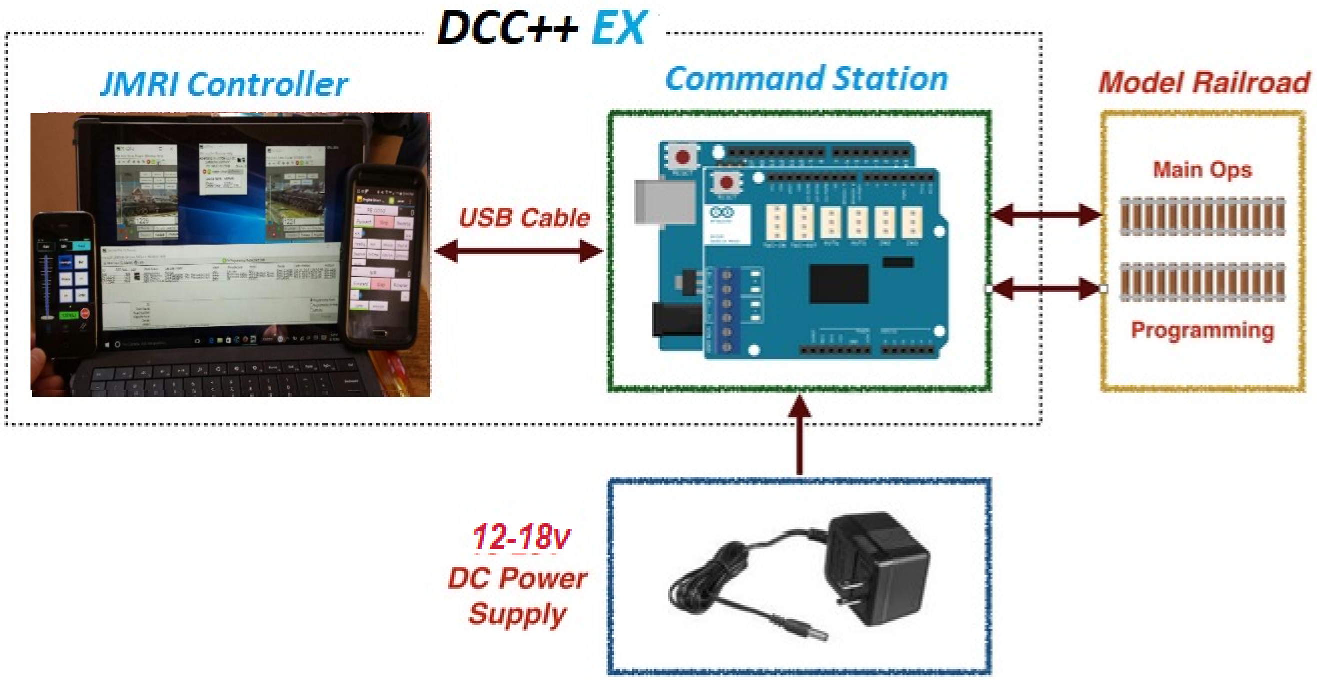
DCC++EX

A Complete Open-Source DCC Command Station
and Interface for Operating Model Railroads

DCC++EX Command Station
Arduino Microcontroller
&
Java Model Railroad Interface
(DecoderPro)

'Getting Started Guide'



DCC++EX Command Station 3.1 and Decoder Pro 4.24 - Getting Started Guide

Version 1.0 August 30, 2021 by Kevin C Smith, © Copyright 2021

Contents {the highlighted website connections in this pdf document work best when viewed through an internet browser.}

DCC++EX Command Station 3.1 and Decoder Pro 4.24 Getting Started Guide

Version 1.0 August 30, 2021 by Kevin C Smith,

Introduction

DCC++EX is a free Open-Source software solution for the Arduino based microcontroller & motor shield platform developed by the DCC++EX Extended development team and was first released in November 2020 as DCC++EX v 3.0. See DCC++EX Website
DCC++EX is more NMRA DCC compliant and supported by JMRI Java Model Railroad Interface with DecoderPro & PanelPro.
The JMRI suite is also available as a free download on PC personal computers and Linux based Raspberry Pi 3 or Pi 4 processors.
Together they form a complete Digital Command Control DCC Command Station and a 'DCC Engine Decoder Programming Station'.
DCC++EX is capable of operating a full layout by itself or by utilizing the control features of JMRI Decoder Pro and Panel Pro.

DCC++EX Features;

- Programs virtually all NMRA compliant DCC decoders
  - 2-byte and 4-byte locomotive addressing
  - 28 and 128 step speed throttling
  - Activate/deactivate all accessory function addresses 0-2048
  - Simultaneous control of multiple locomotives
  - Control of all cab functions F0-F28, F29-F68
  - Programming on the Main Operations Track
    - Write configuration variable bytes
    - set/clear specific configuration variable bits
- Advanced features; operating turnouts, control your own digital inputs, digital outputs, and analog inputs --great for reading panel switches, sensors, and occupancy detectors, as well as controlling servos & LEDs, and I2C bus for LCD/OLED displays
- Programming on the Programming Track
  - Read configuration variable bytes
  - Write configuration variable bytes
  - set/clear specific configuration variable bits
- Easy to use graphical interface with DecoderPro & PanelPro
- USB interface cable standard, or optional Bluetooth and Ethernet for connection to JMRI PC
- USB activity LED on the Arduino shows communication with the PC
- Four Motor Shield LED's turn on separately when the programming & main track power is live
- Signal pin jumper wires are no longer needed or used on the motor shield with the DCC++EX software version.
- Built in wireless WiThrottle communications and command structure support.
- Free Open Source software with on-going development & support

DCC++EX Command Station for Decoder Pro minimum requirements:

- Arduino Uno 328P R3, or an Arduino Mega 2560 R3 microcontrollers
- Arduino or compatible L298P R3 Motor Shield (booster)
- USB A to USB B cable (5vdc power & serial comm port set at 115200 baud)
- Regulated 12.5v - 18.5v DC power supply for the Motor Shield
- JMRI Decoder Pro software via online download or from a USB thumb drive.
- 16-18 gage wires & a 3ft length of track for programming and testing
- No extra hardware is required for programming sound decoders
- Optional 2.1mm Female barrel power (Center Positive) connector for + - dc connection on the motor shield

Specification/Operating Condition

Arduino Uno 328P Revision 3 or a Mega2560 Revision 3 microcontrollers
Input; 7 - 12vdc regulated with a Arduino Vin output supply current 3.3v to 5vdc.
(**Never** input DC power supply greater >12vdc into the Arduino microcontrollers female 2.1mm plug)

Arduino L298P R3 Motor Shield dual full H-bridge driver acts as a booster (2amp per channel 4amp total)
Input; 12 - 18vdc, 2a to 5 amp to the L298P Motor Shield (**Note cut Vin trace on bottom if > 12Vdc is used**)
Input voltage specifications recommend Not to exceed 18vdc.
Notes: Voltage output to main and programming should read 12vac to 16.5vac AC current for N scale to HO scale.

1. Minimum supply voltage depends upon the requirements of the decoder being programmed.
   In general, it is safer to use as low a voltage as possible in case of problems with a newly installed decoder.
2. DCC++EX will remove track power if output current exceeds 250mA standard as measured 100ms after applying power.
3. Arduinos are protected against reverse polarity connection but will not work unless the polarity is correct.

Note: Arduino DCC++EX Command Station has a new <1 JOIN> feature which logically joins the Programming track to operate as a Main track which allows a Loco to be 'Driven Away' onto the main line. Care should be taken to make sure that '+ -' polarity is maintained when wiring the two insulated sections with dual gaped nylon joiners in between with +v Right rail and -g Left rail.

For more information on how to build a Arduino microcontroller DCC++EX Command Station and on JMRI Software please click on;
Reference Arduino DCC++ EX Building & Setup DCC++EX Getting Started
Reference Arduino Assembly Arduino Command Station Assembly
Reference JMRI Software Java Model Railroad

DCC++EX Command Station & Software Notes

At this point it is **assumed** the Arduino DCC++ Command Station is built and that DCC++EX Command Station Software v3.1 [CommandStation-EX.ino] firmware or later has been uploaded onto the Arduino Uno or Mega microcontroller either manually using the Arduino Interactive Development Environment IDE editor 1.8, or with the new Automated Command Station exInstaller.

JAVA Model Railroad Installation steps

This document gives brief instructions for installing it on Windows Operating Systems 8 & 10, and using new DCC++EX features. For instructions to install Decoder Pro on Mac iOS & Linux operating systems, please refer to the Install Guides on the JMRI website at JMRI Downloads & MSteveTodd JMRI Raspberry Pi Image.

The following simple install steps are required on your Windows PC before you can use JMRI for the first time:

- Install Java Runtime 1.8 or newer and the JMRI software release 4.24 or newer on the PC.
- Connect USB A to USB B Cable from the PC to the Arduino DCC++ Command Station USB port
- Plug 12-18Vdc power supply to the Top right Motor Shield vin, gnd. "*NOT to the lower Arduino processor board"*
- Connect wiring A to the Main track and wires B to the Programming track.
- Start Decoder Pro from the PC desktop Icon. {Reminder; Turn Off the hibernate or sleep mode on the PC display}
- Edit Decoder Pro Preferences & Connections by choosing manufacturer DCC++ and then enter your Com Port #.

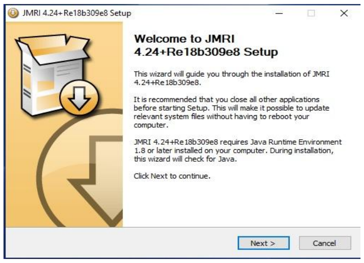
Installing Java & JMRI Decoder Pro Software

Java runtime and Decoder Pro can be installed from an online download JMRI.zip file, or from a USB Thumb drive.
Download and install Java Runtime 1.8 or later on the PC, JAVA Runtime
Download and install JMRI 4.24 or later onto the PC JMRI Downloads
Optional download JMRI install guide Windows Help JMRI Windows Install

Browse to the download directory specific to your operating system to find the Java 1.8 and JMRI 4.24 and Install them.

After JMRI is installed, in the future you can check for updates and download newer releases while in Decoder Pro through the Help > 'Check for Updates' screen.

JMRI Installation Pane



Welcome to JMRI 4.24+Re18b309e8 Setup

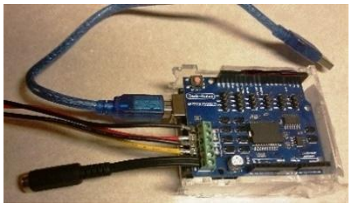
This wizard will guide you through the installation of JMRI 4.24+Re18b309e8.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

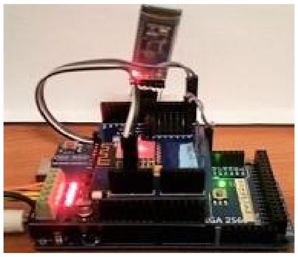
JMRI 4.24+Re18b309e8 requires Java Runtime Environment 1.8 or later installed on your computer. During installation, this wizard will check for Java.

Click Next to continue.

DCC++EX Command Station
Uno + L298P Motor Shield



DCC++EX Command Station
Mega + L298P M S + WiFi Shield + Bluetooth



Connect the USB cable to PC and plug in the External DC Power Supply to the Motor Shield.

Power is supplied to the Arduino Uno & or Mega microcontroller board via the USB B cable connection with 5vdc.
The Separate DC Power connects to the Motor Shield to the right two terminal block connectors labeled +vin & -gnd.
The Programming & Main track power connections are detected as AC current with the + - leads on a voltmeter.

- Main Track wire connect to the middle two terminal block screws labeled a - a+
- Programming Track wire connects to the left two terminal block screws labeled b- b+
- The DCC++EX Arduino or Deek Robot Standard L298P Motor Shields system have track shorting protection and will automatically momentarily turn power off when shorted. Other motor shields or external Boosters may or may not have internal short circuit protection, these and other motor boards may require adding separate circuit protection devices.
- Connect the 12-14vdc for Z & N scale and 16v to 18vdc for HO & G scale power (or connect a Female 2.1mm barrel plug 'Center positive') in the right two +vin & -gnd terminals on the Motor Shield and tighten the screw.
  The Arduino motor shield is protected against reverse polarity connection of the power supply but will not work unless the polarity is correct.
- Connect the USB cable to connect DCC++EX Command Station to the host computer.
  The Arduino Uno & Mega small power LEDs should be lit.
- The four motor shield LED’s will only illuminate steadily later when the Decoder Pro software power button is turned on and the serial USB is connected to the host computer.

Windows systems 8 & 10 should find and connect to the DCC++ and assign a COM port. Piece of cake.

IF needed - how to Identify the USB connection port# and baud rate.

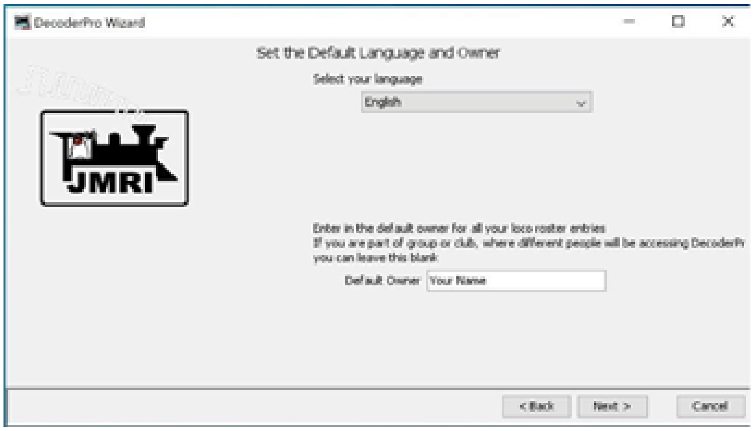
If there is an issue you can check Windows Device Manager to verify which communications Port was assigned.
Right click on the Windows Start Icon, click ‘Device Manager’.
Click the ‘>’ to open the Ports (COM & LPT) category and note the COM port assigned to Arduino {USB Serial Device COM x}
Where x is your com port number and then check and change the baud rate from 9600 to 115200 baud.
Phew, I’m glad I only have to do that twice a year!

Starting JMRI Decoder Pro Wizard setup

Start DecoderPro by double clicking the icon that was placed on the PC desktop, or on the Linux Pi microcontroller.

Assuming this is the first time you've started JMRI DecoderPro, you will see a Startup Wizard window open. click 'Next'.
For Default Owner enter <u>Your Name</u>, click Next,

Wizard Default Language & Owner



DecoderPro Wizard

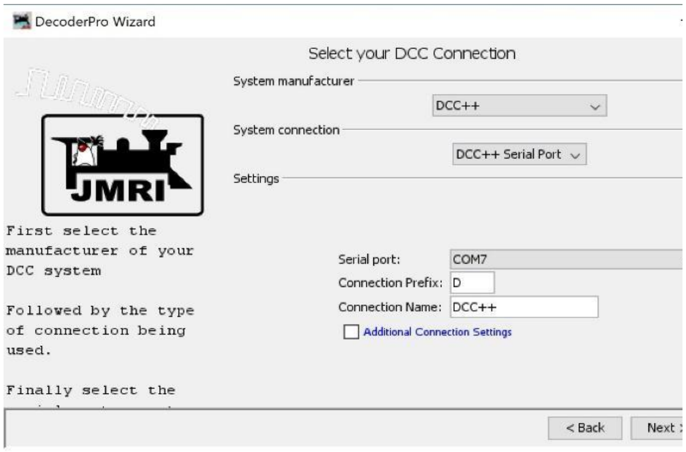
Set the Default Language and Owner

Select your language: English

Enter in the default owner for all your loco roster entries
If you are part of group or club, where different people will be accessing DecoderPr
you can leave this blank

Default Owner: Your Name

Wizard Preferences > Connections



DecoderPro Wizard

Select your DCC Connection

System manufacturer: DCC++

System connection: DCC++ Serial Port

Settings

Serial port: COM7
Connection Prefix: D
Connection Name: DCC++
□ Additional Connection Settings

First select the manufacturer of your DCC system

Followed by the type of connection being used.

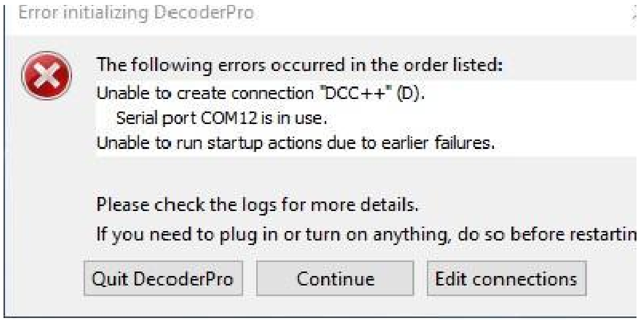
Finally select the

Click the arrow field and scroll down to Select your DCC Connection;
System Manufacture: '**DCC++**'
System Connection: '**DCC++ Serial Port'**
Serial port: select your PC USB device driver assigned **com #,** example 'COM7'
If you do not see a COM # offered, Try unplugging and replugging in the USB Cable to the Arduino.
Connection Prefix: '**D**'
Connection Name: '**DCC++**' This is displayed on the Decoder Pro menu bar as 'DCC++' tools. You may type 'DCC++EX' instead

Click 'Next' then 'Finish'
Configuration is now complete; you can change these settings at any time you need to change the com# port.
To open this Connections panel from the main DecoderPro Roster Entry panel, click 'Edit > 'Preferences' > 'Connections'.

