

TLEdit—NXSYS Track Layout Editor

©Bernard S. Greenberg 1997–2022

Updated 28 June 2022 (NXSYS version 4.7)

TLEdit is the NXSYS Track Layout **E**ditor. It is used to create and edit layouts as seen on interlocking panels.

TLEdit does not create or edit logic in any way. You have to make the interlocking work by designing the relay logic, a complex task whose facilitation is the entire reason for the existence of NXSYS. See the accompanying document on interlocking logic design for an introduction to that not at all simple subject.

Contents

[Basics](#)

[Rodentation summary](#)

[Keystroke repertoire](#)

[Toolbar](#)

[Creating and editing track](#)

[Creating and editing other objects](#)

[Topological joint deletion prohibitions](#)

[Switch orientations](#)

[Switch consistency](#)

[Conditions on joint merging](#)

[Undo/redo policy and semantics](#)

[Setting the initial layout size \(Mac\)](#)

[North of West 4th St \(IJ ID Limit bug\)](#)

[Assigning track-circuit identifiers](#)

[Creating and insulating crossovers](#)

[Number \(lever, track circuit\) management](#)

[Mac-specific issues](#)

Basics

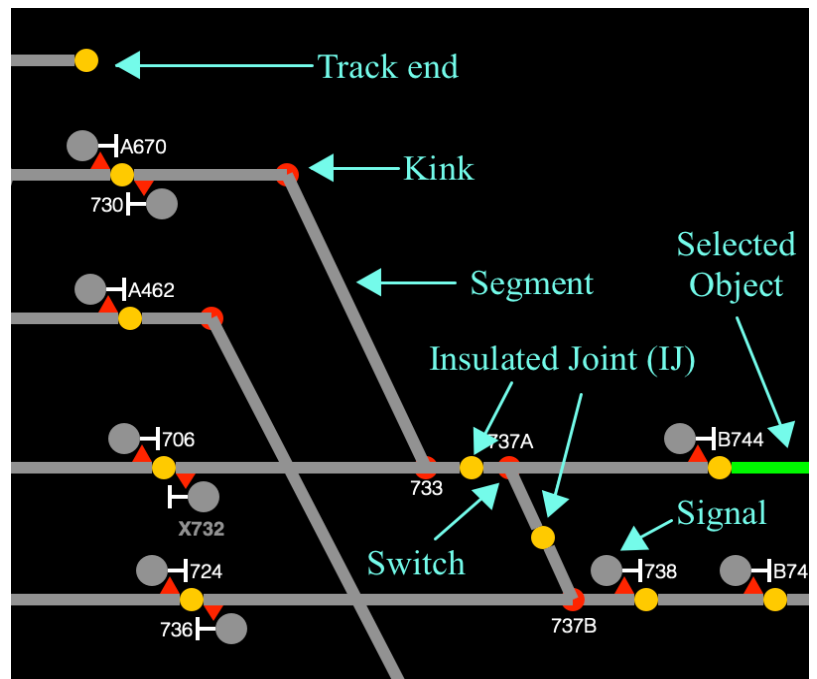
Layouts consist of straight line **track segments** which meet at **joints**, with other objects to be discussed. Tracks can only bend or end at joints. A joint can sport one, two, or three segments (“branches”) emanating from it — a joint with

- **One** branch is the end of a track, a **track end**. Track ends can (and should) be insulated. Note that a track end is not a bumper, but merely the limits of the route managed by the panel.
- **Two** branches is either an **insulated joint (IJ)**, or a (non-insulated) “**kink**” in a track. Being insulated is a settable, reversible state.
- **Three** branches is a **switch** (“turnout”).

In real life, insulated joints and switches *have nothing to do with each other*, but in TLEdit (but not NXSYS proper), they are two subspecies of the same thing, i.e., **joints**.

IJs appear as yellow dots; kinks appear as red dots where two segments join, either at an angle or not. The dots do not appear in NXSYS, where IJ’s appear as little gaps, and kinks do not appear at all (the track appears continuous, and simply diverges at an angle). Use the **Show joints** tool (an on/off button) to see what the panel will look like in NXSYS (i.e., no red or yellow dots).

“Kinks” do not correspond to any feature of real trackage, but they are “standard style” on interlocking panels and other schematic track representations all over the world. A “straight kink”, i.e., one at no angle, (two track segments meeting directly facing at each other) has no reason to exist (as it is, definitionally, not insulated). By (left) click-and-drag, you can “draw” (create) segments between any two points or extant joints (making them into switches, or extending a track end), or to or from arbitrary points on extant segments, turning that point into a switch. You cannot draw a



Basic TLEdit objects

segment to or from a switch (which is a joint already having three branches) or an insulated joint which is not a track end. Clicking and releasing (without moving) on empty space will do nothing. Clicking and dragging from empty space will create a new segment with one (or perhaps two) new joints at its end(s) (depending upon where you release it). When segments are created, they have no track circuit assigned (i.e., the segment is “unassigned”). See the [section below on track circuit assignment](#). Joints of one or two branches can be insulated or de-insulated from their **Properties** dialog, the **Insulate** tool of the toolbar, or the keystroke **Ctrl-I** (Mac **⌘I**). You cannot insulate a switch. You cannot de-insulate a joint which hosts a signal or an exit light, or separates differing, assigned track circuits. Saving the layout insulates all track-ends (this may change).

