are in 8-bit color mode, while the others are in 16-bit color mode; the filenames containing a `1` correspond to night mode, while the others correspond to day mode.
- **For NFSHS:** there are either one (tr0.qfs) or two (tr0.qfs and trn0.qfs) texture files. tr0 is the texture file for the daytime track while trn0 is the texture file for

the nighttime track.

Loading the textures is a slow process (about 10 seconds), so T3ED displays a progress bar while uncompressing and decoding the selected texture file. Choosing a 8-bit color mode texture file speeds up the process significantly without altering the quality of the displayed textures. The only exception is the Summit track, where the 8-bit textures are coded using a specific compression format which cannot be decoded by T3ED; an error message indicating that the QFS file has an incorrect format will then be displayed, and textures will not be displayed until you manually select another QFS file from within the texture chooser.

After the track textures are loaded, T3ED displays the Texture Chooser dialog box: this dialog box displays a miniature view of the current texture of the selected element. This texture can be changed by acting on the scrollbar or by manually entering a texture number. The "Select" button can be used to save the changes made to the polygon texture and close the Texture Chooser; use the "Cancel" button instead if you don't want the changes to be saved. The "QFS File..." button can be used to reload a different QFS texture file.

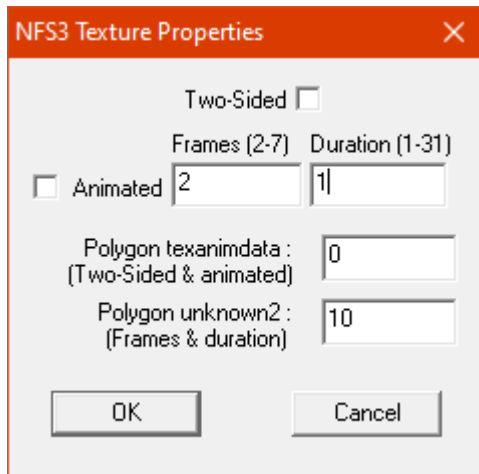
Remarks

- Display quality is extremely poor when the currently selected screen resolution has a 8-bit color depth or lower; 15-bit color depth or higher is necessary for properly viewing the textures.
- The texture numbers displayed in the Texture Chooser are not the same as the texture numbers in the QFS file: a single QFS texture can correspond to several different track textures depending on how it is oriented, tiled, etc... This also explains why different texture numbers can yield the same miniature view.
- When the QFS file cannot be loaded, the textures are not displayed (except for road-lane textures which are not contained in the QFS file and will always remain displayed). To display textures you must then manually select a QFS file by clicking on the "QFS File..." button in the Texture chooser dialog box.

Texture properties

Clicking on the "Properties" button of the Texture dialog will bring up a dialog, which depends on the game format:

NFS3:



Two-sided

If enabled, it will make the polygon visible in both sides.

Animated

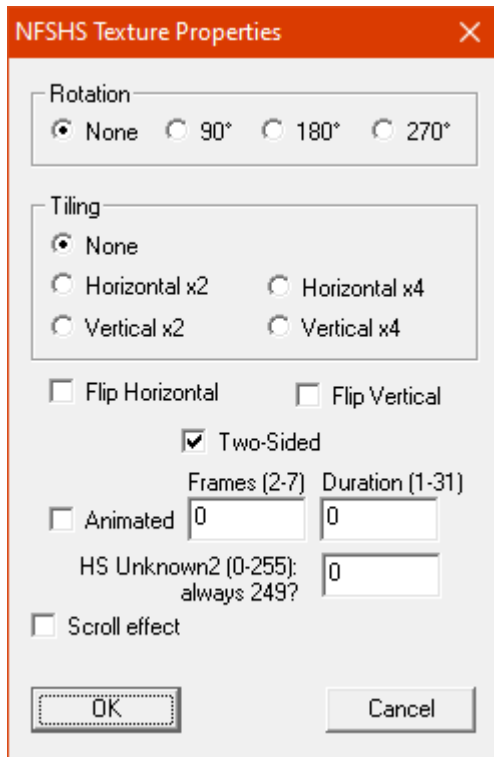
If enabled, it will play an animation based on the Frames and Duration fields.