Possible Com Error not found

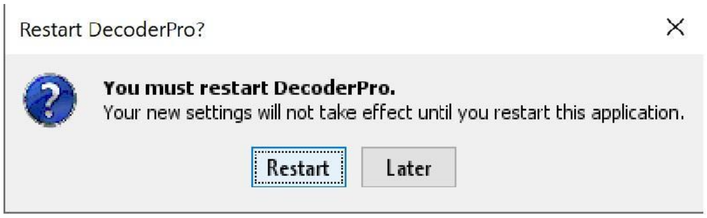
If at Any time in the future when starting up DecoderPro you get a connection error like the following, it means it's expecting the DCC++ system to have COM xx as a connection but your Device Manager assigned another USB COM port.
Click 'Edit connections' and the DecoderPro main Roster screen 'Preferences' > 'Connections' window will reopen as shown above.



Error initializing DecoderPro

The following errors occurred in the order listed:
Unable to create connection "DCC++" (D).
Serial port COM12 is in use.
Unable to run startup actions due to earlier failures.

Please check the logs for more details.
If you need to plug in or turn on anything, do so before restartin



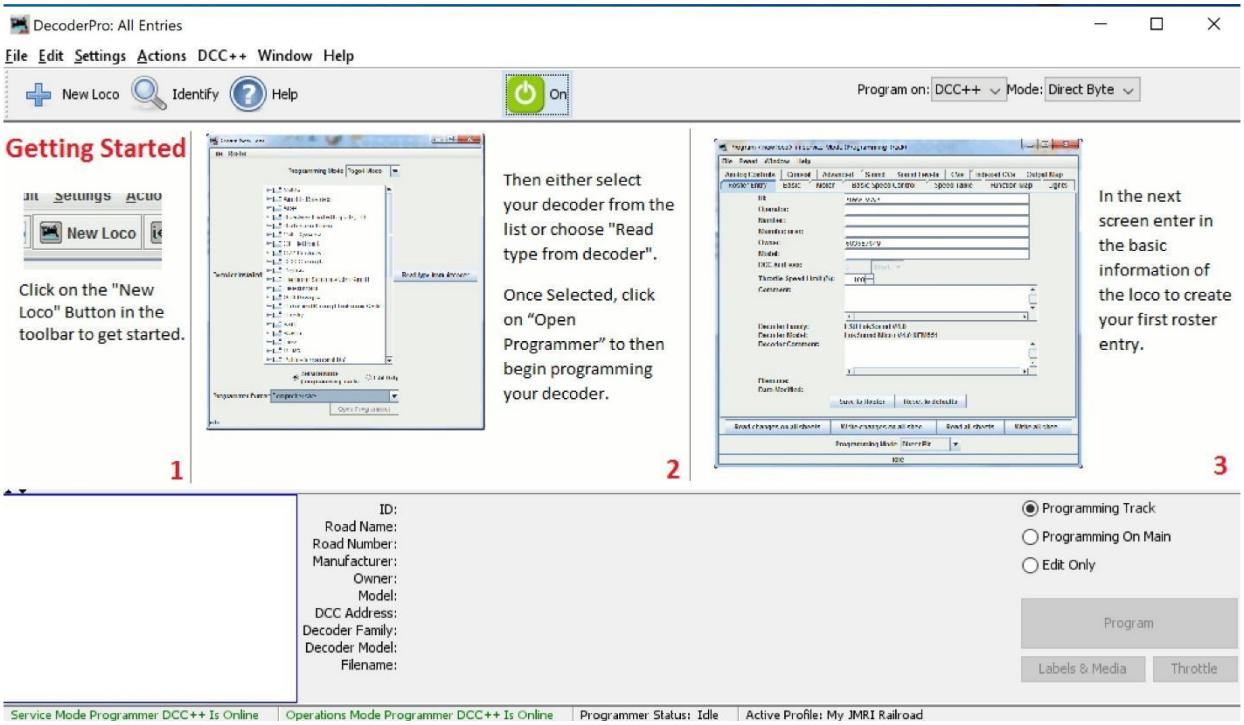
Restart DecoderPro?

You must restart DecoderPro.
Your new settings will not take effect until you restart this application.

Then just click on the Serial Port arrow and choose the new assigned Com port # offered.
Then click 'Save' and 'Restart' again and DecoderPro will start and connect to the main DecoderProAll Roster Entry panel.

DecoderPro – Roster New Loco

Initially the primary DecoderPro Roster window should appear.
Click the Power Button ON. You can now add new engines.



DecoderPro: All Entries

Program on: DCC++ Mode: Direct Byte

Getting Started

Click on the "New Loco" Button in the toolbar to get started. 1

Then either select your decoder from the list or choose "Read type from decoder".

Once Selected, click on "Open Programmer" to then begin programming your decoder. 2

In the next screen enter in the basic information of the loco to create your first roster entry. 3

ID:
Road Name:
Road Number:
Manufacturer:
Owner:
Model:
DCC Address:
Decoder Family:
Decoder Model:
Filename:

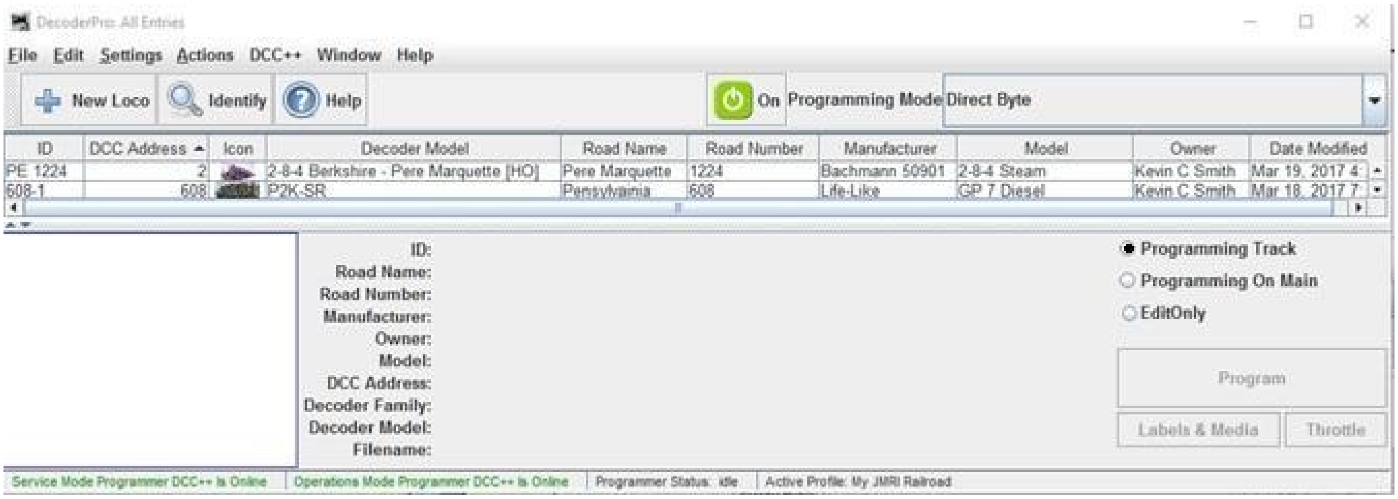
◉ Programming Track
○ Programming On Main
○ Edit Only

Service Mode Programmer DCC++ Is Online | Operations Mode Programmer DCC++ Is Online | Programmer Status: Idle | Active Profile: My JMRI Railroad

This help window above only shows up the very first time you use Decoder Pro. From then on you will only see the Decoder Pro All Entries Roster screen below. This will become your locomotive roster and will continue to fill as you create new Engines and program decoders. Press the center Power button on and choose Program on DCC++ on the top right.

Create Locomotive & Decoder Programmer Selection
To start working with a new Engine or newly-installed decoder You have two choices;
First add a new locomotive by clicking the "+New Loco" button.
Second is the "Identify" button, the programmer reads the decoder and attempts to identify it against the list of your existing engines in the roster.

Sample of Roster Entry Screen with some existing entries



DecoderPro: All Entries

Programming Mode Direct Byte

ID	DCC Address	Icon	Decoder Model	Road Name	Road Number	Manufacturer	Model	Owner	Date Modified
PE 1224	2		2-8-4 Berkshire - Pere Marquette [HO]	Pere Marquette	1224	Bachmann 50901	2-8-4 Steam	Kevin C Smith	Mar 19, 2017 4:
608-1	608		P2K-SR	Pensylvainia	608	Life-Like	GP 7 Diesel	Kevin C Smith	Mar 18, 2017 7:

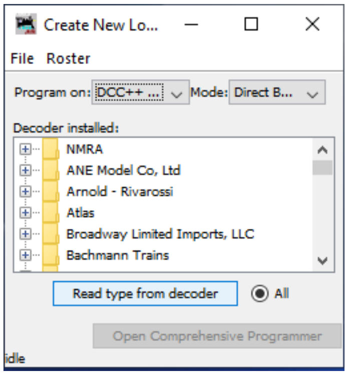
ID:
Road Name:
Road Number:
Manufacturer:
Owner:
Model:
DCC Address:
Decoder Family:
Decoder Model:
Filename:

● Programming Track
○ Programming On Main
○ EditOnly

Service Mode Programmer DCC++ Is Online | Operations Mode Programmer DCC++ Is Online | Programmer Status: Idle | Active Profile: My JMRI Railroad

To add your first and all other new locomotives,
Click on the '**+ New Loco**' button.
A 'Create New Locomotive' window will open which shows a list of decoder manufacturers.

Create a New Locomotive



Program on: DCC++ ... Mode: Direct B...

● All

idle

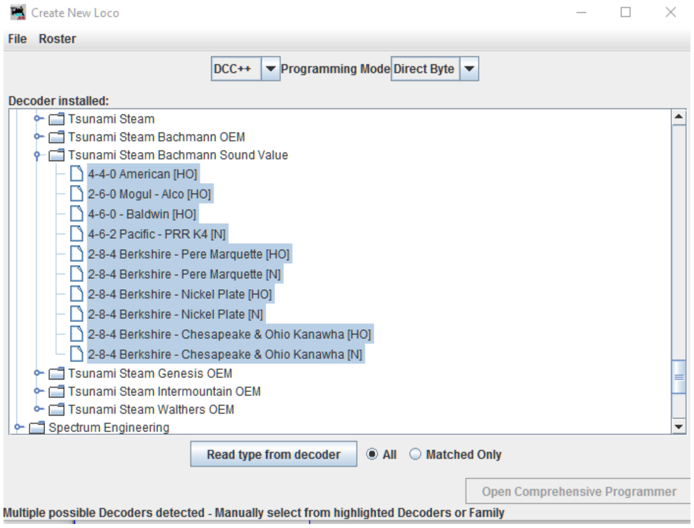
Scroll through the list manually, Or click 'Read type from decoder' button

If you know the specific decoder you want, open or double clicking the ⊞··📁 folder icon next to a manufacturer name, it will open a list of decoder types from that manufacturer. You can always manually select a decoder type in this way.

In most cases, DecoderPro can determine the manufacturer and decoder type automatically.
Place a DCC decoder equipped loco on the programming track and click on "**Read type from decoder**".

In the example below, DecoderPro has identified Tsunami Steam Bachmann Sound Value decoders. Sometimes DecoderPro can identify the manufacturer but not the exact model of decoder fitted. This is because the vendors often use the same readable ID version for multiple similar decoders. Often the difference is physical such as a steam 2-6-0 is different from an 2-8-4, or N vs HO scale or they have specific sounds for a different railroad, these do not affect the programming, and so they use a common number for that whole series of models.

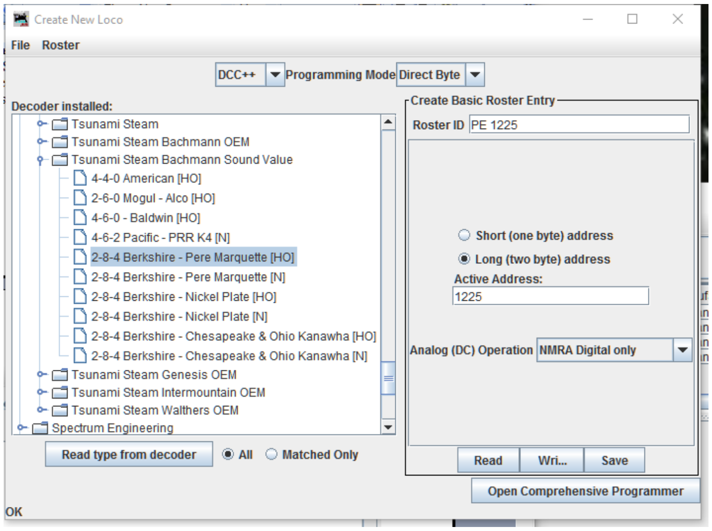
Create a New Loco


Create New Loco
File
Roster
Decoder installed:
Tsunami Steam
Tsunami Steam Bachmann OEM
Tsunami Steam Bachmann Sound Value
4-4-0 American [HO]
2-6-0 Mogul - Alco [HO]
4-6-0 - Baldwin [HO]
4-6-2 Pacific - PRR K4 [N]
2-8-4 Berkshire - Pere Marquette [HO]
2-8-4 Berkshire - Pere Marquette [N]
2-8-4 Berkshire - Nickel Plate [HO]
2-8-4 Berkshire - Nickel Plate [N]
2-8-4 Berkshire - Chesapeake & Ohio Kanawha [HO]
2-8-4 Berkshire - Chesapeake & Ohio Kanawha [N]
Tsunami Steam Genesis OEM
Tsunami Steam Intermountain OEM
Tsunami Steam Walthers OEM
Spectrum Engineering
Read type from decoder
All
Matched Only
Open Comprehensive Programmer

DCC++ Programming Mode Direct Byte

● All ○ Matched Only

Multiple possible Decoders detected - Manually select from highlighted Decoders or Family


Create New Loco
File
Roster
Decoder installed:
Tsunami Steam
Tsunami Steam Bachmann OEM
Tsunami Steam Bachmann Sound Value
4-4-0 American [HO]
2-6-0 Mogul - Alco [HO]
4-6-0 - Baldwin [HO]
4-6-2 Pacific - PRR K4 [N]
2-8-4 Berkshire - Pere Marquette [HO]
2-8-4 Berkshire - Pere Marquette [N]
2-8-4 Berkshire - Nickel Plate [HO]
2-8-4 Berkshire - Nickel Plate [N]
2-8-4 Berkshire - Chesapeake & Ohio Kanawha [HO]
2-8-4 Berkshire - Chesapeake & Ohio Kanawha [N]
Tsunami Steam Genesis OEM
Tsunami Steam Intermountain OEM
Tsunami Steam Walthers OEM
Spectrum Engineering
Read type from decoder
All
Matched Only
Create Basic Roster Entry
Read
Wri...
Save
Open Comprehensive Programmer

DCC++ Programming Mode Direct Byte

Roster ID PE 1225

○ Short (one byte) address

● Long (two byte) address

Active Address: 1225

Analog (DC) Operation NMRA Digital only

● All ○ Matched Only

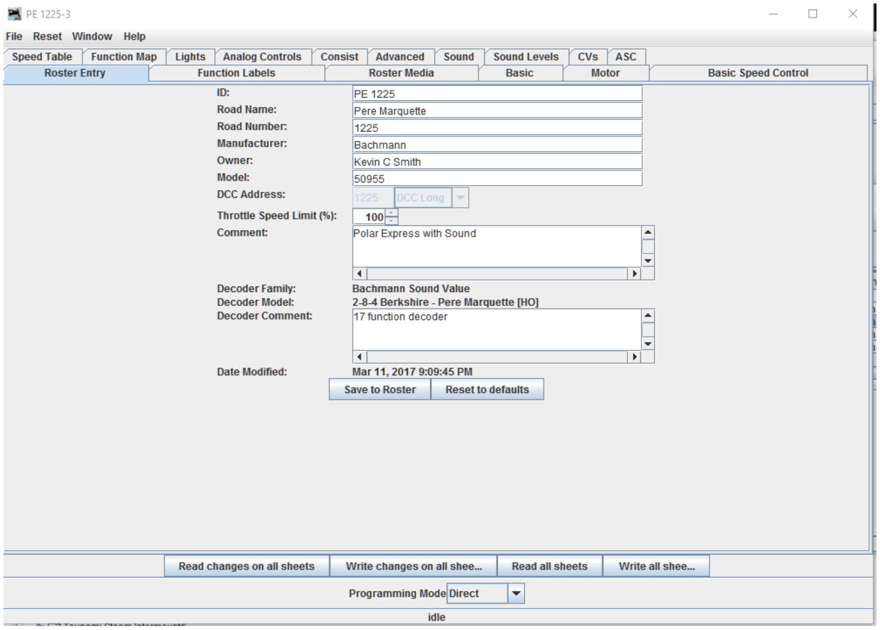
OK

In these cases, it will be necessary to select the decoder type manually, often from several highlighted possibilities.
While the programmer is talking to the decoder, a status will be displayed in the bottom of the window; "Idle" or "OK" means that things are working. If it succeeds, it will select the decoder model in the selection box. Usually, it will only be able to narrow the selection down to a few choices. Check that the right model is selected in the "Decoder Installed" box, then update the selection if desired.