Three-line rodentation summary

- **MOUSE LEFT** (click and drag) **creates new trackage**, including switches. Clicked, it selects extant objects.
- **MOUSE RIGHT** (click and drag) **moves joints**, and adds *new* joints mid-segment
 - **but CONTROL-MOUSE** instead of **mouse right** on the Mac.

Keystroke command (shortcut, accelerator) repertoire

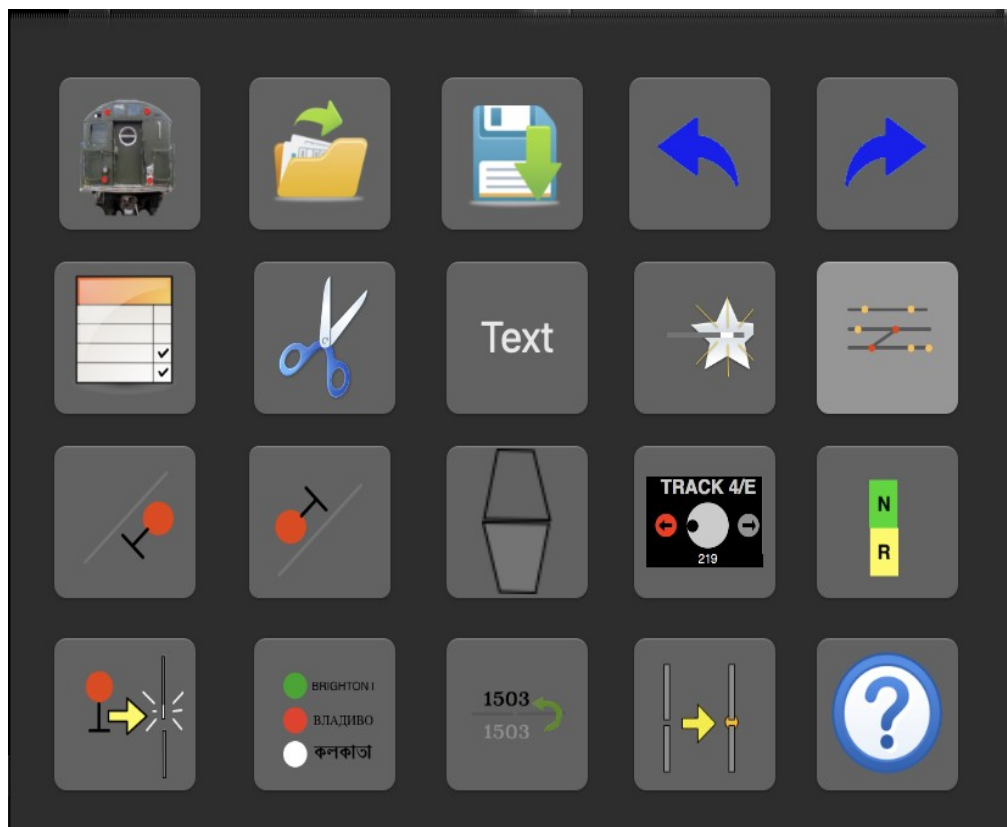
Note that files may also be supplied by pathname to the command executable (easy on Windows, but on the Mac, “not so much”). **Do not attempt to hand-edit** files produced by TLEdit; the results may crash TLEdit and/or NXSYS and be uneditable. Files created by TLEdit should be included in the interlocking definition by use of the **INCLUDE** special form; see the relay language document for more detail.

Windows	Mac	Meaning
F4	⌘Q	Exit application (after checking for unsaved modifications).
F1		Help (show this document).
^O	⌘O	Open layout (.trk) file.
^S	⌘S	Save to file; discard all redoable undone changes and the undo history.
^Z	⌘Z	Undo last forward or redone action.
^Y	⇧⌘Z	Redo last undone action.
^W	⌘N	Wipe (discard) current layout. Create a new empty layout. Not undoable.

^X or Backspace	⌘X or delete	Cut (delete) selected object, undoably.
^P	⌘P	Edit properties of selected object, undoably.
^I	⌘I	Insulate or remove insulation from selected joint (if allowed), undoably.
Alt+S		Shift all objects in layout spatially.

Toolbar

The TLEdit toolbar is central to all operations other than creating track. The function is identical on Mac and Windows. The tools have tooltips on both systems. Here is the Mac toolbar (Windows follows):



First row: **Quit, Open file, Save file, Undo, Redo.**

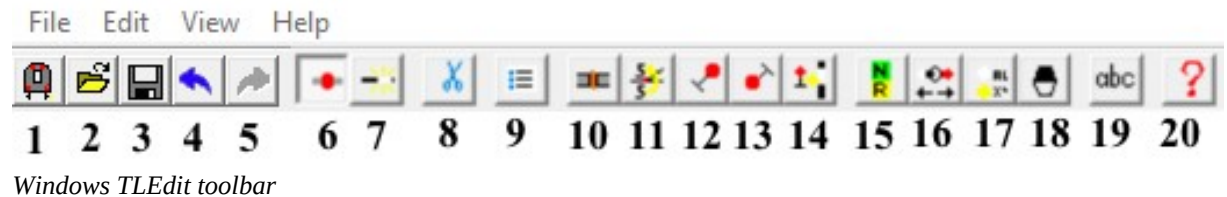
Second Row: **Properties Dialog, Cut, Create text string, Show exit lights, Show joints.**

Third Row: **Add/Select North/East pointing signal, Add/Select South/West pointing signal, Create panel switch, Create traffic lever, Create switch auxiliary**

key.