- **Frames:** how many frames the animation will have (2 minimum, 7 maximum). Frames to use will be counted from the texture you selected.
- **Duration:** frame delay, measured in game tics (1/30th of a second)

Polygon texanimdata & Unknown

Probably displays the current texture flags info, leave them as it is.

NFSHS:



Rotation

Sets rotation of texture in 90 degree steps.

Tiling

Sets tiling (how many times it is repeated in the polygon) of the texture.

Flip horizontal/vertical

If enabled, will flip the texture horizontally or vertically, respectively.

Two-sided

If enabled, it will make the polygon visible in both sides.

Animated

If enabled, it will play an animation based on the Frames and Duration fields.

- **Frames:** how many frames the animation will have (2 minimum, 7 maximum). Frames to use will be counted from the texture you selected.
- **Duration:** frame delay, measured in game tics (1/30th of a second)


HSUnknown2

Unknown data, leave it as it is.

Scroll effect

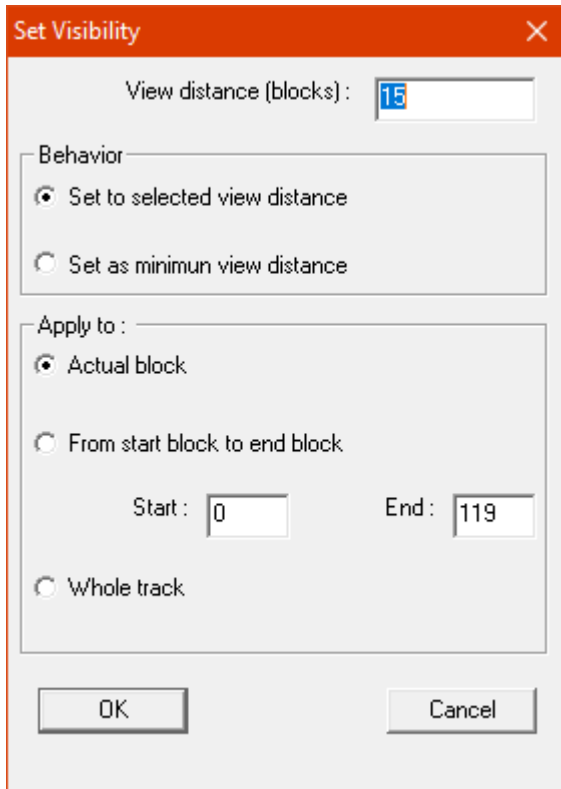
If enabled, the texture will scroll like a marquee.

Shortcuts

Toolbar	Keys
	Ctrl + T

Set Visibility command (Edit-Tools > Visibility menu)

Opens a dialog to set track visibility for track blocks



View distance (blocks)

Sets how many blocks can be seen from the selected block(s).

Behavior

Set to selected view distance

Set block's view distance to the amount specified in View distance.

Set as minimum view distance

Set block's view distance to the amount specified in View distance as a minimum value for visibility.

Apply to

Actual block

Applies tool to currently selected block.

From start block to end block

Applies tool to range of blocks you specify in the Start and End fields.

Whole track

Applies tool to the entire track.

Block Neighbours command (Edit-Tools menu)

Opens up a dialog to set up blocks connected to the currently selected block.

T3ED : Track Block Neighbours

1	-1	-1
119	-1	-1
-1	-1	

Add here numbers of track blocks directly connected to the selected block. For empty entry, leave "-1". Remember that at least Next and previous blocks must be here.

☐ Connect close points to near points of neighbour blocks

if distance is less than :

OK Cancel

Usually only the first and second values of the first column are used; first is the block following the currently selected block and the second is the preceding block.

Connect close points to near points of neighbour blocks if distance is less than:

If enabled, it will connect points closest to the ones of neighbour blocks if the distance is less than the specified one.