After selecting the best description, enter a unique name in the top box initially marked <new loco> this will be the Roster ID name with all the settings for this loco. The Engine road number is often the best DCC choice. Most new engines default with address 3. Type a new "Unique" Roster ID: PE 1225 and assign a unique Active Address:1225 then click on "**Write**" then "**Save**" button. Once the decoder type has been selected and saved click on "**Open Comprehensive Programmer**".

The Comprehensive service mode programmer window opens with a selection of "tabs" grouped in rows below the menu bar. Clicking on a tab selects a group of CVs to be programmed, related to the title of the tab. The selection will vary depending upon the features supported by your decoder and the CVs available. On the "Roster Entry" tab you may enter arbitrary and informative details of the loco to be saved in Decoder Pro on the PC for future reference.

Comprehensive Programmer - Roster Entry



ID: PE 1225
Road Name: Pere Marquette
Road Number: 1225
Manufacturer: Bachmann
Owner: Kevin C Smith
Model: 50955
DCC Address: 1225 DCC Long
Throttle Speed Limit (%): 100
Comment: Polar Express with Sound

Decoder Family: Bachmann Sound Value
Decoder Model: 2-8-4 Berkshire - Pere Marquette [HO]
Decoder Comment: 17 function decoder

Date Modified: Mar 11, 2017 9:09:45 PM

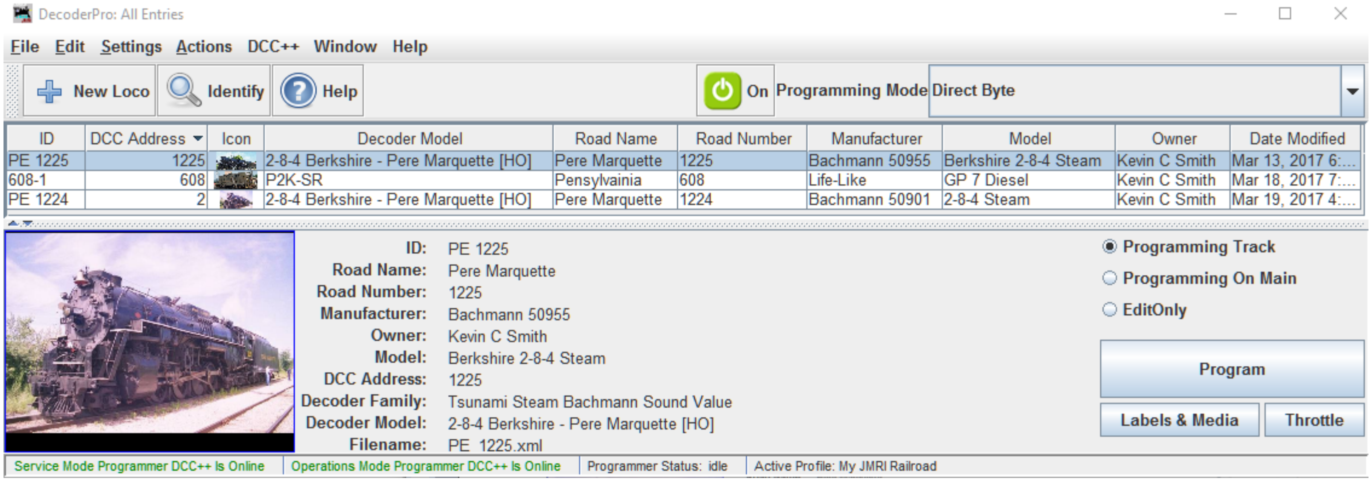
Programming Mode Direct

idle

It preloads the manufacturer ID decoder info, the DCC address and you as the owner. Everything else on this pane is optional, and for your convenience. After you've added information to the Roster, make sure to click "**Save to Roster**" on the Roster Entry tab whenever you have finished programming the decoder. Alternatively, you may wish to program a second loco with similar CV settings, by placing it on the Prog track and Write all sheets to load and duplicate all CVs into the second loco. Then you would just change the Address of the second Loco and a different Roster ID name and they will run similarly if they're the same make and model.

Note on 'ID:', when you click Save, the ID becomes the .xml file name of the roster entry created for this loco, and also the entry in the list for you to select from previously-programmed and saved loco's by using the Identify button. Name these entries wisely! Many people use the loco type and number, or Operator/road name and number, but choose a system that will be meaningful to you.
You can have a duplicate engine DCC address and in the Roster ID by adjusting space in the ID name; PE 1225 & PE1225 or by adding a hyphen# PE 1225-1 PE & 1225-2. This allows you to test two different combinations of CV settings, run them both then choose and keep the one you like best.

Notice the image of Pere Marquette 1225, this is entered later in the 'Roster Media' tab on page 13.



ID	DCC Address	Icon	Decoder Model	Road Name	Road Number	Manufacturer	Model	Owner	Date Modified
PE 1225	1225		2-8-4 Berkshire - Pere Marquette [HO]	Pere Marquette	1225	Bachmann 50955	Berkshire 2-8-4 Steam	Kevin C Smith	Mar 13, 2017 6:...
608-1	608		P2K-SR	Pensylvainia	608	Life-Like	GP 7 Diesel	Kevin C Smith	Mar 18, 2017 7:...
PE 1224	2		2-8-4 Berkshire - Pere Marquette [HO]	Pere Marquette	1224	Bachmann 50901	2-8-4 Steam	Kevin C Smith	Mar 19, 2017 4:...

ID: PE 1225
Road Name: Pere Marquette
Road Number: 1225
Manufacturer: Bachmann 50955
Owner: Kevin C Smith
Model: Berkshire 2-8-4 Steam
DCC Address: 1225
Decoder Family: Tsunami Steam Bachmann Sound Value
Decoder Model: 2-8-4 Berkshire - Pere Marquette [HO]
Filename: PE 1225.xml

Service Mode Programmer DCC++ Is Online | Operations Mode Programmer DCC++ Is Online | Programmer Status: idle | Active Profile: My JMRI Railroad

Note: Roster Program Functions (if an engine already exists you may right mouse click on a roster engine for the following choices)

Programmer Type -->	( ) Programming Track
	( ) Programming on Main
	( ) Edit Only (Allows you to edit roster text without making programing changes)
Labels & Media	Open programming page to select either Label Tab from function keys, or Roster Media Tab for adding pictures.
Throttle	Open a JMRI Throttle and load this engine number
Duplicate	Duplicate the selected roster entry to use for the same type of locomotive.
Delete	Delete roster entry

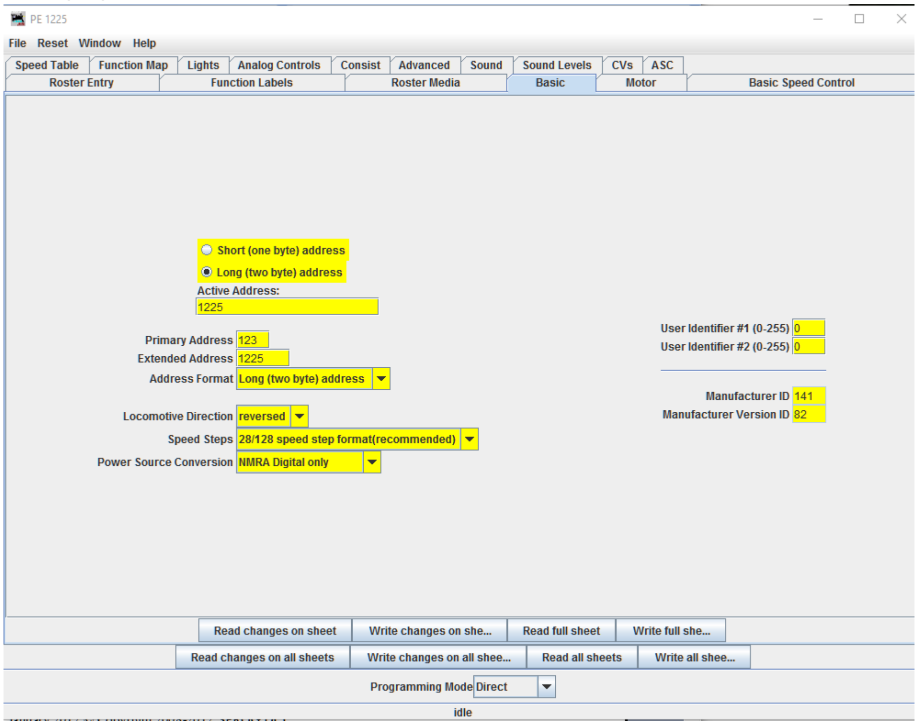
You can also double click on any of the fields in the Roster and directly edit the information in that field.
Now, highlight the engine on the roster you wish to program, then click (@) Programming Track then the 'Program' button.

Programmer Detail

Click the ‘Basic’ tab and you will see the most essential settings for the decoder including the address. Initially, all of the data fields are colored yellow to show Roster stored data and no data has been read from or written to the decoder yet. There are read and write buttons at the bottom of the window. Click ‘Read full sheet’ to read the data for the current tab from the decoder. It will Read CVs 1, 29, 17, 18, 19, 7, 8, 105 and 106 and fill in these values from the loco. Notice the engine pulse forward on the track.

Careful, clicking “Read all sheets” will read every CV from the decoder and may take a considerable time to complete from minutes to half hour or more for a large indexed decoder and increases the chances the decoder may error in reading.

Changing Cv’s



○ Short (one byte) address
● Long (two byte) address
Active Address: 1225

Primary Address 123
Extended Address 1225
Address Format Long (two byte) address

Locomotive Direction reversed
Speed Steps 28/128 speed step format(recommended)
Power Source Conversion NMRA Digital only

User Identifier #1 (0-255) 0
User Identifier #2 (0-255) 0

Manufacturer ID 141
Manufacturer Version ID 82

Programming Mode Direct

idle

Function buttons on each of the screens will read values from or write values to the decoder.

For example, to do a basic configuration, enter values in the fields press keyboard enter key then click “Write changes on sheet”
The variable fields are color coded to represent their states:

Edited
You've changed this value, and it is ready to be written.
This is shown as orange, since the value differs from what's in the decoder.

From File
The field contains values read from a locomotive roster file.
This is shown as yellow, since we're not certain that the file agrees with the decoder contents.

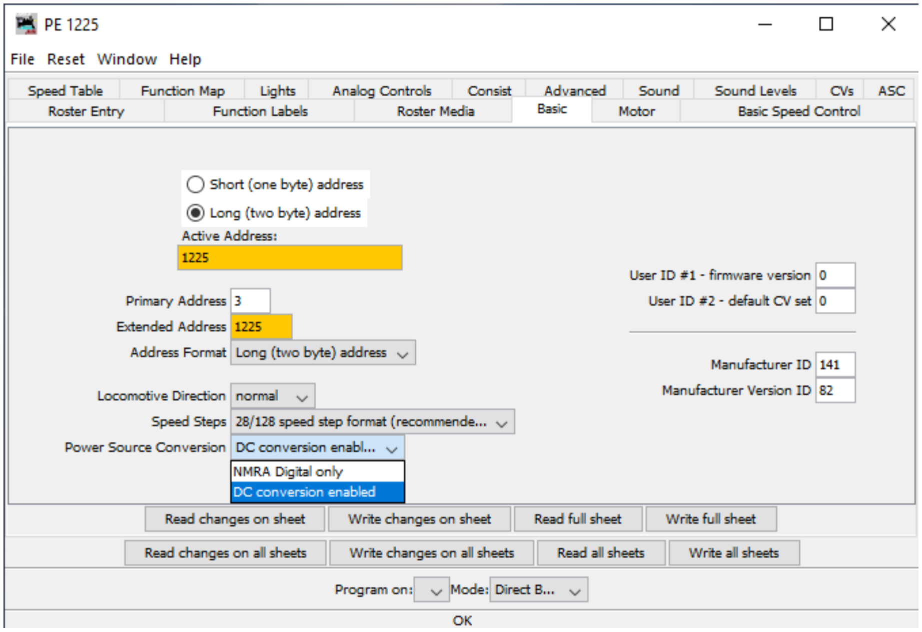
Read
The value shown has been read from the decoder. This is shown as white to indicate that the value is trustworthy.

Stored
The value show has been written to the decoder. This is also shown as white.

Unknown
If something goes wrong with the read or write process, we might have a completely untrustworthy result.
This results in the variable's state being marked as UNKNOWN and shown as red.
Status error 308 means the decoder is not responding by acknowledging to read requests, or no loco is on the track.
Status error 306 Timeout talking to Command Station. Check your wire connection and whether you’re in the correct mode.
Status error 303 may mean you are attempting to "read" while you’re in programming on the Main.
List of JMRI errors Decoder Programmer Error Messages

Each data field should return to white or the background color of the window. Check that what you see is what you expected, especially if you are reading a decoder that was already programmed elsewhere. The exact look and layout of these programming panes may vary as versions of DecoderPro or the specific decoder types are updated, but the essential information and capabilities remain consistent. To change the locos address, Select the type of address to be set (short or long), then type a new address in the ‘Active Address’ followed by carriage return or enter key. The address field will turn orange, indicating that the address has been changed but not yet written to the decoder. Short addresses are saved in CV1, Long addresses are in CV17 & CV18. and CV29.

Basic - Addressing & Manufacturers ID


PE 1225
File Reset Window Help
Speed Table Function Map Lights Analog Controls Consist Advanced Sound Sound Levels CVs ASC
Roster Entry Function Labels Roster Media Basic Motor Basic Speed Control
Read changes on sheet Write changes on sheet Read full sheet Write full sheet
Read changes on all sheets Write changes on all sheets Read all sheets Write all sheets
OK

○ Short (one byte) address
● Long (two byte) address
Active Address: 1225

Primary Address 3
Extended Address 1225
Address Format Long (two byte) address

Locomotive Direction normal
Speed Steps 28/128 speed step format (recommende...
Power Source Conversion DC conversion enabl...
NMRA Digital only
DC conversion enabled

User ID #1 - firmware version 0
User ID #2 - default CV set 0

Manufacturer ID 141
Manufacturer Version ID 82

Program on: Mode: Direct B...

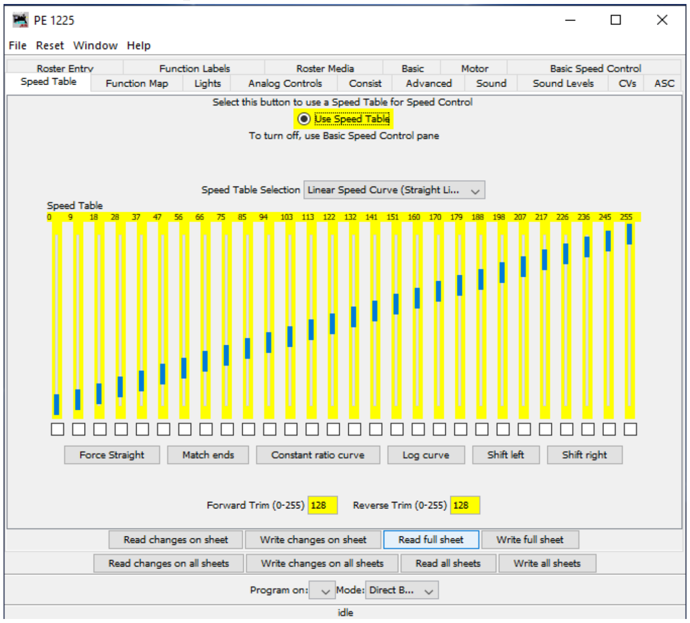
Manufacturer ID: DCC_Shortcuts Card.pdf

Click “Write changes on sheet” to write a new address. The changed field will turn red and then white as it is written to the decoder. The short address becomes primary and the long address becomes extended. You can set other basic properties such as the direction and DC conversion enabled operation on this tab. To Save your data to the Roster record, either pick from the menu bar ‘File’ > ‘Save’ at the top of the screen, or go back to the first ‘Roster Entry’ tab where there is a “Save to Roster” button.
The other tabs work in a very similar way. You may find it useful to have the decoder documentation available when setting more complex or manufacturer specific features.

Speed Table tab and Motor tab

Next, we’ll look at the Speed Control tabs that allow you to fine tune the way the loco responds to the throttle.

Click on the ‘Speed Table’


PE 1225
File Reset Window Help
Roster Entry Function Labels Roster Media Basic Motor Basic Speed Control
Speed Table Function Map Lights Analog Controls Consist Advanced Sound Sound Levels CVs ASC
Speed Table
0 9 18 28 37 47 56 66 75 85 94 103 113 122 132 141 151 160 170 179 188 198 207 217 226 236 245 255
Force Straight Match ends Constant ratio curve Log curve Shift left Shift right
Read changes on sheet Write changes on sheet Read full sheet Write full sheet
Read changes on all sheets Write changes on all sheets Read all sheets Write all sheets
idle

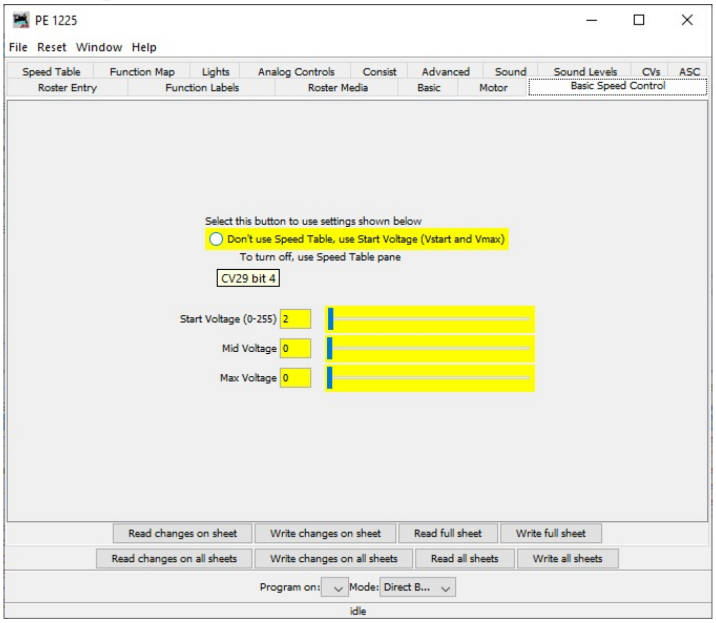
Select this button to use a Speed Table for Speed Control
● Use Speed Table
To turn off, use Basic Speed Control pane

Speed Table Selection Linear Speed Curve (Straight Li...