Fourth Row: **Create/Select exit light, Create generic panel indicator, Flip switch or IJ number position, Insulate or de-Insulate joint, Help.**

Here is the Windows toolbar:



- | | |
|--|--|
| 1. Exit (back of a train) | 11. Flip switch or IJ number side |
| 2. Open (an extant layout file) | 12. Place Signal, northward or eastward |
| 3. Save | 13. Place Signal, southward or westward |
| 4. Undo | 14. Create or select Exit Light at signal |
| 5. Redo | 15. Create Auxiliary Switch key |
| 6. Show Joints (non-selected ones, that is) | 16. Create Traffic Lever knob |
| 7 Show Exit Lights | 17. Create General Panel Light |
| 8. Cut | 18. Create General Panel Switch |
| 9. Properties dialog | 19. Create text string |
| 10. Insulate or de-insulate joint (toggle) | 20. Help |

Creating and editing track

Click and drag mouse left (Windows, or Mac trackpad, single finger) to create a new track segment. Click left on either an extant joint which is not already “full” (i.e., has fewer than 3 branches), some other point on an extant segment, or elsewhere (i.e., “empty space”), and drag the mouse, left button held. You will be dragging the end of a “rubber band” from where you started. Drop the mouse on either an extant joint which is **not full**, some other point on an extant segment, or in empty space—a new segment will be created in any case. An insulated joint with two branches (as normal) is considered “full”. To cancel, i.e., change your mind and create no new segment, drop the mouse end of the rubber band back on its stationary end. Slopes of less than 4% from the vertical or horizontal are assumed to be vertical or horizontal, and “snapped” into place.

To **move an extant joint**, click and drag it with mouse right (on Windows; control-drag on the Mac). Clicking and dragging mouse right in the middle of an extant segment creates a joint and moves it — you can insert a “kink” this way. If you drag a joint to another joint, and

there are no more than three branches among the two of them, the joints will merge into one. The rubber band lines will “click” onto a detente (and show as red) if TLEdit thinks it knows what you are trying to do. You cannot merge insulated joints. See the section on [Joint merging conditions](#) below.

Placing a new joint in the middle of an extant segment via either tool splits the segment in two at that point. For mouse left, the new joint will not come into existence until you actually drag the mouse. For mouse right, a new joint will come into existence immediately. If a segment with a track circuit is split, the new segments share that track circuit. Change one or the other or both only after insulating that, or some more appropriate joint (or TLEdit will refuse changing it).

Any object (joint, segment, signal, exit light, switch key, etc.) may be selected, which turns it green, by clicking left on it. A newly-created object, or a joint just moved, is automatically selected. Only one object may be selected at a time; there is no concept (presently) of a multiple selection. Pressing the **Edit Properties** tool (or typing **Ctrl+P**, both Windows and Mac) when an object is selected brings up its **Properties dialog**, which is different for each type of object, allowing detailed control of its properties. On the Mac, “secondary gesture” (right-click replacement) on any object calls up its **Properties Dialog**, too.

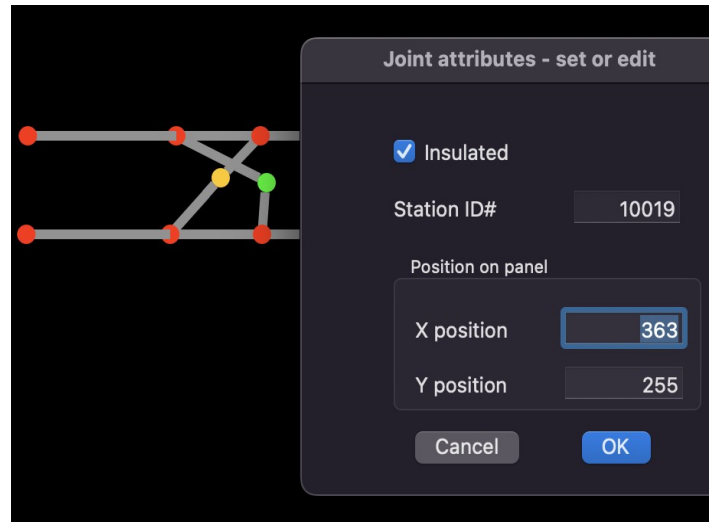
Use **Cut** (via the **Cut** tool or **Ctrl-X** or **Backspace**, **⌘X** or **delete** on the Mac) to delete a selected object. Loose ends of a deleted segment will vanish with it. **Cut** of a selected joint of two branches will remove the joint and collapse the two segments into one, joining the endpoints of the previous two. **You must delete all signals and exit lights at an insulated joint before the joint can be deleted.** See the section on [Joint Deletions](#) for more detailed conditions.

