

Following is a few use cases that I put together to show how I have used the TpOPs program with TrainPlayer.

#### Use Case 1 -- A "Roundy Round" layout

In this case, I will refer to the St. Paul layout from the Kalmbach collection. This layout is a continuous, oval layout with no terminal points or yards. There are sidings for industry operations, however.

#### THE TRAINPLAYER LAYOUT

In Figure 1, you can see a screen shot of the layout; I have disabled the background image and enabled the stations view. Notice how I have put a station at each industry terminal and labeled it. There is also a station defined on the main track called "Start/End"; this station will be used as a starting station for my switch list, whereby all cars considered to be in that station will be "picked up" by default, and used as candidates for the successive stations in the switch list.

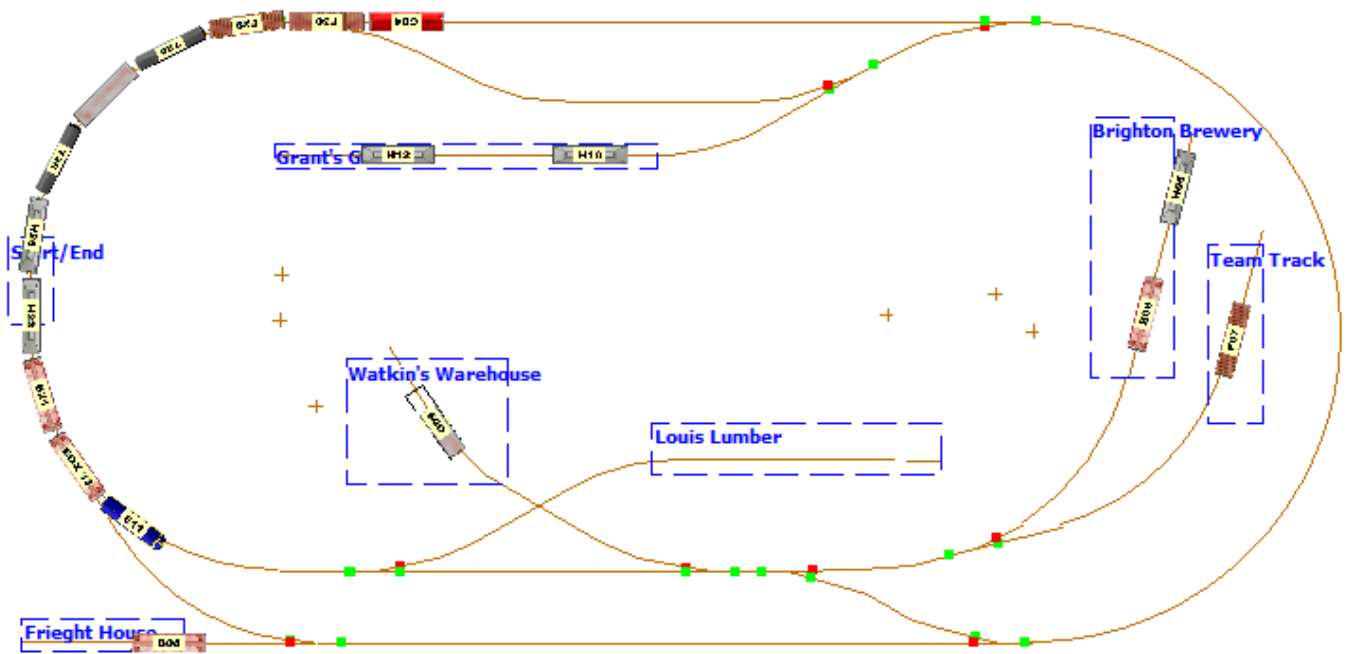
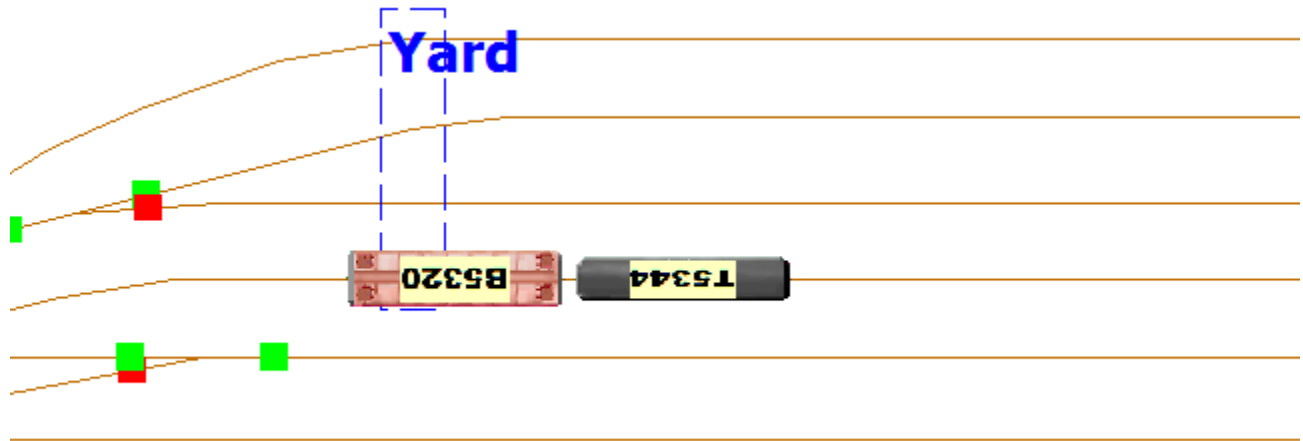