Forward Trim (0-255) 128 Reverse Trim (0-255) 128

Program on: Mode: Direct B...

‘Basic Speed Control’


PE 1225
File Reset Window Help
Speed Table Function Map Lights Analog Controls Consist Advanced Sound Sound Levels CVs ASC
Roster Entry Function Labels Roster Media Basic Motor Basic Speed Control
CV29 bit 4
Read changes on sheet Write changes on sheet Read full sheet Write full sheet
Read changes on all sheets Write changes on all sheets Read all sheets Write all sheets
idle

Select this button to use settings shown below
○ Don't use Speed Table, use Start Voltage (Vstart and Vmax)
To turn off, use Speed Table pane

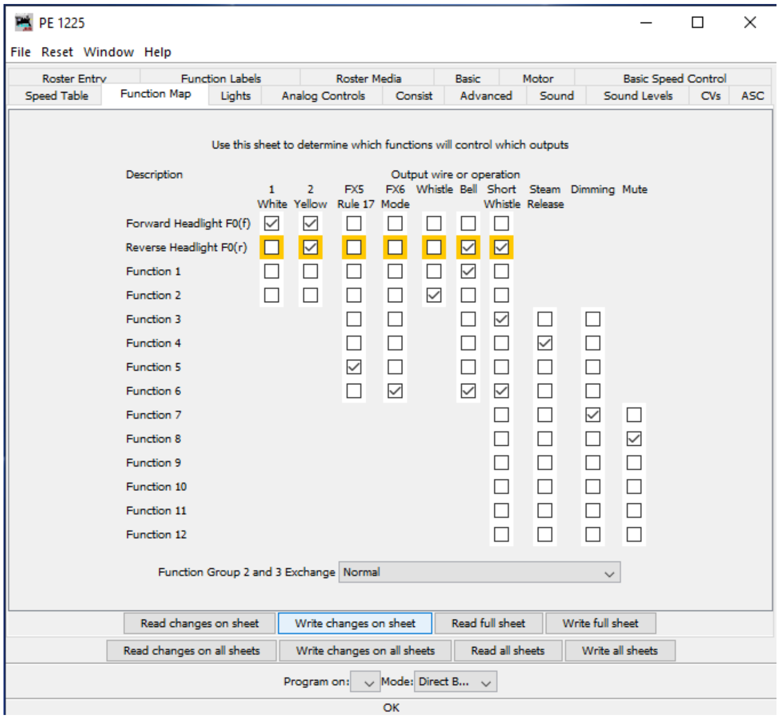
Start Voltage (0-255) 2
Mid Voltage 0
Max Voltage 0

Program on: Mode: Direct B...

This decoder example has a radio dial marked ‘Use Speed Table’ and not default to what is entered on the ‘Basic Speed Control’ tab.
The “look and feel” of these tabs will vary between decoders but most recent decoders support the use of a speed table.
The ‘Speed Table Selector’ allows you to choose from a list of vendors predefined settings by clicking on the down arrow.
It is left as an exercise for the reader to experiment with the sliders for setting the speed table and the range buttons below.
The ‘Force Straight’ option will give a linear relationship between the throttle position and the speed of the locomotive.
The “Constant ratio curve” will give little change in speed at low throttle settings, greater change at higher throttle.
Remember to ‘Read full sheet’, make desired changes then Write Changes on each sheet before moving to a new sheet.

Function Map

The Function Map allows decoders that support it to control which throttle function key is mapped to each output or operation (e.g., sound effect, lighting, mute) of the decoder.



Use this sheet to determine which functions will control which outputs

Description	1 White	2 Yellow	FX5 Rule 17	FX6 Mode	Whistle	Bell	Short Whistle	Steam Release	Dimming	Mute
Forward Headlight F0(f)	☑	☑	□	□	□	□	□			
Reverse Headlight F0(r)	□	☑	□	□	□	☑	☑			
Function 1	□	□	□	□	□	☑	□			
Function 2	□	□	□	□	☑	□	□			
Function 3			□	□		□	☑	□	□	
Function 4			□	□		□	□	☑	□	
Function 5			☑	□		□	□	□	□	
Function 6			□	☑		☑	☑	□	□	
Function 7							□	□	☑	□
Function 8							□	□	□	☑
Function 9							□	□	□	□
Function 10							□	□	□	□
Function 11							□	□	□	□
Function 12							□	□	□	□

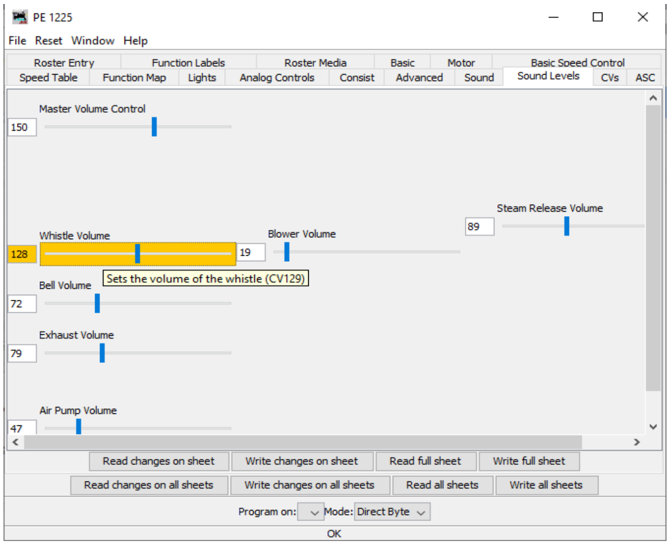
Function Group 2 and 3 Exchange: Normal

Program on: Mode: Direct B...

In this example we've 'Read full sheet' first which changes every field white after being read. Now you may like to add an automatic Bell when backing up and a Short Whistle when changing direction as seen above in the check marks added to Reverse Headlight line, and the change to orange status waiting to be written. Click, 'Write Changes on Sheet' button to only save those items in orange. The Headlights F0 functions key button must be On, in order for the sounds to activate when reversing direction.

DecoderPro comes into its own for programming the many and varied options in a sound decoder.
Here is an example of Sound tabs from the Tsunami 2-8-4 Steam engine where the volume and individual effects can be set.

Sound Level & Sounds tabs



Master Volume Control: 150

Steam Release Volume: 89

Whistle Volume: 128

Blower Volume: 19

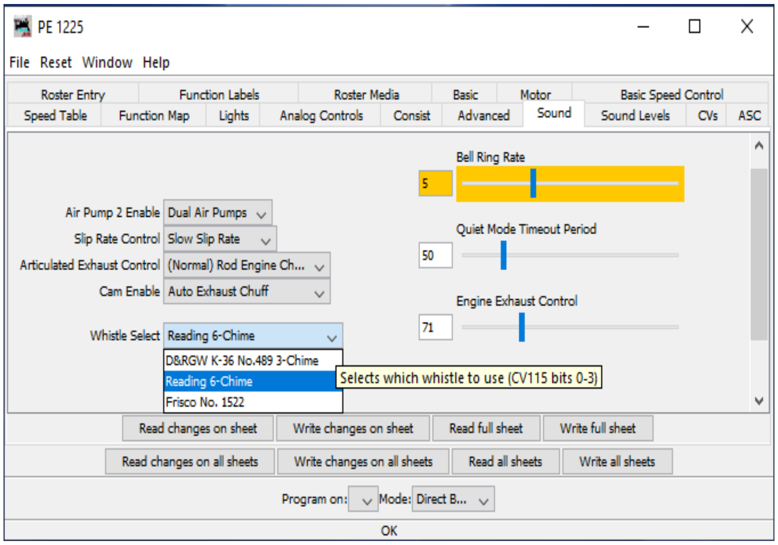
Sets the volume of the whistle (CV129)

Bell Volume: 72

Exhaust Volume: 79

Air Pump Volume: 47

Program on: Mode: Direct Byte



Bell Ring Rate: 5

Air Pump 2 Enable: Dual Air Pumps

Slip Rate Control: Slow Slip Rate

Quiet Mode Timeout Period: 50

Articulated Exhaust Control: (Normal) Rod Engine Ch...

Cam Enable: Auto Exhaust Chuff

Engine Exhaust Control: 71

Whistle Select: Reading 6-Chime

D&RGW K-36 No.489 3-Chime
Reading 6-Chime
Frisco No. 1522

Selects which whistle to use (CV115 bits 0-3)

Program on: Mode: Direct B...

Hovering over an item with the cursor will show a short description, but it is highly recommended that you read the documentation for your decoder to get the best from it with DecoderPro. Again, remember first 'Read full sheet', make your changes and then 'Write changes on sheet'.

Tip: There is a nice DCC Decoder CV Support link at the end of this document under Other Useful Links.

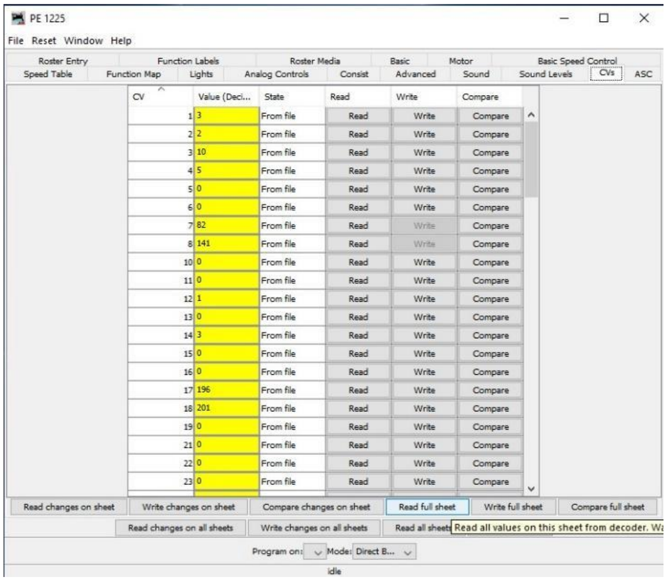
Using the New 'V' Verify command in 'Reading Full Sheet' and 'Comparing Full Sheet'

Verify 'V' Command is a new design feature in DCC++EX v3.1 which significantly speeds up CV reads & verification if the expected value is previously known, and by predicting it's value is still the same. Significantly speeding up the Command Station responses and enhancing JMRI Decoder Pro functionality, performance & speed. *Note: Programming on the Programming track is even faster with the normal mode <1 PROG> feature turned On, and the new.command < JOIN> to main line feature turned Off, as seen on pg15*

With an engine on the programming track, click on the Engine in the main Roster menu then click on radio dial (@)Programming Track, then open the 'CVs ' tab. From here you can Read, Write & Compare every individual CV in the decoder.

One of the most important and last steps is to 'Read full sheet' of All your decoders CV's, then 'File' > 'Save' them All so that you are sure you have a good record of all the engines CV's saved in the Roster.

CV's 'Read Full Sheet'



PE 1225

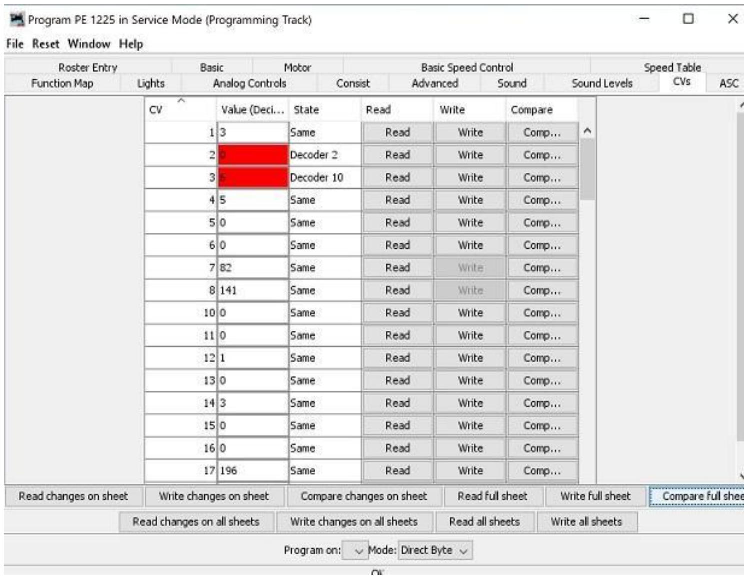
CV	Value (Deci...	State	Read	Write	Compare
1	3	From file	Read	Write	Compare
2	2	From file	Read	Write	Compare
3	10	From file	Read	Write	Compare
4	5	From file	Read	Write	Compare
5	0	From file	Read	Write	Compare
6	0	From file	Read	Write	Compare
7	82	From file	Read	Write	Compare
8	141	From file	Read	Write	Compare
10	0	From file	Read	Write	Compare
11	0	From file	Read	Write	Compare
12	1	From file	Read	Write	Compare
13	0	From file	Read	Write	Compare
14	3	From file	Read	Write	Compare
15	0	From file	Read	Write	Compare
16	0	From file	Read	Write	Compare
17	196	From file	Read	Write	Compare
18	201	From file	Read	Write	Compare
19	0	From file	Read	Write	Compare
21	0	From file	Read	Write	Compare
22	0	From file	Read	Write	Compare
23	0	From file	Read	Write	Compare

Read all values on this sheet from decoder. Wa

Program on: Mode: Direct B...

idle

CV's 'Compare Full Sheet'



Program PE 1225 in Service Mode (Programming Track)

CV	Value (Deci...	State	Read	Write	Compare
1	3	Same	Read	Write	Comp...
2	0	Decoder 2	Read	Write	Comp...
3	6	Decoder 10	Read	Write	Comp...
4	5	Same	Read	Write	Comp...
5	0	Same	Read	Write	Comp...
6	0	Same	Read	Write	Comp...
7	82	Same	Read	Write	Comp...
8	141	Same	Read	Write	Comp...
10	0	Same	Read	Write	Comp...
11	0	Same	Read	Write	Comp...
12	1	Same	Read	Write	Comp...
13	0	Same	Read	Write	Comp...
14	3	Same	Read	Write	Comp...
15	0	Same	Read	Write	Comp...
16	0	Same	Read	Write	Comp...
17	196	Same	Read	Write	Comp...

Program on: Mode: Direct Byte

Reading full sheet CV's can take as little as 20 seconds on many decoders, but several minutes on the newer TCS WOW and ESU Loksound decoders with 1000's of stepped Indexed CV's. Now 'Read full sheets' then 'File' > 'Save' when you're done.

Compare Full Sheets, fun time!
When comparing, CV's turn White if the Roster matches the decoder value and turn Red then display both values if they differ. Be sure the power is turned on and then Close & Reopen this CV's tab pane from service mode Programming Track, but this time press the 'Compare Full Sheet' button and watch the 'V' Verify work up to 10x greater speed Comparing CV's forevermore.

New ' V' Verification & Validation Feature that Speeds up Decoder Pro Reads & Compares by up to 10 times faster than DCC++ Classic Base Station..
Sample Speed Comparison of JMRI Decoder Pro and DCC++ Classic vs DCC++EX with V Command in minute : sec : ms

				Function	Speed	Sound	Compare	Approx.
	Decoder	Loco#	Basic	Map	Table	Levels	Full Sheet	# of CV's
Classic	Tsunami Sound Steam	1225	0.12.0	0.19.8	0.41.6	0.9.2	2.07.0	109+ CVs
EX	Tsunami Sound Steam	1225	0.1.1	0.1.6	0.3.6	0.0.9	0.10.5	109+ CVs

Don't blink, I'm not kidding. Guaranteed to bring a smile to your face or your money back. Oh, and Smiles are Free too.

The 'V' command can also be used directly through either the Decoder Pro Send DCC++ Command panel, or through the Arduino Interactive Development Editor IDE, Tools > Serial Monitor.

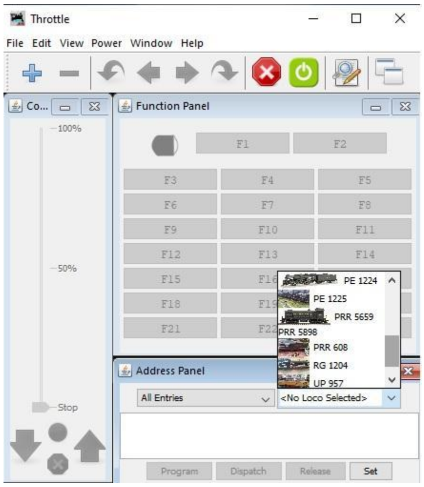
Command Structure;
V Command is <V cv value> for Byte verify, or <V cv bit 0|1> for Bit verify if it is a 1 or a 0.
<V 1 3> Guess that CV 1 was a 3, and the station acknowledges quicker. response -r 3.