A joint of two branches can be made insulated by selecting it and clicking the **Insulate** tool (insulated joints usually appear as yellow dots, but you won’t see it as long as the joint is selected, and showing green) Creating a signal at an uninsulated joint of two branches it insulates the joint automatically. Insulation can be removed with the joint’s **Properties** dialog, the insulation tool, or **Ctrl-I** (**⌘I** on the Mac).

As TLEdit now makes it virtually impossible to overlay joints on top of each other (when they cannot be merged), the one case where this overlay is necessary, to wit, the center of a double-crossover, where the two insulated joints of both crossovers coincide, requires unusual means. Adjust the exact location of both joints with their **Properties Dialogs**. It will likely be difficult or impossible to click on the “lower” one—move the “upper” one reasonably far away (with either the mouse or the dialog), set the location of the lower one, and then the upper one (both with the **Properties Dialog**) identically. See the screenshot at right.

Scroll down or to the right to extend layout those ways. To extend up or to the left, use **View|Shift Layout (Alt+S** on Windows, no accelerator on Mac). This takes much time to run. To set the default view origin that comes up in NXSYS, scroll until you are looking at it, and use **View|Set Viewport here**.

The problem of how to establish an empty space of known dimensions, or “large enough” on the Mac is non-obvious, and discussed [below](#).



Aligning double-crossover IJs

Creating and Editing Other Objects

All objects other than track and its joints are created via the toolbar. For signals and exit lights, select the associated object (IJ or signal, respectively), and click the tool. For other objects (this is slightly non-standard), click the tool *first*, **release it**, and move the mouse to, and click the place where you want it to go. If you change your mind, press **ESC** before dropping it, or drop it anywhere, select it, and **Cut** it. **Do not attempt to drag from the toolbar**.

To create a signal: Select any joint of 1 or 2 branches (i.e., not a switch), or make one with mouse right. Choose the one of the two **Create signal** with the closer of the two orientations (If you guess wrong, cut it and try again). Signals move around as you move the joint. Click right on a signal calls up its **Properties dialog**, which allows editing or assignment of lever numbers, lenses, absence of stop, etc. **Station Number** is the decimal number that will be used to identify relays associated with the signal (when automatic) and from which the signal ID plate will be computed. IRT and BMT/IND conventions for choosing this differ. Select and **Cut** to delete a signal.

To create an (entrance)/exit light: Exit lights can be created by selecting a signal and clicking the **Create or Select Exit Light** tool to create an exit light for exiting at that signal. Normally, exit lights, other than a selected one, including one just created, are hidden. To show all exit lights, toggle the **Show/Don't Show Exit Lights** tool; they can only be clicked for cutting or properties (lever number) editing when lit. To create a same-direction exit (when there is no signal in the entrance direction), create such a signal, create its exit light, and **Cut** the signal. Exit lights, when created, inherit the lever number, if any, of their signals. If there are two exit lights at a joint in different directions but only one signal, assign the unsignalled one a lever number with its **Properties Dialog**.

Insulated joint station numbers and **switch numbers** can be assigned via the joint's **Properties dialog**. All insulated joints and switch joints are automatically assigned unique numbers (to facilitate reload) over 10000, which do not display on the panel. Set them as you want via the dialog. You can “flip” IJ station numbers and switch numbers between sides of the track (to address visual collisions) with the **Flip Numbers** tool.

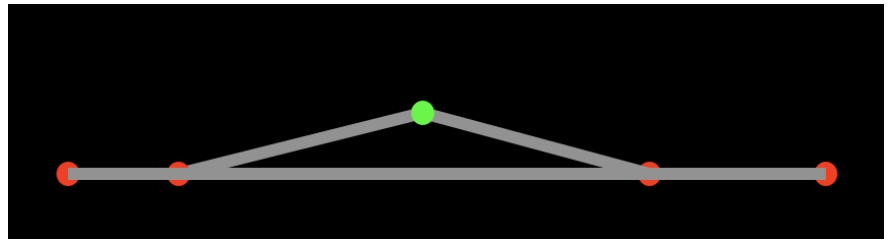
Auxiliary switch keys, generic panel switches, and lamps are created by pressing the tools that resemble them: move the mouse to the final position, and click left to drop it there. Type any character or click right to abort. To edit a key's switch number or board position, click right on it or use the **Properties dialog** tool. To move one, select it first, release the mouse button, and then click and drag left to a new position (the need to click twice may change). Select it and **Cut** to destroy it later.

Traffic levers and **generic panel switches** and **lamps** are created with the three tools that resemble them. Creation and property editing is identical to that for auxiliary switch keys (above). Traffic lever graphics may overlay tracks; although they may (currently) appear “behind” trackage in TLEdit, they will show in front in NXSYS. Use an **arbitrary text string** to label one to taste beyond the lever number. “Normal” direction (left or right — affects relay nomenclature and initial state) is set from the traffic lever's **Properties Dialog**.

Arbitrary text strings of any size or font are created by pressing the tool labelled “abc”: move the mouse to the final position, click left to drop. Dragging/moving, selection, destruction, and editing is the same as for auxiliary switch keys. Click right or use the properties tool to set text content, font, size, etc. Avoid private fonts not likely to be on someone else's machine.