Whole Track menu commands

The Whole track menu offers the following commands:

Change all textures	Changes textures for all the track by the offset provided. (e.g. a value of 5 will set every texture of the track by the next 5th texture from its original texture)
Clear / Modify all	Clear selectively track shape, objects and textures. Can also recalculate Virtual Road data of the whole track.
Remap objects to blocks	Reassigns all objects' ownership (sans global objects) to the blocks they are placed on
Resize	Scales up/down the track based on the multipliers you input. Values above 1 scales it up, values below 1 scales it down)
Replay Camera Editor	For NFSHS tracks only. Opens up the Replay Camera Editor, which allows for editing replay camera angles of the track.
Shadow Ray Tracer	Opens up the Shadow Ray Tracer dialog, a tool to create track shading.

Clear / Modify all command (Whole Track menu)

This command can be used to erase selectively the track's shape, objects or textures; can be used to the point of removing virtually everything from a track, making it possible to restart almost from scratch.

T3ED : Clear / Modify all...

☐ Track shape (X & Y)
X multiplier : Y multiplier :
☐ Create additional curves :
Number of Curves (2 to ?) : Size :
Number of Curves (2 to ?) : Size :

☐ Road elevations
Set road height to :
☐ Create additional hills & valleys:
Set number (2 to ?) : Size :
Set number (2 to ?) : Size :

Delete
☐ Scenery elevations
☐ Lanes & fences
☐ Sound sources
☐ Polygon objects & light sources
☐ Extra objects
☐ Global animated objects (can't undo)
☐ Clear All Shading Values. (No Shadows)

Speedfiles
☐ Forward Speedfile (spdfa.bin) ☐ Backward Speedfile (spdra.bin) Start: End:
☐ Set float values to zero ☐ Recalculate zero float values (straight) ☐ Recalculate zero float values (curve)

☐ Road textures : set to Browse...
☐ Scenery textures : set to Browse...
☐ Cut FRD-file to Blocks:
☐ Set default block neighbours
☐ Recalculate PolyVRoad
☐ Recalc AI_Lanes of speedfiles? It can be necessary if float or width values have changed.
☐ Connect close points to near points of neighbour blocks, if distance is less than :

OK Cancel

Track shape (XY)

If enabled, removes all turns from the track, shaping it into a large circle instead.

X/Y multipliers

Defines the scaling of the resulting circle, can be used by making ovals if the X value differs from the Y value.

Create additional curves

If enabled, defines how many additional curves will be made, along side their size.

Road elevations

If enabled, removes all slopes from the track, but only affects the mean elevation of each track node: all small bumps will be kept, and the surrounding scenery retains its relative elevation.

Set road height to

Sets road height to the value you provide.

Create additional curves

If enabled, defines how many additional hills and dips will be made, along side their size.

Delete

Scenery elevations

If enabled, flattens the whole track 3D structure (but not the objects), setting its Z coordinate to equal that of the corresponding track node: the slopes of the track are kept, but the scenery and small bumps are flattened. When combined with the previous option, this flattens everything.

Lanes & fences

If enabled, removes the miscellaneous track polygons corresponding to road lanes and fences.

Sound sources

If enabled, removes the sound sources of the track.

Polygon objects & light sources

If enabled, removes all polygon objects (displayed in blue), as well as the light sources.

Extra objects

If enabled, removes all extra objects (displayed in green).

Global animated objects

If enabled, removes all global animated objects from the track (not displayed by T3ED). If this option is enabled, then it will not be possible to undo the changes; making a backup copy of the track is therefore recommended.

Clear All Shading Values (No Shadows)

Deletes all shading values of track polygons.

Speedfiles

Forward speed file/backward speedfile

If enabled, it will reset the data in the for the forward and/or backward speedfiles based on the settings below:

- **Start/End:** beginning and end to apply the command in track slices
 - Track slices are 1/8 of a block.
 - In order to get a track slice number, multiply the track block number by 8 and add up to 8 in order to get the desired slice.
 - Some track blocks (especially the last ones) have less than 8 slices.
- The options to reset the speedfile entry data are the following:
 - Set float values to zero
 - Recalculate float values (straight)
 - Recalculate float values (curve)

Misc

Road textures

If enabled, sets the textures of all passable road polygons to a given value (modifiable using the corresponding input box and Browse button). This also affects low and medium-resolution polygons, although the heuristics for guessing their passable or non-passable state sometimes fail.

Scenery textures

If enabled, sets the textures of all non-passable road polygons to a given value (modifiable using the corresponding input box and Browse button). This also affects low and medium-resolution polygons, although the heuristics for guessing their passable or non-passable state sometimes fail. The objects are not affected.