If the value given matches the CV, then the -r return is much faster. If however, it does not match, then the CV is read normally for a very small overhead compared with the original <R> read command.

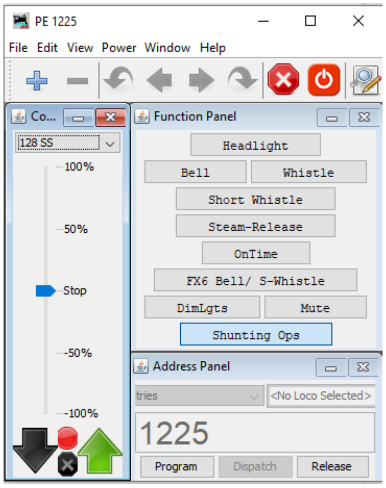
Using the DecoderPro Throttle
On the main DecoderPro Roster panel menu, click on the 'Actions' then 'New Throttle'.
You can click on loco selection arrow and pick one, or enter the loco address and then click 'Set', or press the Enter key
Normally with DCC++, loco addresses of 127 or less are short (2 digit) addresses.
Addresses 128 and greater are long or extended (4 digit) addresses. This may vary with other proprietary command stations.
Setting the loco address will enable the function buttons and speed slider specific to that locomotive.

Alternatively, on the Decoder Pro Roster panel mouse click and highlight the locomotive you'd like to use, then click on the 'Throttle' button at the bottom right hand side of the page and the engine# will prefill into the throttle.

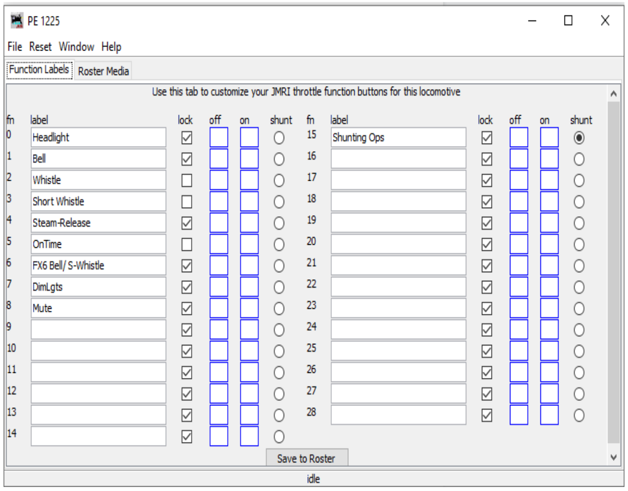
JMRI Throttle on a PC



Engine 1225



Labels & Media Panel
Function Labels & Roster Media tabs
edit the throttle function key buttons



Use this tab to customize your JMRI throttle function buttons for this locomotive

fn	label	lock	off	on	shunt	fn	label	lock	off	on	shunt
0	Headlight	☑	□	□	○	15	Shunting Ops	☑	□	□	●
1	Bell	☑	□	□	○	16		☑	□	□	○
2	Whistle	□	□	□	○	17		☑	□	□	○
3	Short Whistle	□	□	□	○	18		☑	□	□	○
4	Steam-Release	☑	□	□	○	19		☑	□	□	○
5	OnTime	□	□	□	○	20		☑	□	□	○
6	FX6 Bell/ S-Whistle	☑	□	□	○	21		☑	□	□	○
7	DimLgts	☑	□	□	○	22		☑	□	□	○
8	Mute	☑	□	□	○	23		☑	□	□	○
9		☑	□	□	○	24		☑	□	□	○
10		☑	□	□	○	25		☑	□	□	○
11		☑	□	□	○	26		☑	□	□	○
12		☑	□	□	○	27		☑	□	□	○
13		☑	□	□	○	28		☑	□	□	○
14		☑	□	□	○						

Click once on the Power icon button on the top right of the menu bar to cycle power on|off to the track..
The DCC++ command stations track power LED's will turn on when the power is live.

The 'Function Labels' pane can be edited and modified to best describe what the function key does, then press 'Save to Roster
The function keys mostly have a latching operation (or lock on). Click once to turn a function on, click again to turn the function off.
The whistle & horn are non-latching (lock off) and will only sound when the button is depressed and will stop when it is released.
You can drag and drop images of function keys into the off [ ] and on [ ] boxes which will then display on your PC throttles.
Notice the F15 key, I labeled it 'Shunting Ops' with shunt marked (*). This changes a throttle slider to show speed settings from -100, -50, 0, 50+ 100+, making it easier to perform 'Switching' and yard coupling tasks with the PC based throttle.

To control the loco speed, click and drag the slider caret.
For fine control of speed, click on the slider just above or below the caret.
The 'Idle' button allows the engine to come to a slow stop.
The 'STOP!' button below that will stop the loco but does not turn off the track power.
The Emergency X Stop button at the top right will stop all engines on the main line track but leave the track power on.

Note; Pressing the 'Program' button on the throttle will open the Comprehensive Programmer panel of the Engine# in the throttle for 'Operations mode' programming on the main line track. Which means you can only Write changes to CV's and Not Read CV's.

A Brief note on Programming on the Prog track 'service mode' vs Programming on Main track 'operations mode'.

Programming track and Main line track how do they differ?
In the original DCC++ Classic version

- Programming track can 'read & write' CV's but you can't move or hear the sound changes on the Prog track.
- Main track programming can only 'write' CV's, Not read CV's, you can listen to sounds, run & check speed settings live.

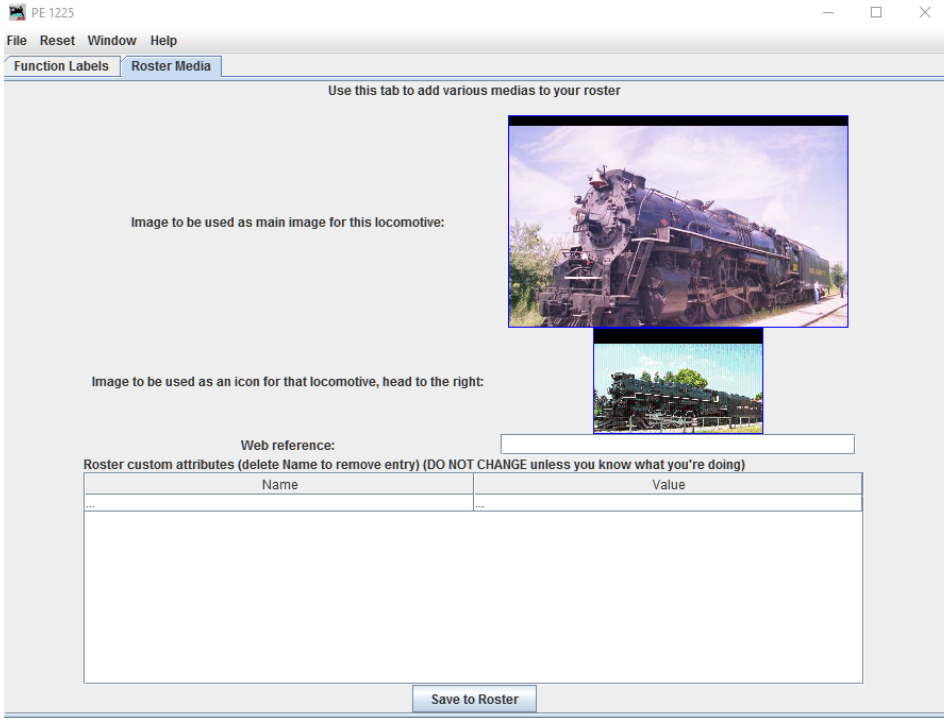
The Enhanced DCC++EX v3.1 has a new JOIN command which defies this standard and now allows the Prog track to become both at the same time! Think of it as a software version to replace a hardwired DPDT switch between the Programming & Main tracks.

When "Programming on Main POM" i.e., 'operations mod' it is possible to immediately test run a loco after writing new CVs, to test the effect of changing the speed table, function mapping or listening to the different whistles and bell sounds available.
More than one Locomotive may be present on the main track layout but only one Locomotive address may be selected on the throttle and programmed at one time. If you Program on the Main line track, care should be taken when using the 'operations mode' programmer and only attempt to change the user friendly sound and lighting CV's while programming, and leave the more difficult Consisting and BEMF control for the 'service mode' programming track.

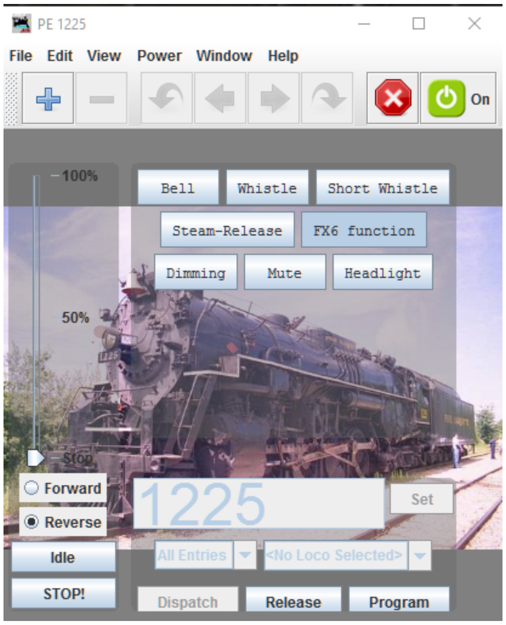
Roster Media

The second tab on the Labels & Media panel is ‘Roster Media’ which allows you to add images to your engine roster and throttles by clicking on the ‘Roster Media’ tab. Now find photos on your PC and drag & drop them into the boxes below.   The top picture displays on the PC Throttles, the lower picture displays on the Roster line under the Icon column and on wireless WiFi Throttles.

Directly access Roster & Media from the Roster by clicking on an Engine ID line, then the 'Labels & Media' button on bottom right.


PE 1225
File
Reset
Window
Help
Function Labels
Roster Media
Use this tab to add various medias to your roster
Image to be used as main image for this locomotive:
Image to be used as an icon for that locomotive, head to the right:
Web reference:
Roster custom attributes (delete Name to remove entry) (DO NOT CHANGE unless you know what you're doing)
Name
Value
Save to Roster

Click the Icon to the right of the On|Off button to display your engine on your throttle.
Or by clicking ‘View’ then ‘Switch throttle frame view mode’.

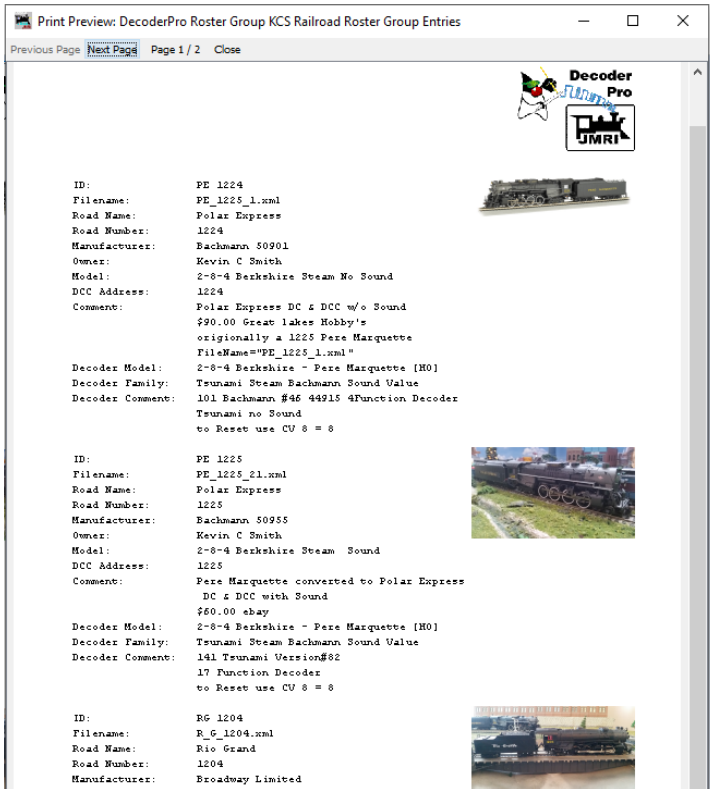


To set a different loco address, click the “Release” button.
 enter a new engine number and click 'Set', or press the Enter key.

Print Previews and Roster Summary

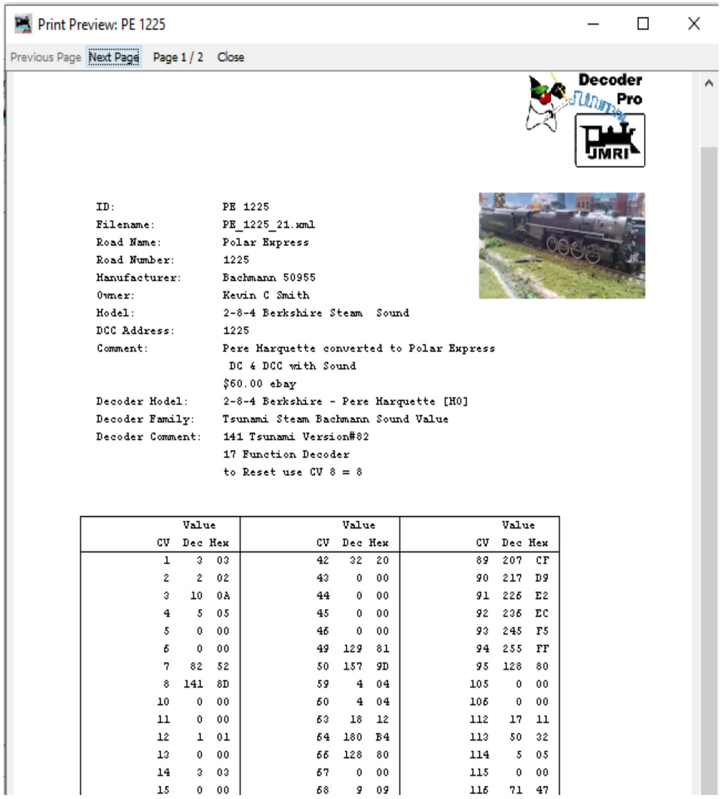
You can print Roster Entries and/or Summaries by clicking ‘File’ > ‘Print Preview Roster’ > ‘Summary’& ‘Selected Entry’
This gives you a nice summary sheet of your locomotives and a good way to compare how CV’s are set in your Engines decoder versus the way the decoder came with defaults shown on the manufacturer's published data sheet.

Engine Roster Summary


Print Preview: DecoderPro Roster Group KCS Railroad Roster Group Entries
Previous Page
Next Page
Page 1 / 2
Close
Decoder Pro
JMRI

```
ID:               PE 1224
Filename:         PE_1225_1.xml
Road Name:        Polar Express
Road Number:      1224
Manufacturer:     Bachmann 50901
Owner:            Kevin C Smith
Model:            2-8-4 Berkshire Steam No Sound
DCC Address:      1224
Comment:          Polar Express DC & DCC w/o Sound
                  $90.00 Great lakes Hobby's
                  origionally a 1225 Pere Marquette
                  FileName="PE_1225_1.xml"
Decoder Model:    2-8-4 Berkshire - Pere Marquette [HO]
Decoder Family:   Tsunami Steam Bachmann Sound Value
Decoder Comment:  101 Bachmann #46 44915 4Function Decoder
                  Tsunami no Sound
                  to Reset use CV 8 = 8

ID:               PE 1225
Filename:         PE_1225_21.xml
Road Name:        Polar Express
Road Number:      1225
Manufacturer:     Bachmann 50955
Owner:            Kevin C Smith
Model:            2-8-4 Berkshire Steam  Sound
DCC Address:      1225
Comment:          Pere Marquette converted to Polar Express
                   DC & DCC with Sound
                  $60.00 ebay
Decoder Model:    2-8-4 Berkshire - Pere Marquette [HO]
Decoder Family:   Tsunami Steam Bachmann Sound Value
Decoder Comment:  141 Tsunami Version#82
                  17 Function Decoder
                  to Reset use CV 8 = 8

ID:               RG 1204
Filename:         R_G_1204.xml
Road Name:        Rio Grand
Road Number:      1204
Manufacturer:     Broadway Limited
```

Engine CV’s Summary



```
ID:               PE 1225
Filename:         PE_1225_21.xml
Road Name:        Polar Express
Road Number:      1225
Manufacturer:     Bachmann 50955
Owner:            Kevin C Smith
Model:            2-8-4 Berkshire Steam  Sound
DCC Address:      1225
Comment:          Pere Marquette converted to Polar Express
                   DC & DCC with Sound
                  $60.00 ebay
Decoder Model:    2-8-4 Berkshire - Pere Marquette [HO]
Decoder Family:   Tsunami Steam Bachmann Sound Value
Decoder Comment:  141 Tsunami Version#82
                  17 Function Decoder
                  to Reset use CV 8 = 8
```

	Value			Value			Value	
CV	Dec	Hex	CV	Dec	Hex	CV	Dec	Hex
1	3	03	42	32	20	89	207	CF
2	2	02	43	0	00	90	217	D9
3	10	0A	44	0	00	91	226	E2
4	5	05	45	0	00	92	236	EC
5	0	00	46	0	00	93	245	F5
6	0	00	49	129	81	94	255	FF
7	82	52	50	157	9D	95	128	80
8	141	8D	59	4	04	105	0	00
10	0	00	60	4	04	106	0	00
11	0	00	63	18	12	112	17	11
12	1	01	64	180	B4	113	50	32
13	0	00	66	128	80	114	5	05
14	3	03	67	0	00	115	0	00
15	0	00	68	9	09	116	71	47

What’s New on DCC++ EX vs. DCC++ Classic

In addition to no longer needing any jumper wires on the motor shield, we have also added new software features.
Let's take a look at one of many areas that DCC++EX has made major enhancements with additional single letter Command codes and Diagnostics ‘D’ commands for interrogating DCC decoders.