Topological joint deletion prohibitions

There are some joint creation and deletion scenarios which, if not caught by TLEdit, would produce invalid layouts that could not be saved, or worse, loaded, edited/fixed, or programmed. Typical of these is



The selected (green) joint may not be deleted.

attempting to create a segment between a joint and a point on a (or the) track segment already emanating from it, including “connecting” two joints already connected. Another is attempting to delete one of the joints of a three-joint “segment triangle” (see image at right). If you are trying to obliterate such a triangle, delete the segments which are its unwanted sides. Note that this does not rule out “sidings”, even though expressing one in this way is a

poor idea (i.e., one track circuit with no parallel extent!); this restriction is about the proper procedure for *deleting it*.

Of course, real-life tracks do not bend at angles (although converging routes at a switch do), but trackage on panels does. However, it is (or, rather, was) all too easy to create corrupt layouts by accident when editing trackage in a small space, and nearly impossible to diagnose or remedy the resultant chaos. TLEdit will now diagnose and disallow dropping nodes being created or dragged in known cases of such conditions. If you can create unloadable or uneditable trackage, please let me know (via GitHub mail).

Switch Orientation (which route is “normal”?)

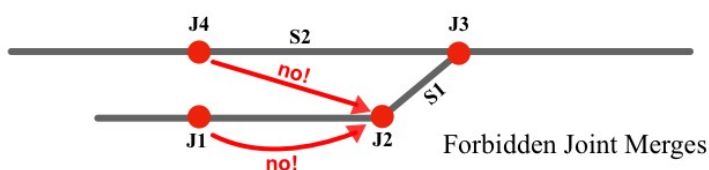
TLEdit by default assigns the three branches emanating from a switch joint the designations **stem**, **normal**, and **reverse**, based upon the angles between them. In some cases, notably for slip switches, this default is not correct. The two buttons **highlight normal** and **swap normal** on the switch’s **Properties dialog** can be used to see which branch is “normal” (selecting it) and swap the definitions of normal and reverse (highlighting the new **normal**, as it were) to facilitate manual adjustment.

Switch Consistency (A and B switches vs. singletons)

Any switch is either a “singleton” or half of an A-B pair, i.e., a crossover. You choose which from the switch’s **Properties** dialog, either by the “radio buttons” or typing (for example) 15A or 15B (or 15a etc.) as the lever number. 15A and 15B are different switches but operated and interlocked by the same logic. Normally, A and B are two ends of a crossover; NYCT practice at double-crossovers is to group the As and Bs together on one track, although NXSYS does not require or force this. TLEdit will not let you save, and NXSYS will refuse to load, a layout in which an A or B switch exists without its other half.

Conditions necessary for joint merge

In order for **Undo** to work reliably, some uncommon operations have new restrictions, which are all reasonable. Chief among these is joint merging. The joint being moved will be called the *movend*, and the joint onto which you are trying to drop it will be called the *receiver*. These are the current restrictions on joint merging. All these “bad moves” are diagnosed. Please refer to this diagram:



- You may not drop a joint on the other end of either segment emanating from it, i.e., try to reduce a segment to zero length

that way. Cut the segment or the joint as appropriate instead, if that is what you are trying to do. That is, in the diagram, **J1** may not be dragged onto **J2**.

- You may not drop the movend on a joint at the end of a colocated segment, that is, a segment **S1** one of whose ends is the receiver, and whose other end is a third joint, **J**, which latter is connected by another segment, **S2**, to the movend itself.

That is, if there is a path from the movend to the receiver through exactly two existing segments, the merge is disallowed. Were it to be permitted, **S1** and **S2** would coincide, connecting **J** to the merged receiver, which is impermissible. That is, in the above, **J4** may not be dragged onto **J2**, as segments **S1** and **S2** both “go to” joint **J3**.

- You may not drop a switch on any other joint, nor drop any joint on a switch.
- You may not drop a joint on an insulated joint. If you want to make the receiver be a switch, remove the receiver's insulation first.
- You may not drop an insulated joint on any other joint. If you want to merge them, remove the insulation from the movend first.
- You may not drop a joint not a track-end on any other joint not a track-end; that would cause 4 segments to emanate from the joint (slip switches and grade-crossings are not done that way).
- You may not merge joints unless all of the adjoining segments of the movend and the receiver have the same track circuit status, that is, either all unassigned or all assigned to the same circuit.

Undo/Redo policy and semantics

The TLEdit undo system (new with version 2.7) adds the long-needed ability to undo erroneous or unintended keystrokes and other “bad moves”, as in other mature interactive applications. It is supposed to work similarly in the expectable way; nevertheless, here is a more detailed discussion.

Note that unlike some other applications (on the Mac in particular), **TLEdit does not (currently) auto-save. You must/should still save your work** regularly. When you do save, you may no longer undo any changes made previously. This latter behavior, too, may differ from that of some other applications, and may change in the future.

For purposes of this discussion, we will divide all TLEdit user actions into **forward actions** and **presentation actions**.