Figure 1: Screen shot of the St. Paul layout in TrainPlayer with the background image disabled, and the stations view enabled.

A note about the stations: You'll notice that some of them are not stretched far enough to cover the full track that would be considered part of the station. For instance, the Team Track may have a defined maximum of 6 cars, but the station perimeter is smaller than that. TpOps will consider a car to be in a station if any part of the car falls within the perimeter of the station, or any other car in that car's train falls within the perimeter. Hence, if I had a train of 6 cars, and only one of those cars' footprint was within the station perimeter, then all cars would be considered in the perimeter.



*Figure 2: Example of a station encapsulating several tracks in a staging yard.*

Consider the yard in Figure 2 where 4 yard tracks are all touched by the "Yard" station. If the operator is careful to have all cars be part of one and only one train within each track, then all cars in all tracks will be considered within the "Yard" station.

Caution: It is possible in TrainPlayer for a car to technically be in more than one station. For instance, if the "Team Track" station were to be larger to the right, toward the main line at the right, then cars passing thru on the main could go thru that station. This has to do with how TrainPlayer calculates if a car is in a station based on its coordinates on the screen; I'll save you the geometry lesson for now. Just beware.

#### CREATING A NEW TPOPS CONFIGURATION

Every time you create a new TpOps configuration, you will need to associate it to a TrainPlayer (ie. .rrw) file. The TpOps program will not change any data in the TrainPlayer file; it only reads the file to determine what stations exists, and which cars are in which trains and stations.