Cut FRD-file to blocks

If enabled, shortens the track by cutting down the amount of track blocks to the specified value.

Set default block neighbours

If enabled, sets block neighbour data automatically.

Recalculate PolyVRoad

If enabled, recalculates polygon Virtual Road flags with the specified settings from the drop down:

- Old mode (per polygon)
- One block vector, based on VRoad points for all polygons
- New mode. Own vectors for each polygon (straight)

Recal AI_Lanes of speedfiles?

If enabled, will recalculate AI_Lanes values of speedfiles, use this if float or width values have changed.

Connect close points to near points of neighbour blocks, if distance is less than:

If enabled, it will connect points closest to the ones of neighbour blocks if the distance is less than the specified one.

Replay Camera Editor (Whole Track menu)

For NFSHS tracks only. Opens up the Replay Camera Editor, which allows for editing replay camera angles of the track.

The screenshot shows the 'T3ED : Replay Camera Properties' dialog box. At the top, there's a title bar with a close button. Below it, a 'Selected Camera:' dropdown menu is set to 'Camera 1, Type 3', with 'Copy' and 'Delete' buttons to its right. The 'Basic Camera Properties' section contains a 'Camera Type:' dropdown set to '3: Static position, tracking car, with zoom'. Below this are 'Position:' fields for X (664.702), Y (-434.039), and Z (34.9442), along with a '< - vroad pos' button. To the right are 'Slice Numbers (0-958):' fields for Start (115), Render (144), and Stop (144). A 'Zoom Factor:' field is set to 0.987796. The 'Unknown Rotate Values' section displays a 3x3 grid of numerical values: [0.852896, -5.36442e-005, 0.522036, -0.0907557, 0.984599, 0.148509, -0.514086, -0.174068, 0.839733]. An 'Edit Unknown RotateValues' button is at the bottom of this section. At the very bottom of the dialog are 'OK' and 'Cancel' buttons.

Unknown Rotate Values:		
0.852896	-5.36442e-005	0.522036
-0.0907557	0.984599	0.148509
-0.514086	-0.174068	0.839733

Selected Camera

Selects the replay camera you want to edit

Copy

Copies the currently selected camera into a new one

Delete

Deletes the selected camera

Cameras can also be [duplicated](#) and [deleted](#) as any track object in the main T3ed window.

Basic Camera Properties

Camera Type

Selects the type of the selected camera. The following types are available:

0. Static position, fixed heading
1. Static position, tracking car
2. Static position, fixed heading, with zoom
3. Static position, tracking car, with zoom
4. Moving and tracking car
5. Bumper
6. Revolving around and facing car
7. Attached to the passenger side wing of car, facing forward
8. Angled view of car from the driver's side, facing forward
9. Angled view of car from the driver's side, facing backward

Position X/Y/Z

Position coordinates of the replay camera; only works for camera types 0 to 3. It's advised to not put the camera too close to track geometry in order to avoid clipping, especially with high Zoom Factor values.

- Camera position can also be changed by moving the replay camera objects with [Move X/Y](#) or [Move Z](#)

<- vroad pos

Click on this button to set the replay camera position at the Virtual Road point of the slice corresponding to its Render value under Slice Numbers.

Slice Numbers (0-<last track slice>)

Sets the start, render and end slices for the replay camera activation.

- Track slices are 1/8 of a block.
- In order to get a track slice number, multiply the track block number by 8 and add up to 8 in order to get the desired slice.
- Some track blocks (especially the last ones) have less than 8 slices.
- **Start:** slice where camera is activated.
- **Render:** slice where the camera belongs; it's usually where the camera is placed or in between the Start and Stop slices.
- **Stop:** slice where camera is deactivated.
- Slice numbers for the selected camera also can be adjusted in the main T3ed window. They will be shown as a thick cyan line on track.
 - To adjust Start/Render/Stop slices at once while moving the camera with Move X/Y or Move Z, hold the Alt key while moving the camera.
 - To adjust Start/Stop points, select the camera and press the following:

- PgUp/PgDown to adjust the Stop slice.
- Alt + PgUp/Alt + PgDown to adjust the Start slice.