Forward actions are those commands and mouse gestures that affect the content (i.e., the

description) of the interlocking, and will be saved when it is written out, and will be visible when it is loaded either in NXSYS or a subsequent invocation of TLEdit. Examples:

- **Creating and deleting** trackage, joints, and signals.
- **Creating and deleting** auxiliary objects (exit lights, traffic levers, auxiliary keys, text strings, etc.).
- **Changing properties** of trackage, joints, or auxiliary objects, including label position, insulation, and nomenclature, whether with the properties dialogs or toolbar commands.
- **Assigning** track circuits, including “wildfire” propagation.
- **Moving** joints or auxiliary objects, either with the mouse or by editing coordinates.
- **Redefining** the viewport origin, and globally shifting all object positions.

Presentation actions are all the rest, i.e., those commands and mouse gestures that affect what you see, but ***do not*** affect the content of the interlocking, and are not reflected in NXSYS. Examples:

- Scrolling, zooming, or reshaping the display window
- Selecting objects
- Setting the ostensible geometrical size of the layout
- Making exit lights visible or invisible
- Showing track circuits

The “undo system” records all forward actions in a *Stack*, that is, a list kept like a stack of papers, such that only the last one placed on it (“pushed onto it”) is accessible, i.e., *LIFO*, “last in, first out”. When the last one pushed is removed, the previous one is exposed, and so on. This stack is called the *Undo Stack*. When the interlocking is successfully written out to a file, the undo stack is cleared, i.e., all its records discarded.

If the undo stack has records in it, the file needs to be written out. If it has none, the file which would be written out would be identical to the last file read in, or written out, i.e., the “document” has effectively “not been modified”.

The **last forward action** (i.e., the top of the undo stack) can be “undone” by typing **Ctrl-Z** (Mac ⌘Z), the **Undo keystroke**. That means that if the last forward action was a “move”, the moved object will be returned to its previous position. If the last forward action was the creation of a signal or auxiliary object such as a text string, the object will be deleted. If the last forward action was the deletion of a signal or other object, the object will be restored with all its attributes. And so on. The concept should be familiar.

When a forward action is undone via the **Undo** keystroke, the record of the action is removed from the **Undo** stack, and pushed onto a second stack, the **Redo stack**, which facilitates the undoing of undos (as is common). After you've done that, you are in a state (the “**redoable**

state”) where three courses of action are possible:

- ◆ You can type the **Undo** keystroke again, and undo the next older forward action, moving it to the **Redo** stack. Doing this repeatedly, you undo entire substantive stretches of modification.
- ◆ You can type the **Redo** keystroke (**Ctrl+Y**, Mac ⌘Z), which causes the last action undone to be *redone*. It also removes it from the **Redo** stack, but also pushes it back onto the **Undo** stack (so you can undo it again, and perhaps repeat this loop many times). Or you can issue the **Redo** keystroke repeatedly, and so restore entire lengthy stretches of undone actions.
- ◆ You can perform some **new forward action**, which will, as all, be pushed on the **Undo** stack, **but the Redo stack will be cleared, i.e., the redoable state will be terminated**. You can still undo that action, and earlier not-undone actions, but you cannot redo undone (but not redone) actions recorded on the **Redo** stack before the new forward action; they will be gone. Every forward action (actually issued, not just restored via **Redo**) clears the **Redo** stack.

Presentation actions do not figure into this, and do not alter the status of either stack or the redoable state.

The **Edit** submenu of the application menu is of extreme interest here. When either the **Undo** stack or **Redo** stack has actions in it, the menu items **Undo** and **Redo** will be respectively enabled (and disabled if not). What is more, the menu items will appear with a brief description of the action at the top of the respective stack, e.g., **Undo create signal**, **Redo move joint**, etc. Thus, selecting the application's **Edit** menu allows you to see if there are undoable and/or redoable actions available, and, if so, what they are.

When the **Undo** and/or **Redo** menu items are enabled on the **Edit** menu, they can, of course, be used instead of the corresponding keystrokes (which will be displayed there).

Setting the initial layout size (Mac-specific)

There is a chicken-and-egg problem concerning the limits of the “canvas”, i.e., the area extent of the panel, viz., placing objects in empty panels created with the **New** command. The size of panels is determined when loaded by NXSYS (or TLEdit) by the objects they contain: as a new panel starts off empty and tiny, as it contains no objects, you can't place any. To establish a large empty space capable of receiving new objects, use the “Set Canvas Extent” command in the **View** menu, and immediately place objects as far to the right and bottom as you wish the (initial) extent to be. The **New** command will throw this dialog in your face. While this command may be used to expand a panel at any time during a first or subsequent editing session, its values are *not saved or restored* with the interlocking – the size of the panel is determined solely by the dispersion of the objects it comprises. The initial suggestion

will be that of the window size, but that does not absolve you of the need to establish your intent.

North of W. 4 St. on the IND (current IJ ID limitation bug)

When NXSYS was born in the early 1990's, I did not imagine there was a need for Insulated Joint/track circuit identifiers of 5 digits or greater. That means either by the BMT/IND convention, where 4762 means "Track 4, station 762+00" or the IRT convention, where it means "Track 2, station 476+00". In fact, no BMT or IRT stationing numbers (hundreds of feet from "origin") reach 1000 (i.e., 100,000 feet = 19 miles). But, in fact, the entire IND is stationed from a point South of Staten Island... about 18 miles along the center-line of West 4th St station (where it is defined), and *all* IND stationing north of 14th St (6th or 8th Avenue) shows numbers exceeding 1000.