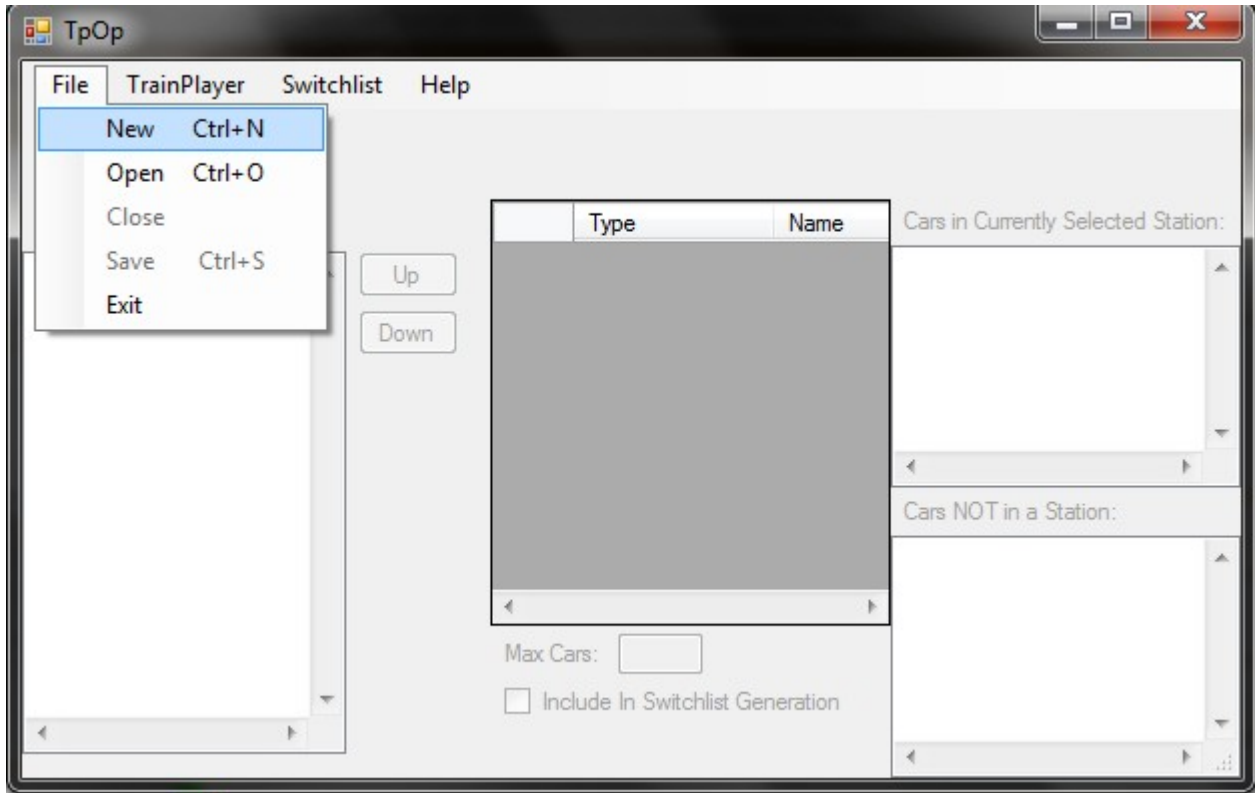


Figure 3: Example of user creating a new TpOp configuration.