DCC++ Classic had a ‘f’ command to test Function key commands on the base station.
DCC++EX has a new ‘F’ command to test Function key commands which doesn’t require a throttle register number to be sent.

DCC++ Classic had one ‘D’ Diagnostic command to test the motor shield DCCwaveforms with alternating flashing LEDs.
DCC++EX has over 15 new ‘D‘ Diagnostic commands to test Decoders, CABS, WiFi & WiThrottle traffic, Layout Control & more.

Lets focus on how to use the DCC++ Send Command & DCC++ Traffic Monitor and use a few commands to see how they work.
Then we’ll learn new Decoder diagnostic techniques by using a few new powerful ‘D ACK’ commands to improve Reading, Writing and Verifying DCC decoders.

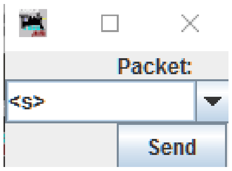
New DCC++ EX Command Codes for Diagnostics and Decoder Pro Programing

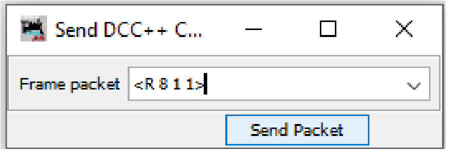
See a complete list and functional use of DCC++ EX Commands on our website;
[DCC++EX Command References] single character, track power, engine decoder operation, CAB functions, & accessories decoders.
[DCC++EX Diagnostic 'D' Commands] diagnostic commands
[DCC++EX Diagnostic 'D ACK' Commands] diagnose decoder response commands

Send DCC++ Command pane and the DCC++ Traffic Monitor pane
DCC++EX command codes are sent through the 'SEND DCC++ Command' pane and are used to better detect and read decoder CV's and then display important information on the serial 'DCC++ Traffic Monitor'.

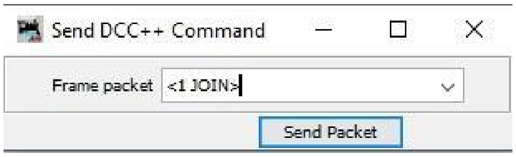
On the DecoderPro All Roster Entry panel, click on 'DCC++' then 'Send DCC++ Command'

Some examples of a few new DCC++EX commands with both symbols "<   >" and the command inside.
<s> small 's' displays the DCC++EX Command Station configuration & the status of Sensors, Turnouts & Outputs.
<R> Reads and returns the Engines address or CAB #
<R 8 1 1> Reads CV8 and returns the value of the Mfr. ID.   The 1 and 1 are any callback numbers and are arbitrary.
<1 JOIN> Joins the Programming track to a Main Ops track to allow Throttle control of F-keys, Speed & Direction.
<1 PROG> places the Programming track back to normal mode and speeds up Read & Writes and Verify's.


Packet:
<s>



Frame packet <R 8 1 1>

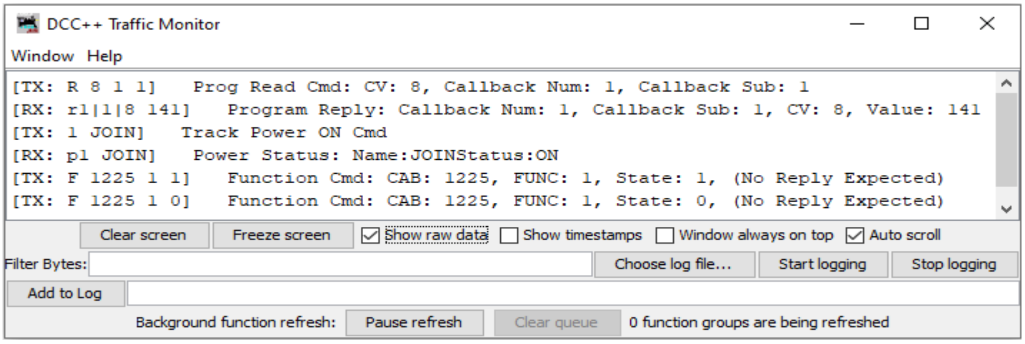


Frame packet <1 JOIN>

DCC++ Traffic Monitor
On the Main Decoder Pro Roster screen, click on 'DCC++' then 'DCC++ Traffic Monitor'
This comes in very handy when trying to identify CV values and command responses back from the engine.
You can see character commands or the throttle commands and function key commands displayed on the Monitor window below, as well as compound Diagnostic commands sent through the Send DCC++ Command panel. Some examples with their responses are;

<R 8 1 1> Read CV8 Mfr. ID and returns the Mfr. ID value: 141 - Soundtraxx Throttle-UP
<1 JOIN> the Prog track, simultaneously functions as a Main track. Aids in testing programming changes and listening to the loco.
<F 1225 1 1> Function Loco1225 Bell On
<F 1225 1 0> Function Loco 1225 Bell Off
<1 PROG> returns the programming track back into Programming mode to be able to Read,Write, Verify and diagnose decoders.

DCC++ Traffic Monitor responses



```
[TX: R 8 1 1]   Prog Read Cmd: CV: 8, Callback Num: 1, Callback Sub: 1
[RX: r1|1|8 141]   Program Reply: Callback Num: 1, Callback Sub: 1, CV: 8, Value: 141
[TX: 1 JOIN]   Track Power ON Cmd
[RX: p1 JOIN]   Power Status: Name:JOINStatus:ON
[TX: F 1225 1 1]   Function Cmd: CAB: 1225, FUNC: 1, State: 1, (No Reply Expected)
[TX: F 1225 1 0]   Function Cmd: CAB: 1225, FUNC: 1, State: 0, (No Reply Expected)
```

☑ Show raw data ☐ Show timestamps ☐ Window always on top ☑ Auto scroll
Filter Bytes:
Background function refresh:
0 function groups are being refreshed

click on 'Show raw data' for more detail

If an Engine decoder won't respond or errors while on the programming track, use these commands to correct it so it does respond.

Advance Settings and Command Codes for Diagnostics and enhanced Decoder Pro Programming

Adjusting the Acknowledgment ACK response time and ranges on the Command Station to improve reading decoders;
The NMRA standard AckPulseDuration is 6000us +- 1000us, so between 5000us <=pulse=> 7000us for a ACK response or 5ms & 7ms

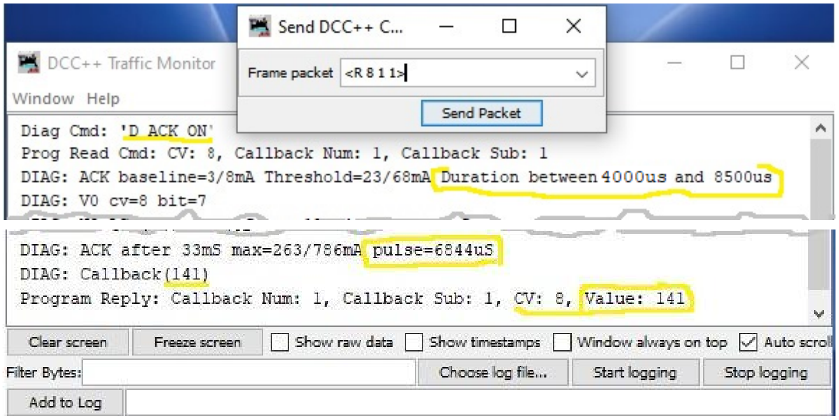
DCC++EX acknowledge ACK default is more forgiving and is set a little wider for a decoder to acknowledge ACK between a AckPulseDuration of 4000us <=pulse=> 8500us, or from 4ms to 8.5ms for a decoder to ACK acknowledge our request.

You can temporarily modify these settings with the DCC++ Send Command to test new ranges to listen to the decoder.
Example: sending <D ACK MAX 13000> would increase the top end from 8500 to 13000 or raise it to 13ms.
We've found higher settings improved reading of Broadway Limited QSI decoders which ACK respond pulse = 12500 or longer.
and MTH Electronics decoders with a ACK response pulse = 19000 or longer.

Use these commands if a decoder doesn't respond to Decoder Pro when pressing '+New Loco' or 'Identify' loco buttons.
At some point you will come across either a complex or stubborn or defective or older decoder like QSI and others.
The following are new DCC++EX features and commands that can improve your ability to read, write and compare CV's.

First turn the Command Station On and the new ACK diagnostic feature, and be sure to open the serial DCC++ Traffic Monitor.
Enter and send <D ACK ON> Diagnostic Acknowledge On which sends diagnostic responses back to the Traffic Monitor.
Then send <R 8 1 1> command to request Read CV8, Mfr. ID and watch the decoder respond with a duration MAX pulse= xxxx.
The following Traffic DCC++ Monitors show examples of two different Locomotive responses;

A Good decoders response from Loco #1225 with a pulse=6844us is within NMRA standard pulse of 5000us <=pulse=> 7000us range, and the locomotive responds with a valid Mfr. ID of 141 – Soundtraxx Throttle-Up decoder.

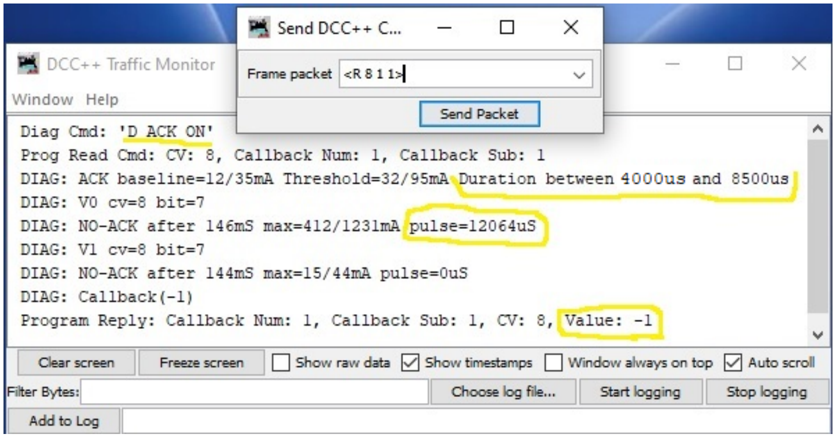


Frame packet <R 8 1 1>

```
Diag Cmd: 'D ACK ON'
Prog Read Cmd: CV: 8, Callback Num: 1, Callback Sub: 1
DIAG: ACK baseline=3/8mA Threshold=23/68mA Duration between 4000us and 8500us
DIAG: V0 cv=8 bit=7
DIAG: ACK after 33mS max=263/786mA pulse=6844uS
DIAG: Callback(141)
Program Reply: Callback Num: 1, Callback Sub: 1, CV: 8, Value: 141
```

unclick 'Show raw data' to cleanup the screen response.

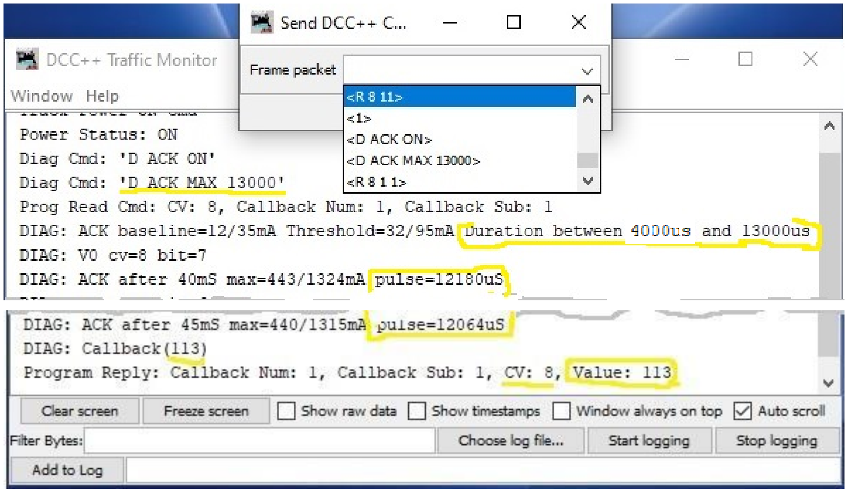
An out of standards QSI decoder response from Loco #1204 with a high ACK MAX pulse=12064us range.
Far outside the acceptable and forgiving DCC++EX range of minimum 4000us <=pulse=> 8500us maximum, and the locomotive responds with an Invalid Mfr. ID of -1. No habla inglés



Frame packet <R 8 1 1>

```
Diag Cmd: 'D ACK ON'
Prog Read Cmd: CV: 8, Callback Num: 1, Callback Sub: 1
DIAG: ACK baseline=12/35mA Threshold=32/95mA Duration between 4000us and 8500us
DIAG: V0 cv=8 bit=7
DIAG: NO-ACK after 146mS max=412/1231mA pulse=12064uS
DIAG: V1 cv=8 bit=7
DIAG: NO-ACK after 144mS max=15/44mA pulse=0uS
DIAG: Callback(-1)
Program Reply: Callback Num: 1, Callback Sub: 1, CV: 8, Value: -1
```

We'll now use the new <D ACK MAX> command to wait for a longer length of time for an ACK response.
Send command <D ACK MAX 13000> set to wait longer for 13000 or 13ms for an ACK response.
Then again resend <R 8 1 1> Read CV8 and the locomotive responds with a valid Mfr. ID of 113 – QSI Industries



```
Power Status: ON
Diag Cmd: 'D ACK ON'
Diag Cmd: 'D ACK MAX 13000'
Prog Read Cmd: CV: 8, Callback Num: 1, Callback Sub: 1
DIAG: ACK baseline=12/35mA Threshold=32/95mA Duration between 4000us and 13000us
DIAG: V0 cv=8 bit=7
DIAG: ACK after 40mS max=443/1324mA pulse=12180uS
DIAG: ACK after 45mS max=440/1315mA pulse=12064uS
DIAG: Callback(113)
Program Reply: Callback Num: 1, Callback Sub: 1, CV: 8, Value: 113
```

Press the arrow & scroll, choose a command to rerun

See Manufacturer's ID DCC_Shortcuts Card.pdf

Now when you Press '+New Loco' and or 'Identify Loco' button on Decoder Pro it should find and highlight the Engine ID in your Roster list. Ready now for you to either Program & change CV's, or Run the Engine on the main line.

Current ACK LIMIT Command
Additional diagnostic tuning can be done with a current ACK LIMIT command.
The NMRA current AckLimitmA standard threshold is between =7/20mA and =27/80mA.
DCC++EX current AckLimitmA baseline standard is set to = 60mA.
This can be modified in a similar way as before by using the <D ACK LIMIT xx > command to help increase the reading of difficult decoders. Adjust this xx range by setting the limit in between '20' for more sensitive, and to '100' for less sensitive.
We found older Digitrax decoders ACKlimitsmA in the 35 range so we set <D ACK LIMIT 30> and it Responded to 'Identify loco'.

These are all temporary D ACK xx user adjustments numbers and the Command Station will reset them to the DCC++EX standard defaults when the Arduino is powered off & on again. But we have a way to save the ranges you've tested and deemed as best to use.

Keeping and saving the new diagnostic range numbers as your new default standard at startup.

Make those new <D ACK MIN & MAX > and <D ACK LIMIT > range input numbers more permanent by first creating and editing the new mySetup.h file in the CommandStation-EX.ino firmware sketch, by using the Arduino Interactive Development Editor IDE 1.8. The mySetup.h file is Not part of JMRI software. It's a simple text file you create once and place it in the CommandStation-EX folder.

Quick Overview of a new mySetup.h file located in the DCC++CommandStation-EX.ino firmware sketch

New to DCC++EX 3.1, this feature provides a method for sending commands and changing settings and preferences that will be run at every startup. Think of the mySetup.h file as an "autoexec" file or "macro" to make changes so that you don't have to manually enter commands every time you start the Command Station for things you want to customize. mySetup.h is a simple text file, you don't have to be a programmer or get into the command station code to use it. Created with a simple text editor or development editor.

* try different settings and find what works best then make them permanent
* have the Command Station start with power on
* change CV MIN and MAX read settings (ACK Detection) when programming CV's, then make them permanent
* send many other <DCC++> commands and make them part of your startup procedures.

Inside the mySetup.h file you enter SETUP( ) commands and can also comment them out without having to delete them by putting "//" in front of the SETUP line so that it is bypassed and not implemented. You can copy and paste these samples if you like.