NXSYS currently (2.7 and all earlier) appropriates IJ (or other objects) numbers of 10000 or greater as temporary (system-generated) ID's, which need need not be shown. It is thus currently impossible to correctly express IND interlockings north of West 4th St, including the entire Bronx, Queens, and Washington Heights IND rights-of-way, as well as massive interlockings such as 59th St and 34th St 6th Av. Neither I nor anyone else has tried to code one in the past 25 years!

Upcoming versions of NXSYS and TLEdit will have the ability to assign and use such numbers, and use a different convention for temporary IJ identifiers, which will, of necessity, require incompatible changes to the expression of track layouts, i.e., the creation (when elected) of layouts that cannot be loaded or edited by earlier versions of the application.

Assigning Track Circuit Identifiers

Track segments assemble into track **track circuits**; NXSYS generates and manages T relays and "line-o-lite"s on a per-circuit basis. Select and **Property Edit** a track segment to assign a numeric track circuit relay ID. All track segments reachable from the one being edited without passing an insulated joint will have their Circuit IDs changed as well. You can assign the default ID (0), which means "unassigned" ("not yet assigned", the initial state of all). Track circuits thus spread by "wildfire". If you make a mistake with this, many unintended segments may acquire the wrong circuit: simply type the **Undo** keystroke to revert to the previous state.

Track segments with circuits assigned light with white "line-o-lites" in TLEdit. The algorithm for managing the lighting of segments dependent on switch position, within one circuit, is a "complex heuristic" built into NXSYS (which occasionally fails). Individual segments are not (now) "programmable" as they are IRL (in real life, or *Interboro Rapid Life*, if you prefer).

NXSYS does not use the ambiguous term "track section", which has utility in real life where

circuits and track extents coincide.

Let us call a sets of contiguous track segments delimited from other sets by insulated joints (or end of track) *segment groups*. In NXSYS 2.7, you can no longer assign track circuits to individual segments. The **Properties Dialog** on a segment now brings up the **Track Circuits Dialog**, which allows you to assign a track circuit (by number) *to an entire segment group*. The dialog will tell you how many segments are in it before you do. All segments are created unassigned. By specifying track circuit 0, or a blank entry field, you can restore a whole segment group to the “unassigned” state. Segments with assigned track circuits appear white (as do “routed circuits” in the NXSYS application).

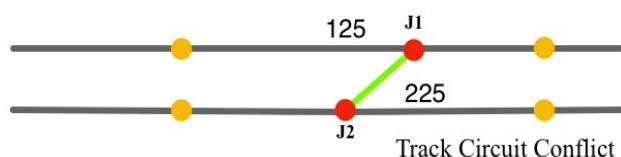
The new rules in version 2.7 are designed to prevent you from creating a situation where different track circuits (including “unassigned”) meet at an uninsulated joint. For instance, if two circuits meet at an insulated joint (the only valid state), *you may not delete (Cut) that joint*, because the track circuit of the resulting segment would be ill-defined; the application will prevent such an action. If you really want to eliminate a track circuit by eliminating a joint, you must decide in advance which circuit will become part of the other, and assign that circuit to the other's segment group, at which point you can delete the joint. The same applies to deinsulating a joint; the application will disallow this when the insulation separates two differing circuits. Separating circuits is the purpose of insulating joints!

Note that joints between segments of which neither has a track circuit currently assigned may be insulated and deinsulated freely. To break an established track circuit into two, create a joint where you want it (click right on the segment), insulate it, and reassign the segment group you wish to make into a new track circuit.

Creating and insulating crossovers

One can and does create crossovers with the greatest of ease by clicking (left) on one track and dragging to another. As IRL, crossovers have to have an insulated joint in their middle to enable trains to pass from the track circuit of the starting track to that of the destination track. In TLEdit, of course, you must create the crossover first, and only then insert, and then insulate, the joint.

It is best if you do this *before* you assign any relevant track circuits. If you do, everything will work optimally, and the crossover halves will each belong to the correct track circuit. But should you fail to create the insulated joint before assigning a track circuit, the following situation would arise:



Assume that the upper track, between the two yellow insulated joints, is track circuit 125, and the lower 225. When you create the crossover, which will be green as is always the case in TLEdit, it connects the

two track circuits; the situation depicted is inconsistent. Were you to attempt to change the single track circuit of the crossover, not only it, but *both circuits at its ends form a single segment group, and **all would be changed to the circuit identified***. TLEdit will catch this, and refuse, with the following message:

The group of segments reachable by non-insulated joints includes more than one assigned track circuit. Create and/or insulate some joints and try again.

Number management (Lever and IJ numbers)

Obviously, two (real or virtual) levers cannot have the same lever number. As NXSYS interlockings are, or can be, virtual “central towers” containing multiple interlockings, lever numbers must be unique across all of them, just as at real central towers. Track circuit numbers, which “track” IJ numbers, so to speak, need to be more complex (think of places where rights-of-way diverge, for one), but right now they are constrained to numbers. More structure and flexibility are planned.