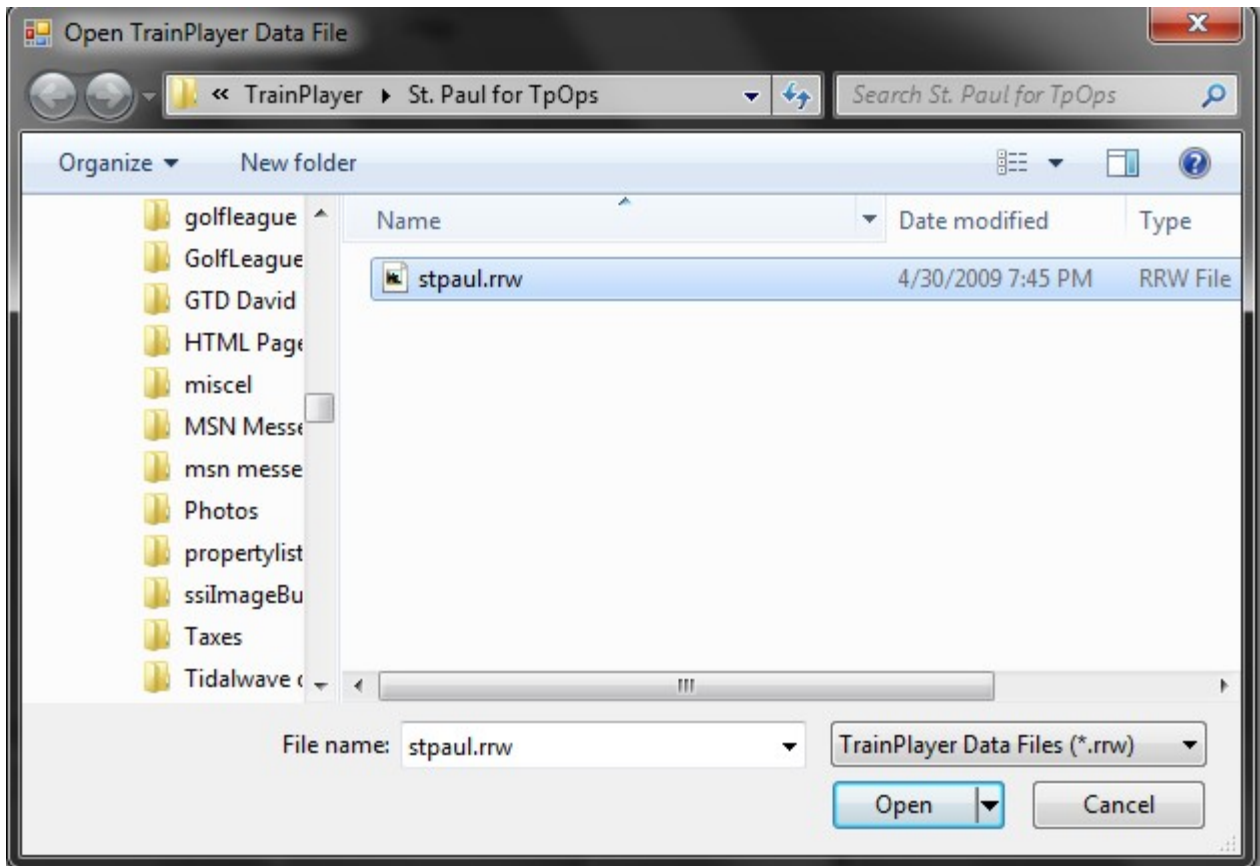


Figure 4: TpOps prompts for which TrainPlayer data file to associate with the new TpOps configuration.

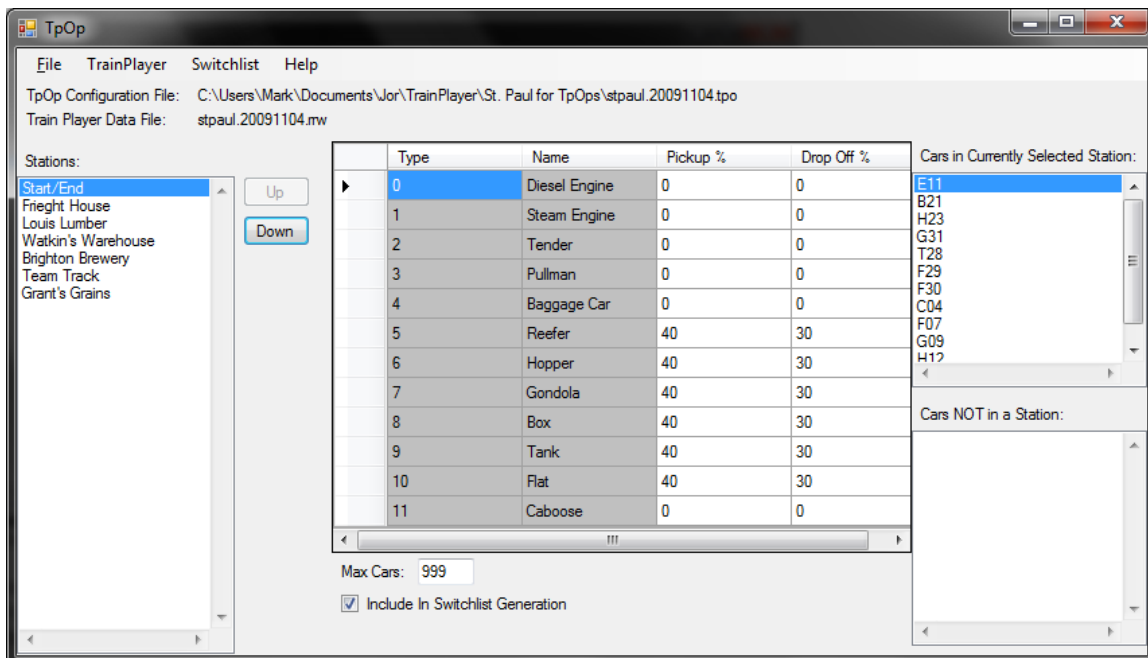


Figure 5: A new, but yet unsaved, TpOps configuration.

After creating a new TpOps configuration, you will notice at the top of the form that the "TpOP Configuration File" label is blank; this is because the configuration has not been saved yet. Once saved, this label will be filled. You can save by navigating to the "File / Save" menu item.

In the Stations pane on the left side of the form, you see a list of the stations that are defined in the TrainPlayer data file. These stations can be reordered if desired using the "up" and "down" buttons. This will be the order that the stations are visited in the final switch list.