Examples for using mySetup.h user file: Note; Make sure you capitalize the 'S' in mySetup.h when you first create the file.

```
mySetup.h
 // DCC++EX Commands and Command Station Diagnostics settings to begin at each boot or startup.
 // SETUP("<          >");             // open for user defined command,  this setup will not be executed at startup
 // SETUP("<1>");                      // Commented out the turn power on to both tracks, this setup will not be executed at startup
 // SETUP("<1 MAIN>");                 // Start with main track power on.  By entering just a "1" it would turn both tracks on
    SETUP("<1 PROG>");                 // Start with Programming track power on.
 // SETUP("<1 JOIN>");                 // Isolated PROG track Joined to Main for Testing & Drive Away.  Sending a <1 PROG> or <0 JOIN> will turn it off
 // Diagnostic Adjustments.
 // SETUP("<D ACK ON>");               // turn on Prog track ACK diagnostics.  Not recommended unless you only use this CS as a Programming Station
 // SETUP("<D ACK MIN 2000>");  // to Lower the Minimum Acknowledgement range to xxxx sec or 2ms
   // DCC++EX standard AckPulseDuration or ACKnowledgement settings are currently set from MIN 4000 to MAX 8500, or 4mS to 8.5mS.
    SETUP("<D ACK MAX 13000>"); // Raise the Upper Acknowledgement range to allow pulses up to 13 microseconds to read old QSI decoders
 // SETUP("<D ACK LIMIT 30>");     // Override Std default of ackLimitmA = 60  by tuning it between 20 to 100 to try and improve CV reads
 // Display Information on LCD or OLED screens.
 // LCD(7, F("\n mySetup has been run\n"));  // message sent to the LCD display
 // Special Hardware settings.
 // Serial2.begin(115200);             // set Serial2 to 115200 baud to communicate with xxx device like a Bluetooth HC-06 Module
                                      // on Serial2 so it doesn't conflict with a USB cable connection on Serial0 when updating Firmware.
// End of myStup.h
```

All Command Station commands are valid but some make little sense to use during startup.

The mySetup.h file is optional, if you don't have one, the Command Station will just start as normal using standard defaults
But if the Command Station sees a mySetup.h file, it will process the commands in the order given.

mySetup.h is specific to the user, it is not part of the standard install and is not in any zip file. If you want one, you simply create it on your PC with a text editor and place it in the same CommandStation-EX folder with all the other DCC++ files, periodically edit it, and then upload the 'CommandStation-EX.ino firmware' back into your Arduino Uno or Mega Command Station.

Any future DCC++EX releases will not overwrite mySetup.h or replace it when you download a new version of the CommandStation-EX firmware. It stays as is and unique to your layout until you specifically choose to make future changes to it through the Arduino IDE or other editors. You simply copy and paste the mySetup.h file from your previous release folder to the new release folder and upload the firmware to the Arduino DCC++ EX Command Station as normal.

Please visit our website for further instructions of how to upload CommandStation-EX.ino firmware and a mySetup.h file using the.Arduino Interactive Development Environment IDE editor here; Installing using Arduino IDE
and the new startup configuration mySetup.h file advanced-setup/startup-config mySetup.h

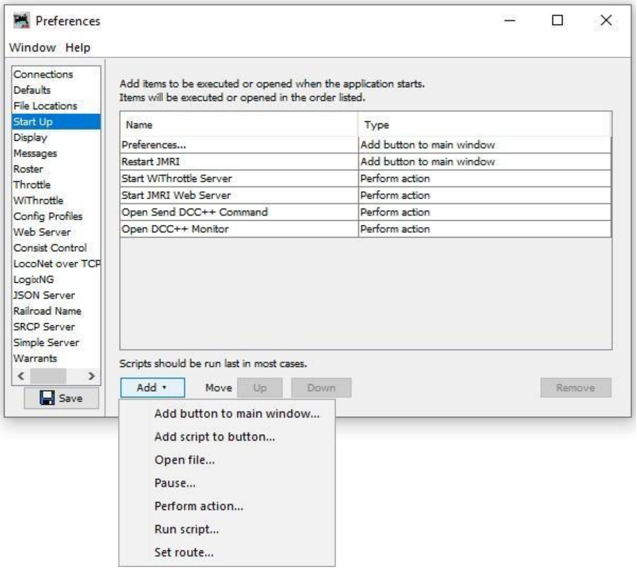
DecoderPro Housekeeping and Other Set Up panels - Options and Settings for Decoder Pro 'Preferences'

From DecoderPro All Roster entries screen click, 'Edit' > 'Preferences'
Preferences is the single largest pane in Decoder Pro where you set up your customization of functionality.
Your DCC++ Decoder Pro Programming Station has choices in the Preferences panel where there are many items we need to set or change, from Connections, Start Up, Roster Owner, Railroad Name, Config Profile, File Locations, WiThrottle, Defaults, and Alternate JMRI Connections, etc. Here are a few of the more important panel setting suggestions and on how to initially set them up.

Preferences > Start Up
This Start Up pane is initially empty and JMRI displays a simple Decoder Pro Roster panel screen, as seen on page 5.
You can use this Start Up pane to Start other panes, like Send DCC++ Command, Traffic Monitor, WiThrottle Server, etc.
This Start Up pane could take a user's guide all by itself.

We recommend at a minimum starting the following by clicking on 'Add' then choosing from a drop down list of items;
'Preferences' displays a GUI button on the main Roster panel to open this Preference pane.
'Restart JMRI' displays a GUI button on the main Roster panel to restart JMRI.
'Start WiThrottle Server' for WiFi throttles to be able to connect through WiFi to JMRI and your layout.
'Start JMRI Web Server' to allow media images of your engine to be sent to WiThrottles like the Engine Driver App.
'Open Send DCC++ Command' pane to be able to send DCC++ Commands.
'Open DCC++ Monitor' pane to see the serial send and receive data from DCC++ and JMRI.

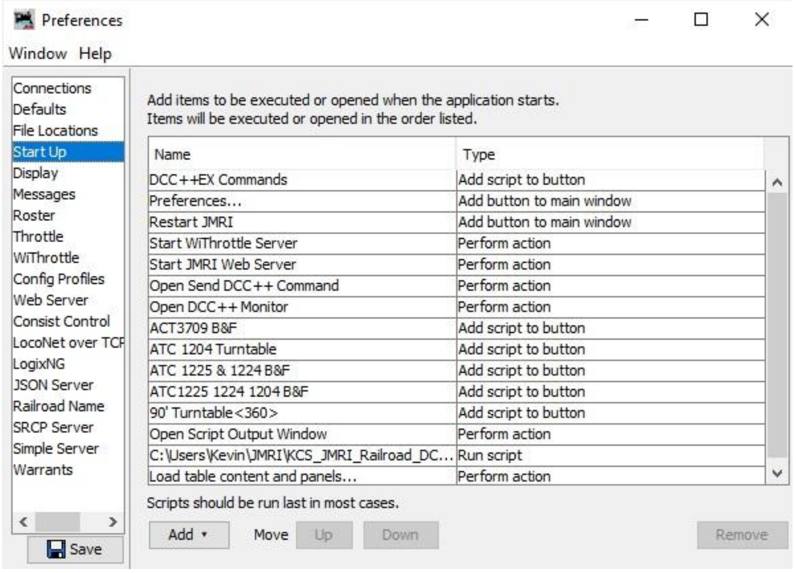
These are a few more advanced Start Up features shown below on the right. More details can be found in the 'Help' > 'Window Help'.



Add items to be executed or opened when the application starts.
Items will be executed or opened in the order listed.

Name	Type
Preferences...	Add button to main window
Restart JMRI	Add button to main window
Start WiThrottle Server	Perform action
Start JMRI Web Server	Perform action
Open Send DCC++ Command	Perform action
Open DCC++ Monitor	Perform action

Scripts should be run last in most cases.



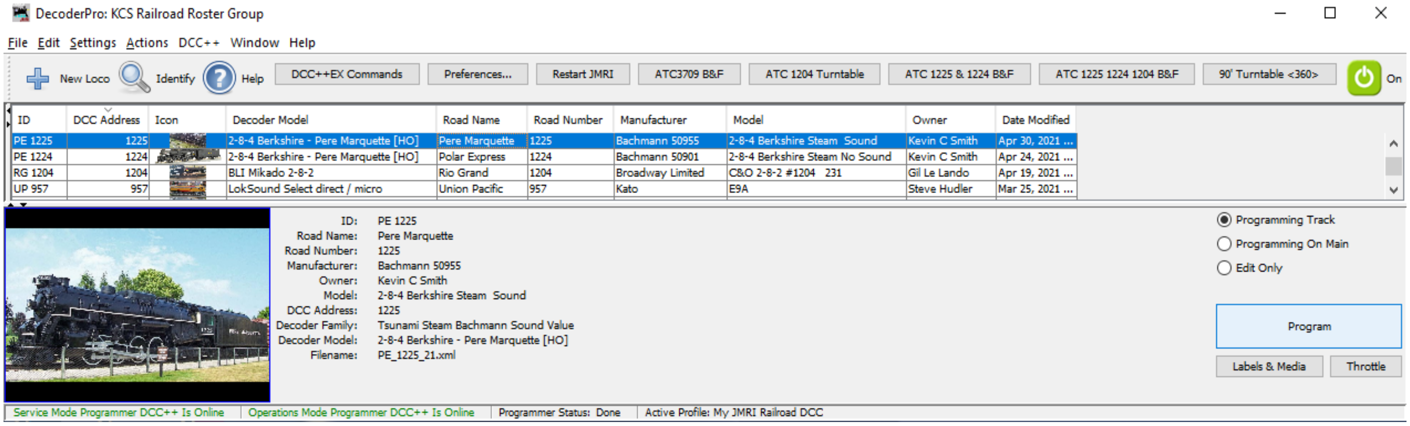
Add items to be executed or opened when the application starts.
Items will be executed or opened in the order listed.

Name	Type
DCC++EX Commands	Add script to button
Preferences...	Add button to main window
Restart JMRI	Add button to main window
Start WiThrottle Server	Perform action
Start JMRI Web Server	Perform action
Open Send DCC++ Command	Perform action
Open DCC++ Monitor	Perform action
ACT3709 B&F	Add script to button
ATC 1204 Turntable	Add script to button
ATC 1225 & 1224 B&F	Add script to button
ATC1225 1224 1204 B&F	Add script to button
90' Turntable<360>	Add script to button
Open Script Output Window	Perform action
C:\Users\Kevin\JMRI\KCS_JMRI_Railroad_DC...	Run script
Load table content and panels...	Perform action

Scripts should be run last in most cases.

After 'Saving' then Restarting JMRI, the Start Up panel above right, creates a DecoderPro Roster Comprehensive panel seen below, with Commands, Preferences, Restart, plus predefined GUI buttons to run customized Automatic Throttle Control ATC Script.py files.

Sample GUI Jython Script.py button run ATC Automated Throttle Control for running one or more Engines around the layout



ID	DCC Address	Icon	Decoder Model	Road Name	Road Number	Manufacturer	Model	Owner	Date Modified
PE 1225	1225		2-8-4 Berkshire - Pere Marquette [HO]	Pere Marquette	1225	Bachmann 50955	2-8-4 Berkshire Steam Sound	Kevin C Smith	Apr 30, 2021 ...
PE 1224	1224		2-8-4 Berkshire - Pere Marquette [HO]	Polar Express	1224	Bachmann 50901	2-8-4 Berkshire Steam No Sound	Kevin C Smith	Apr 24, 2021 ...
RG 1204	1204		BLI Mikado 2-8-2	Rio Grand	1204	Broadway Limited	C&O 2-8-2 #1204 231	Gil Le Lando	Apr 19, 2021 ...
UP 957	957		LokSound Select direct / micro	Union Pacific	957	Kato	E9A	Steve Hudler	Mar 25, 2021 ...

ID: PE 1225
Road Name: Pere Marquette
Road Number: 1225
Manufacturer: Bachmann 50955
Owner: Kevin C Smith
Model: 2-8-4 Berkshire Steam Sound
DCC Address: 1225
Decoder Family: Tsunami Steam Bachmann Sound Value
Decoder Model: 2-8-4 Berkshire - Pere Marquette [HO]
Filename: PE_1225_21.xml

● Programming Track
○ Programming On Main
○ Edit Only

Service Mode Programmer DCC++ Is Online | Operations Mode Programmer DCC++ Is Online | Programmer Status: Done | Active Profile: My JMRI Railroad DCC

We've included this to let you know that there are many more features worth exploring in 'Start Up' pane to customize Decoder Pro.
Learn more about this by clicking the 'Help' > 'Window help' tab on each individual JMRI pane.

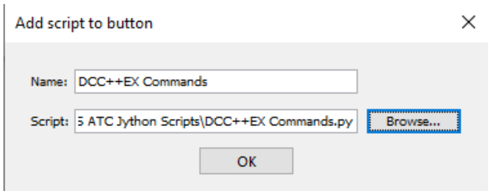
Advanced Features of DecoderPro Roster and Preferences > Start Up

Take a break and grab some Java, Some Tips and Tricks for Tinkerer Level users.

Customize DecoderPro - User Defined Buttons and Scripts

You can add additional custom DecoderPro GUI buttons as seen above that point to and run predefined jyton.py scripts that you have built or previously downloaded from the website. You place them in your C:/ User/JMRI/My_JMRI-Railroad.jmri/subfolder/
This example uses DecoderPro 'Add script to button' feature to create a button and display text in the 'Script Output' panel when the button is pressed a jython script creates a print summary of DCC++EX commands. *Jython is* simply the combined names of *Java & Python*.

DecoderPro main menu Edit> Preferences > Start Up > Add > 'Add script to button'



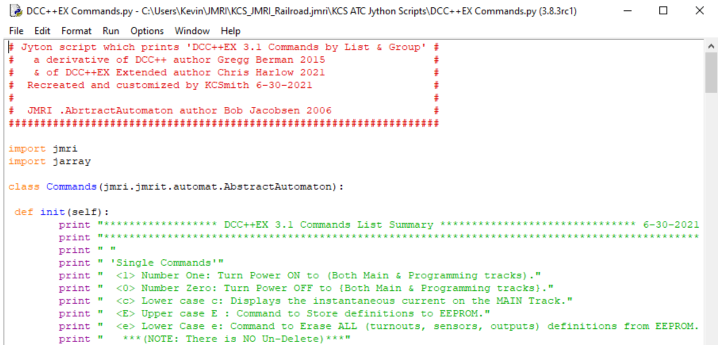
Name: DCC++EX Commands

Script: 5 ATC Jython Scripts\DCC++EX Commands.py

Enter a button Name: 'DCC++EX Commands'

Download the free DCCEX_Commands_3.1.zip file below, unzip it and place it in your C:/Users/name/My_JMRI_Railroad folder

Browse your PC and point to the folder with the Script in it: C:\Users\Kevin\JMRI\KCS_JMRI_Railroad.jmri\KCS ATC Jython Scripts\DCC++EX Commands3.1.py   And press OK.


DCC++EX Commands.py - C:\Users\Kevin\JMRI\KCS_JMRI_Railroad.jmri\KCS ATC Jython Scripts\DCC++EX Commands.py (3.8.3rc1)
File Edit Format Run Options Window Help

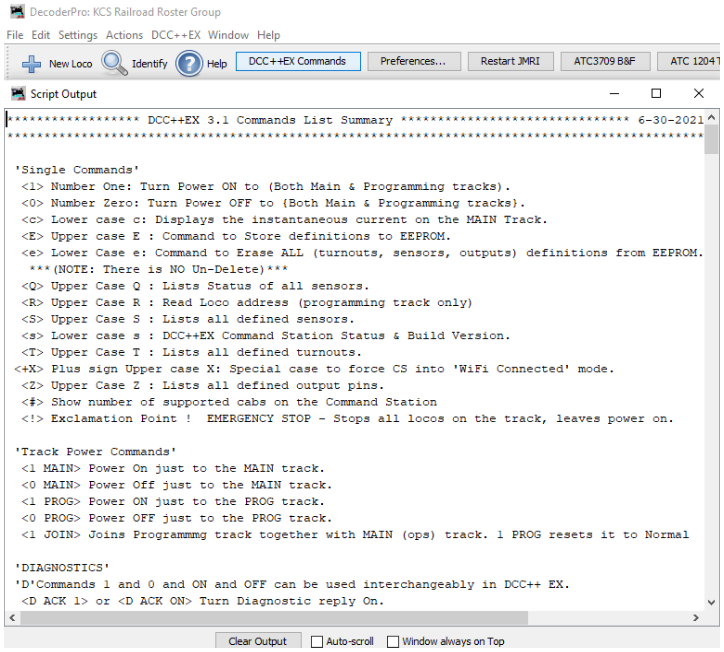
```
# Jyton script which prints 'DCC++EX 3.1 Commands by List & Group' #
#   a derivative of DCC++ author Gregg Berman 2015                  #
#   & of DCC++EX Extended author Chris Harlow 2021                  #
#  Recreated and customized by KCSmith 6-30-2021                    #
#                                                                   #
#  JMRI .AbrtractAutomaton author Bob Jacobsen 2006                 #
####################################################################

import jmri
import jarray

class Commands(jmri.jmrit.automat.AbstractAutomaton):

 def init(self):
        print "****************** DCC++EX 3.1 Commands List Summary ****************************** 6-30-2021
        print "*******************************************************************************************
        print " "
        print " 'Single Commands'"
        print "  <1> Number One: Turn Power ON to (Both Main & Programming tracks)."
        print "  <0> Number Zero: Turn Power OFF to {Both Main & Programming tracks}."
        print "  <c> Lower case c: Displays the instantaneous current on the MAIN Track."
        print "  <E> Upper case E : Command to Store definitions to EEPROM."
        print "  <e> Lower Case e: Command to Erase ALL (turnouts, sensors, outputs) definitions from EEPROM.
        print "   ***(NOTE: There is NO Un-Delete)***"
```

[DCC++EX Commands 3.1](#) download this Command script

Then at startup we'll open the 'Script Output' pane first and then print the new DCC++EX Commands.py file by setting;
  Edit> Preferences > Start Up> Add> Perform action> 'Open Script Output Window',
  Edit> Preferences > Start Up> Add> Run Script> C:\User folder\ DCC++EX Commands3.1.py

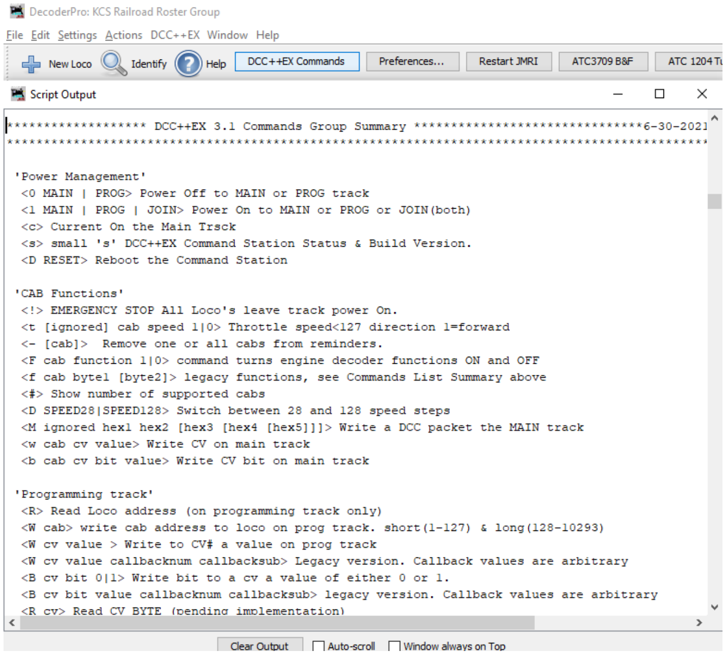
When JMRI starts it will automatically Open the Script Output Window and Start DCC++EX Commands3.1.py script.
You can also press the [DCC++EX Commands] button & produce the scrollable DCC++EX commands in the Script Output window.