Zoom factor

Specifies the zoom and viewing angle of the replay camera. Usual values are:

- 0.5: Very wide
- 1.0: Regular
- 2.0: Narrow
- 10.0: Strong zoom

It's advised to not use very high values since they can cause clipping and z-fighting.

Unknown Rotate Values

Only for type 0 and 3 cameras. They determine the angle of a fixed headed camera, which is expressed as a rotation matrix.

You can use this 3D Rotation Calculator (<https://www.andre-gaschler.com/rotationconverter/>) to convert XYZ axis with magnitude rotation angles to a rotation matrix. Just make sure both input and output units are set to degrees. Then you can copy the matrix values of the calculator to the editor's.

XYZ rotation axis work as the following:

- X will rotate the camera up and down
- Y will rotate the camera left and right
- Z will tilt the camera left and right

Shortcuts

Keys
(with a selected camera object) Alt + Enter

Shadow Ray Tracer (Whole Track menu)

Opens up the Shadow Ray Tracer dialog, a tool to create track shading.

- [Sun](#)
- [Lights](#)
- [Start/end blocks](#)
- [Create new shadows for](#)
- [Object types that can cast shadows](#)

The screenshot shows the 'T3ED : Shadow Ray Tracer' dialog box. It has a title bar with a close button. The dialog is organized into several sections: 'Sun', 'Lights', 'Start at block', 'End block', 'Create new shadows for', and 'Object types that can cast shadows'. The 'Sun' section includes a checked 'Raytrace sun and shadows' checkbox, 'Sun Theta' and 'Sun Rho' input fields, and 'Sun color' and 'Shadow color' color pickers. The 'Lights' section has a checked 'Ray Trace Lights' checkbox and a 'Light distance multiplier' input field. The 'Start at block' and 'End block' fields are set to 0 and 119 respectively. There are three checked checkboxes: 'Ignore transparent vertices (alpha below 255)', 'Check middle between points (Black & blue polygons)', and 'Copy shadow to very close points'. The 'Create new shadows for' section contains two columns of checkboxes for 'Track polygons', 'Track lanes', 'Extra Objects', 'Track fences', 'Objects', and 'Global objects'. Below this is a sub-section 'Extra & global object types' with checkboxes for '1 Simple (global only)', '2 Simple (HS only)', '3 Animated objects', '4 Collision', '5', and '6 Complex (global only)'. The 'Object types that can cast shadows' section at the bottom has a similar set of checkboxes. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

T3ED : Shadow Ray Tracer

Sun

☒ Raytrace sun and shadows

Sun Theta (try sun values from tr.ini) : 0 0.56 Sun Rho (try values close to 1) : 0.85 0.870000

Sun color R: 255 G: 255 B: 255

Shadow color R: 102 G: 102 B: 102

Lights

☒ Ray Trace Lights Light distance multiplier: 0.1

Start at block: 0 End block (119 = whole track) : 119

☒ Ignore transparent vertices (alpha below 255)

☒ Check middle between points (Black & blue polygons)

☒ Copy shadow to very close points

Create new shadows for

☒ Track polygons ☒ Track fences

☒ Track lanes ☒ Objects

☒ Extra Objects ☒ Global objects

Extra & global object types

☒ 1 Simple (global only) ☒ 2 Simple (HS only)

☐ 3 Animated objects ☒ 4 Collision

☐ 5 ☐ 6 Complex (global only)

Object types that can cast shadows

☒ 1 Simple (global only) ☒ 2 Simple (HS only)

☐ 3 Animated objects ☒ 4 Collision

☐ 5 ☐ 6 Complex (global only)

OK Cancel

Sun

Raytrace sun and shadows

If enabled, will generate sun (light) and shadow values for the specified slices using the values below.

Sun Theta

Horizontal angle of the sun, measured in revolutions (1 revolution is equivalent to 360 degrees). Usually should match the sun's angleTheta value of the Tr(N)(W).ini.

Assuming your track points north from its start/finish line, you can use this cheat sheet to guide yourself:

- 0.00: East
- 0.25: North
- 0.50: West
- 0.75: South

Sun Rho

Vertical angle of the sun, measured in revolutions (1 revolution is equivalent to 360 degrees). Usually should match the sun's angleRho value of the Tr(N)(W).ini.