Clicking on a station will cause the data in the middle and right panes to refresh, and show data based on the selected station. The middle pane's data can be modified to adjust probabilities for pick ups and drop off's based on the type of car, and one can also indicate the max number of cars and whether the station is to be considered or not.

### GENERATING THE SWITCH LIST

After entering the configuration data, you can generate a new switch list based on it. In the "Switchlist" menu, you will see three choices:

- Generate Switchlist -- pops up a dialog in which you can generate a switchlist and review it before saving it to a results file on the hard drive; you can regenerate as many times as needed until a suitable switchlist is generated.
- View Switchlist -- pops up a dialog in which you can open a saved switchlist results file for viewing.
- Compare Switchlist Results -- pops up a dialog in which you can specify a saved switchlist results file, and a saved TrainPlayer data file (presumably after performing the switching as directed on the switchlist and saving the new TrainPlayer session). Then you can compare the expected results of the switchlist with the actual state in the TrainPlayer data file. The dialog will indicate which cars may be outside of expectation.

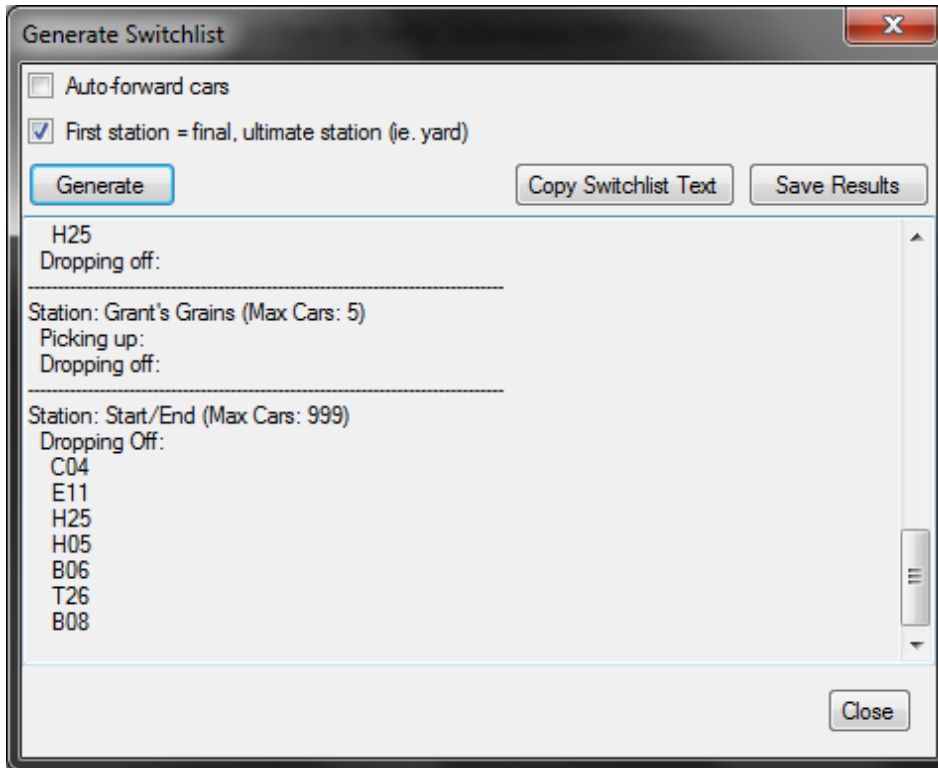


Figure 6: Generate Switchlist popup dialog

In figure 6, you can see an example of the Generate Switchlist dialog that I would use for this use case. As you can see in figure 5, I have the "Start/End" station ordered first, and when generating the switchlist, I've indicated to have the first station be the last station. In this scenario, then, the first station will effectively pickup all cars (ie. The probability grid is ignored), and the last station (which is the same station) will drop off all cars (again, ignoring the probability grid). Then successive stations will react according to the probability grid.

#### Use Case #2 -- A Point-to-point Layout

- Create a TpoP configuration for each direction of the point-to-point... ie. One for West, and one for East
- Only visit certain stations for each direction, maybe determining that only trailing point switches (ie. No facing points) are performed.
- [Editor's note: still working on this section.]