Therefore, TLEdit will prevent you from assigning the same signal number or switch number to two or more objects, and will stop you if you try. This applies to traffic lever numbers as well, which are drawn from the same conceptual “pool” of odd integers as switches (the NYCT convention is to use numbers ending in 9 for them, but that is not enforced by NXSYS or TLEdit.).

This gets tricky when undo and redo are taken into account; you can “resurrect” a signal you have deleted via “undo”, restoring its assigned lever number to currency. But if, in the meantime, you have reused that number ... well, quite happily, you can’t possibly be able to resurrect the signal in question before you have undone the (of necessity) subsequent actions that reused its number: *you cannot undo an arbitrary action at an arbitrary time!* This is all supposed to work.

Switch numbers are also subject to the [Switch Consistency Constraints](#).

Objects are created with lever (or circuit) number 0, which means that a number has not (yet) been assigned. You assign numbers with the **Properties Dialogs**. There are times, such as when a layout is saved, that TLEdit assigns *temporary numbers* to objects that lack a number, to enable them to be read back in and identified, whether they correspond to real-life interlocking objects (e.g., signals, switches) or not (e.g., kinks). This is quite inconsistent, and may be improved in the future, but should not affect usage of either application. These temporary numbers are always 5 digits—no other numbers in a layout can be (please see [this section on IJ ID limits](#)—again, more expressive, unambiguous and flexible IJ identifiers are

planned). You will see them in **Properties Dialogs** for objects read in whose lever/circuit numbers you have not yet set. You should not take advantage of them (i.e., deliberately create objects, especially relays, with them).

Exit lights, generic lamps and switches, and auxiliary keys, right now can bear any “identifier” number at all (they are often associated with other objects, such as signals or switches), but the number *must be unique within their class*, i.e., there can be a signal 252 and an exit light 252, but there cannot be more than one of either (2.7 feature).

IJ numbers are necessary in NXSYS for signals to know their (default) plate numbers. Track circuits are assigned explicitly via the segment **Properties Dialog**; they are not inferred from IJ identifiers. Assigned IJ numbers *do* appear on the NXSYS panel display.

Track circuit identifiers present larger challenges, beyond their roots in IJ identifiers and their problems. Track circuits are aggregations of segments, which must to be contiguous, but might not be while you are in the process of making them so, e.g., eliding an IJ. What’s more, such multiple segments are *expected* to share the same track circuit, so that is a normal, not error, state of affairs. Thus, no checking is done for reuse of track circuit identifiers. It is also possible in small, toy interlockings, for small track circuit numbers and lever numbers to collide. You’re on your own there, until planned enhancements to IJ identifiers ameliorate disambiguation. But this may not even be a problem, as the only standard relay nomenclatures that will collide are **NS**, “north stick” (route locking) for track, with “normal south” (nonvital move selector) for a singleton switch (although NXSYS relay menus will suffer confusion).

Mac-specific issues

Rodentation differences

All the differences here devolve from the lack of a “drag right mouse button” on the Mac. Control-mouse-left (tapping or clicking the mouse or trackpad while holding the control key) has been substituted throughout.

- To drag an extant track joint somewhere else, hold the “control” key while clicking and dragging it.
- To drop a new joint into a piece of track, hold the “control” key and click where you wish to drop it.
- Because “mouse right” is now available, “click-right” (double-finger tap) calls up the appropriate **Properties Dialog** on any object (whereas on Windows it cannot do so for joints and track segments) (that is, except joints: use **^P**).

Toolbar differences

The Mac toolbar is large and floating: you can move it around by dragging any part of it not a button. The icons are much clearer than the 16 x 16 bit icons of the Windows program. All of the tool buttons have “tooltips”; if you hover the mouse pointer over them for a few seconds they will show their meaning.

The toolbar may be hidden and called up with the **Show Toolbar** command in the **View** menu (⌘T (option-command-T)).

Other minor differences

Set Viewpoint Here and **Shift Layout** are on the **View** menu.

The document-modified query, i.e., that asked when you attempt to quit the application or read in a new layout when you haven't saved changes you have made, differs slightly, including its “sense”, from its Windows forebear. While the latter poses “You've made changes, do you want to write them out?”, its Mac avatar asks “Do you really want to quit/new file and lose your changes?”, that is, you are not offered writing them out, just continuing editing. Hence, if you really want to discard your work, where you would click “no” (to “save it”) to Windows, you now must click “yes” (to discard your edits) to the Mac. The query message explains this clearly in any case.

The relationship of the native Macintosh Font Dialog to its calling applications differs markedly from that on Windows, and some changes to the flow of the **Text String Properties** dialog devolve from that. The Macintosh font and color dialogs are not modal; when exposed, you can interact with them or with the **Text String Properties** dialog. Furthermore, there seems a font-size incompatibility between Mac and Windows, whereby Mac fonts have be scaled down 20% to “look right” in well-designed track layouts designed on Windows; thus, when you choose a 16-point font in the Mac native font dialog, you will see “20”, not “16”, appear in the custom height field of the TLEdit text properties dialog, and the sample will appear 20% smaller that the number in the dialog would suggest.

(END)