A value of 1.0 would put the sun is at it's highest point in the sky, while a value of 0.5 would put it right onto the horizon line.

Sun/shadow colors

The colors that will be used for lighting (sun) and shadows, as RGB255 units. Keep on mind they will be mixed with the AmbientRed/Green/Blue values of theTr(N)(W).ini.

Lights

Ray Trace Lights

If enabled, it will perform light raytracing on the adjacent track geometry using the light sources and their [glow settings](#). Light radius can be adjusted with the Light distance multiplier field.

Start/end blocks

Specify the range of blocks in which the ray tracer will be applied.

Ignore transparent vertices (alpha below 255)

If enabled, it will ignore points with alpha value lower than 255.

Check middle between points (black & blue polygons)

If enabled, it will check for shared points between polygon objects and track polygons to apply the ray tracer to.

Copy shadow to very close points

If enabled, it apply the same kind of lighting/shadow to points that are very close to each other.

Create new shadows for

Toggle on/off shadowing for the following elements:

- Track polygons
- Track lanes
- Track fences
- Objects
- Extra objects
- Global Objects

Extra & global object types

Toggle on/off shadowing for the following extra/global object types:

1. Simple (global only)
2. Simple (HS only)
3. Animated objects
4. Collision
5. Unknown
6. Complex (global only)

Object types that can cast shadows

Toggle on/off shadow casting from the following extra/global object types:

1. Simple (global only)
2. Simple (HS only)
3. Animated objects
4. Collision
5. Unknown
6. Complex (global only)

Help menu commands

The Help menu offers the following commands:

Help Topics	Offers you an index to topics on which you can get help (non-functional from Windows 10 onwards)
About	Displays the version number of this application.

Help Topics command (Help menu)

Use this command to display a list of available help topics.

About T3ED command (Help menu)




















Use this command to display the copyright notice and version number of your copy of T3ED.




Toolbar



The toolbar is displayed across the top of the application window, below the menu bar. The toolbar provides quick mouse access to the most frequently used commands in T3ED.

To hide or display the Toolbar, choose Toolbar from the View menu (ALT, V, T).

Button	Action
	Open an existing track.
	Save the active track with its current name.
	Zoom in.
	Zoom out.
	Select translation mode , where the scrollbars scroll the track view.
	Select rotation mode , where the scrollbars control the observer's view point.
	Toggles Track View
	Select block mode .
	Select point mode .
	Select extended point mode (single point with automatic smoothing).
	Select object mode .
	Select polygon mode .
	Select Virtual Road Points mode
	Undoes the last editing action
	Select the move XY tool .
	Select the move Z tool .
	View and change texture of an element.
	Delete an element
	Create or duplicate an element.

	Select the merge/split tool.
	Display the About dialog box.
	Contextual help (non-functional from Windows 10 onwards)

Scroll bars

Displayed at the right and bottom edges of the document window. The scroll boxes inside the scroll bars indicate your vertical and horizontal location in the document. You can use the mouse to scroll to other parts of the document. The arrow keys can also be used for the same purpose.

In [translation mode](#), the scroll bars act in the usual way, by simply scrolling the track view.

In [rotation mode](#), the scroll bars are used to rotate the observer's view point.

Status Bar

The status bar is displayed at the bottom of the T3ED window. To display or hide the status bar, use the Status Bar command in the View menu.

The left area of the status bar describes actions of menu items as you use the arrow keys to navigate through menus. This area similarly shows messages that describe the actions of toolbar buttons as you depress them, before releasing them. If after viewing the description of the toolbar button command you wish not to execute the command, then release the mouse button while the pointer is off the toolbar button.

The right areas of the status bar provide basic information about the current mode:

Indicator	Description
Block ###	Indicates the track block to which the current selection belongs.
TRANSL/ROTATE	Indicates whether the scrollbars are configured in Translation mode or in Rotation mode .
X/Y/Z/InMemZ	Displays coordinates of the selected element.
Normal (z)/Shader (z)/Average (z)/Line (x,y)/Line (x,y,z)	Displays current Paste Mode.