TRAINS.EXE Readme file

This application is an experimental, yet fully functional, version of a modern CRT based dispatching system I setup on the All Points North Model RR club layout near Houston, TX. The program was written in Microsoft's Visual C++ 6.0 and designed to run on Windows 9x, 2000, ME, and NT 4.0 platforms. The application needs a fairly fast system ≥ 400 MHz while polling the C/MRI and to execute the mouse events as close to real time as possible. Slower systems will cause mouse events, e.g. turnout throws on the screen, to lag behind the actual action of the user since the constant serial port communication loops are using up a majority of the processor clock cycles leaving precious little clock cycles available to handle the mouse interrupts. Faster processors and/or multi-processor workstations have enough bandwidth to make both the mouse events and serial communications appear to execute in a seamless manner. The track schematic is designed to be run on dual 15 inch monitors (see my mediocre quality photos) configured side-by-side, hence the need for scroll bars when viewed on a single monitor and repeated information at the top of each left and right screen. Several pictures have also been added to files APN\ section to better explain the multi-monitor installation. The installation currently only uses a single RS422 serial based 24-line node with 2 output and 2 input cards. A file called DEFAULT.DAT has also been added to the APN\Trains folder and is necessary for the Trains.exe application to work properly (see more info on the default.dat file below).

Once you open the application you may see that the black background area ends prematurely and does not reach all the way to the bottom of your monitor. The reason for this is because the application was written to fit exactly on two 15 inch color monitors with 800 x 600 resolutions without any scroll bars being visible. As stated above, the scrolling view enables development and viewing of the application while using only a single monitor. The video card controlling the two monitors is a Matrox Millineum G450 Dual Head video card.

Copy all the files in the Trains directory over to your drive and then run the Trains.exe file. The application is predominantly mouse driven and controls fundamental dispatching functions such as turnouts and signals. In addition, there are several troubleshooting tools which will be discussed below. To start the continuous polling of the C/MRI, you first click the [Comm Port] in the top level menu and then click on [Settings] in the drop down menu. This will bring up the COM port settings dialog box which you can use to set the appropriate port number and baud rate. The COM port data is persistent between program runs so you will not need to reset this information each time you run the program once you configured it for your particular system. However, you will have to go through the connect sequence each time you run the application to get the remainder of the menu items enabled. This leaves the COM port free for use by other things, like an external modem, when you are not directly working with the C/MRI hardware. Click [Ok] in the Communications Port Settings dialog to connect to COM port. Once you successfully connected the application to an available COM port on your computer, a message box will pop up telling you so. Click [Ok] and the remainder of the menu items will now be enabled and ready for use. To begin the constant polling cycle with a connected C/MRI system, click [Usic] in the top-level menu, then click [Start] in the drop down menu. If everything is set correctly, you should see both the communication LEDS on the Usic and the RS323/RS-422 -485 card flashing in sync with the communication packets. To stop the application from communicating with the C/MRI you should simply click on [Usic] in the top level menu and then click on [Stop] in the drop down.

Once the communication has begun, you may use the mouse to control turnouts and signals. The turnouts can be thrown by RIGHT clicking on them. The signals are called by LEFT clicking on them. To "drop" a called signal, simply DOUBLE CLICK on that signal and a dialog will appear asking if you really want to drop the signal. click [Yes] and the signal called will be dropped. The signal drops instantly as I did not see the need to incorporate the need to "run time" on the interlocking before it becomes available for reuse since the application was experimental. If the OS or CP (Control Point) is not occupied but the block ahead is occupied, then you may still call the signal, which will cause the signal icon and corresponding turnout to flash at a rate of approximately once per second until the block clears. Once the block clears, the signal, turnout, and block will line up as solid green. To simulate this without actually triggering a respective occupancy in the field, you may RIGHT click on a corresponding block and select "Occupied" in the short cut menu that pops up. The block on the screen will now turn to red to show occupancy. Call a signal into the block to start the flashing procedure, then RIGHT click on that same block once again to activate the short cut menu and select [Clear]. The signal, block, and turnout should now all go to solid green. Note: The "Add Symbol" selection on the short cut menu is not yet functional.

In addition, there are six rectangles at the bottom of the screen to toggle between local and dispatcher control of the turnouts. Left click within the boundaries of the rectangle to toggle the turnout control state. The rectangle text and background color, in addition to the turnout object color up on the track diagram will change. The five rectangles grouped near the middle of the view will only work if the turnout or cross-over OS section is not occupied. A warning dialog will pop-up if the you try to toggle the turnout control state if it lined for a route or it is occupied. The rectangle on the bottom left of the view containing the text [All TOs <CTC>] will release all turnouts to local control or vise versa regardless of whatever state of occupancy or signal line-up they are in. This control feature comes in handy during demos when others want to do local switching while the C/MRI system remains running. Again, this was a demo program so only five turnout controls actually have this local control lock-out capability via small custom-made relay circuits installed near the SMC cards.

Also worth noting are the tools available for verifying and trouble shooting 24 line input and output cards. Click on the top level [Tools] menu and a selection of 4 different tools; walk test, wrap-around test, toggle bit, and read card will appear in the drop down menu. Note: The walk test does not have a different menu item for 24 and 32 bit output cards since that information is taken from the USIC information parsed from the *.dat file. A default data file called...well... default.dat is provided for an initial sample. The tests listed under the [tools] menu MUST have the proper card orientation correctly defined in the *.dat file to work properly for your C/MRI installation. This file can be changed or modified to match your particular system using any text editor, e.g. Notepad.exe, and comments are included in the file to assist in making changes. To add your own remarks, simply add a semi-colon at the beginning of your comment line. The Trains.exe application makes the card-type calculations and fills the program variables based upon the information in the *.dat file.

Lastly, about the only items that are fully coded under the [File] drop down menu is the [Open] and [Exit] menu options. All of the other files drop down menu options are not yet coded. The [Open] menu option will allow you to open a different USIC and I/O card configuration DAT file without having to restart the application. This feature is useful especially when you want to test some recently assembled I/O cards but do not want to have to remove all the other cards from the I/O card motherboard or change the card type variables within the code. You can simply make your own test.dat file which supports only the cards and USIC you want to test.

For further assistance please refer to the Help file, Trains.hlp, which is quite thorough in explaining the features of the application.

Enjoy!
Scott Kurzawski