Example DCC++EX Commands by List



```
****************** DCC++EX 3.1 Commands List Summary ****************************** 6-30-2021
*******************************************************************************************

 'Single Commands'
  <1> Number One: Turn Power ON to (Both Main & Programming tracks).
  <0> Number Zero: Turn Power OFF to {Both Main & Programming tracks}.
  <c> Lower case c: Displays the instantaneous current on the MAIN Track.
  <E> Upper case E : Command to Store definitions to EEPROM.
  <e> Lower Case e: Command to Erase ALL (turnouts, sensors, outputs) definitions from EEPROM.
   ***(NOTE: There is NO Un-Delete)***
  <Q> Upper Case Q : Lists Status of all sensors.
  <R> Upper Case R : Read Loco address (programming track only)
  <S> Upper Case S : Lists all defined sensors.
  <s> Lower case s : DCC++EX Command Station Status & Build Version.
  <T> Upper Case T : Lists all defined turnouts.
 <+X> Plus sign Upper case X: Special case to force CS into 'WiFi Connected' mode.
  <Z> Upper Case Z : Lists all defined output pins.
  <#> Show number of supported cabs on the Command Station
  <!> Exclamation Point !  EMERGENCY STOP - Stops all locos on the track, leaves power on.

 'Track Power Commands'
  <1 MAIN> Power On just to the MAIN track.
  <0 MAIN> Power Off just to the MAIN track.
  <1 PROG> Power ON just to the PROG track.
  <0 PROG> Power OFF just to the PROG track.
  <1 JOIN> Joins Programmmg track together with MAIN (ops) track. 1 PROG resets it to Normal

 'DIAGNOSTICS'
 'D'Commands 1 and 0 and ON and OFF can be used interchangeably in DCC++ EX.
  <D ACK 1> or <D ACK ON> Turn Diagnostic reply On.
```

Scroll down for DCC++EX Commands by Group



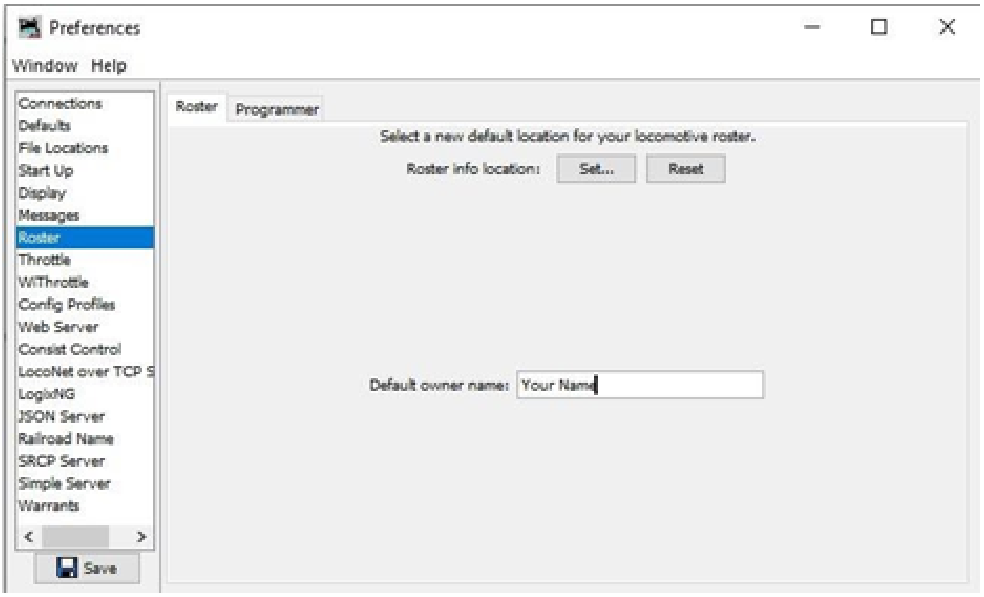
```
****************** DCC++EX 3.1 Commands Group Summary *******************************6-30-2021
*******************************************************************************************

 'Power Management'
  <0 MAIN | PROG> Power Off to MAIN or PROG track
  <1 MAIN | PROG | JOIN> Power On to MAIN or PROG or JOIN(both)
  <c> Current On the Main Trsck
  <s> small 's' DCC++EX Command Station Status & Build Version.
  <D RESET> Reboot the Command Station

 'CAB Functions'
  <!> EMERGENCY STOP All Loco's leave track power On.
  <t [ignored] cab speed 1|0> Throttle speed<127 direction 1=forward
  <- [cab]>  Remove one or all cabs from reminders.
  <F cab function 1|0> command turns engine decoder functions ON and OFF
  <f cab byte1 [byte2]> legacy functions, see Commands List Summary above
  <#> Show number of supported cabs
  <D SPEED28|SPEED128> Switch between 28 and 128 speed steps
  <M ignored hex1 hex2 [hex3 [hex4 [hex5]]]> Write a DCC packet the MAIN track
  <w cab cv value> Write CV on main track
  <b cab cv bit value> Write CV bit on main track

 'Programming track'
  <R> Read Loco address (on programming track only)
  <W cab> write cab address to loco on prog track. short(1-127) & long(128-10293)
  <W cv value > Write to CV# a value on prog track
  <W cv value callbacknum callbacksub> Legacy version. Callback values are arbitrary
  <B cv bit 0|1> Write bit to a cv a value of either 0 or 1.
  <B cv bit value callbacknum callbacksub> legacy version. Callback values are arbitrary
  <R cv> Read CV BYTE (pending implementation)
```

All other predefined User built and Automated Throttle Control scripts are added to the DecoderPro panel the same way.

Preferences > Roster

Two tabs, 'Roster' and then 'Programmer'

Earlier during the initial 'Roster' install you entered Your Name in the Default owner name field and all new Locomotives entered now default to your name as the owner in the DecoderPro Roster panel.



Select a new default location for your locomotive roster.

Roster info location:

Default owner name: Your Name

Preferences > Roster >Programmer

Now initially set the Preferences > 'Programmer' pane as shown

Format: >Advanced is the most robust, so set format to: Advanced. Later try and test them all to see what works best for you.

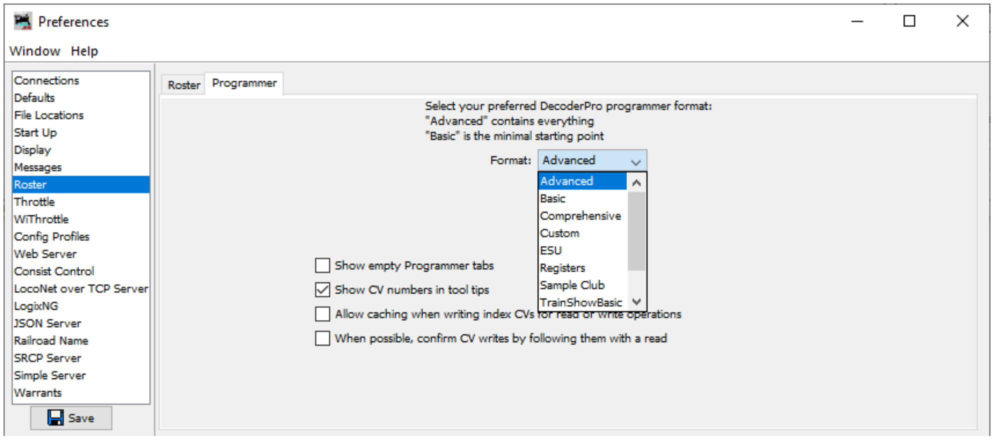
This opens all the available CV tabs on the Comprehensive Programmer Roster Panel so all become visible.

Check mark to turn on the Show CV numbers in tool tips feature which shows you the fields CV# when you hover over it.

The amount of seconds the tool tips display time lasts can be changed in Preferences > Display pane.

,

Roster Programmer



Select your preferred DecoderPro programmer format:
"Advanced" contains everything
"Basic" is the minimal starting point

Format: Advanced

□ Show empty Programmer tabs

☑ Show CV numbers in tool tips

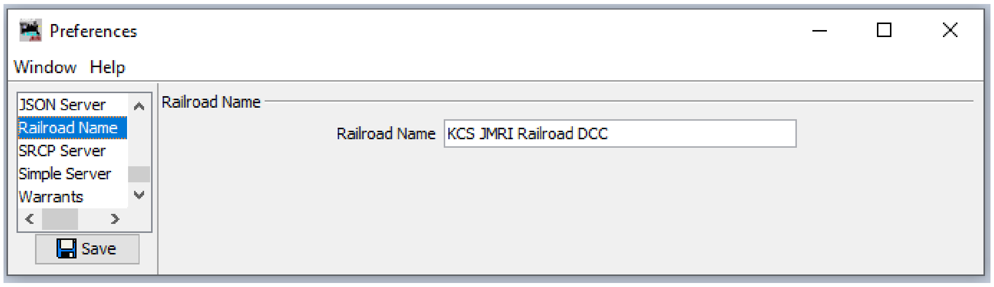
□ Allow caching when writing index CVs for read or write operations

□ When possible, confirm CV writes by following them with a read

You can come back to this Programmer pane any time and try different settings as desired.

Preferences > Railroad Name

Change 'My JMRI Railroad' to a more Identifiable name like your Initials or your real model railroad.



Railroad Name KCS JMRI Railroad DCC

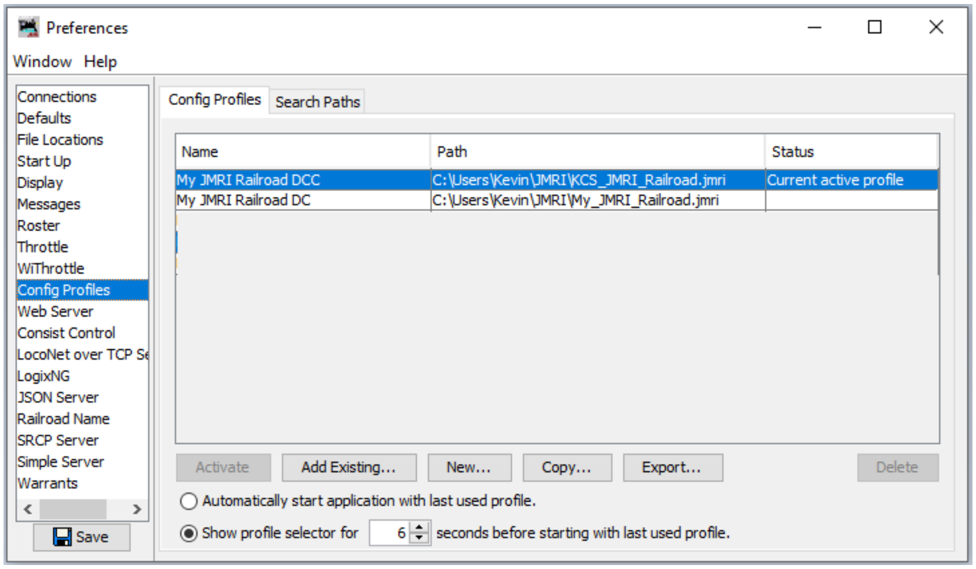
Preferences > Configuration Profiles

Initially we can just default to the current Config Profile name as is, or copy and Rename it if you like.
But in the future, you can set up multiple JMRI sessions or Configuration Profiles through this pane.
These Profile file folder names will be inside your main C:/Users/Name//JMRI/folder.

Example; Copy My JMRI Railroad and rename the profile 'My JMRI Railroad DCC' to use a DCC++EX Command Station.
Then setup a New second command station config profile by Copying My JMRI Railroad and rename that profile 'My JMRI Railroad DC' for Legacy Analog DC layouts using DC+ CAB 2.1 Base Station software developed by Geoff Bunza running on a second Arduino and motor shield.   You can also use it to change the word 'My' to Your initial's 'ABC JMRI Railroad' if you like.

Note: You can run both profiles simultaneously with each Command Station CS profile or session having a separate and different Preferences > Connections > Serial Port Com# & Connection Prefix set up defined as well, as previously seen on the Connections panel on page 4.   Save it, then restart JMRI.

At the startup of JMRI it will pause and give you a choice of which profile you wish to open and use for that session, either My JMRI Railroad DCC or My JMRI Railroad DC. Then it proceeds to start JMRI DecoderPro with all your specific Profile Preferences.
For our purposes we will automatically default to your first and only Config Profile until you decide to Change or add Profiles.



Name	Path	Status
My JMRI Railroad DCC	C:\Users\Kevin\JMRI\KCS_JMRI_Railroad.jmri	Current active profile
My JMRI Railroad DC	C:\Users\Kevin\JMRI\My_JMRI_Railroad.jmri	

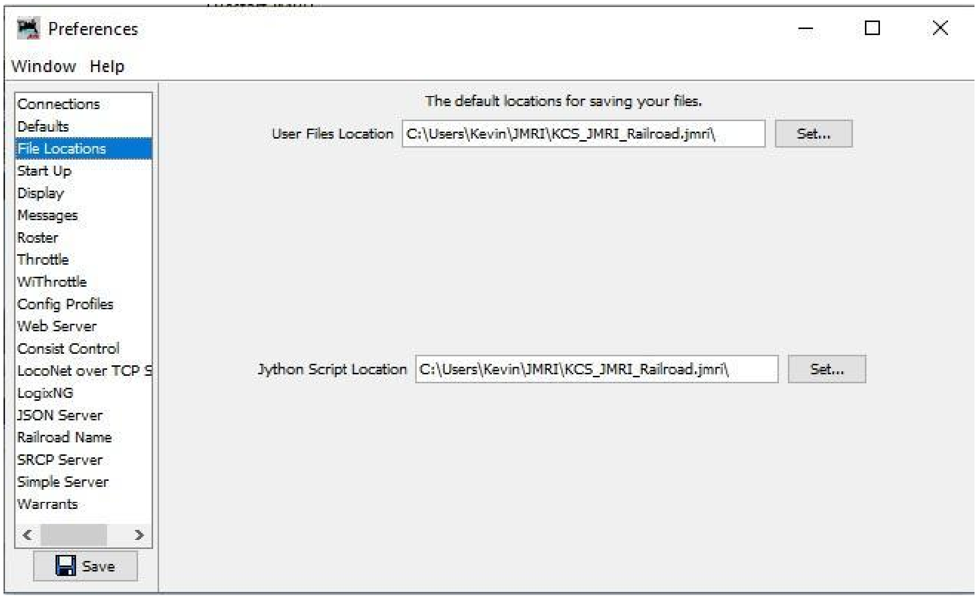
○ Automatically start application with last used profile.

● Show profile selector for 6 seconds before starting with last used profile.

Preferences > File Locations

Initially you may wish to leave these settings as is. Later if you build automation scripts you can change the lower file directory to point at your script folder location which you might rename by using your initial's 'ABC JMRI Railroad scripts' to better Identify the file folder and profile from the standard default JMRI setup.

File Locations



The default locations for saving your files.

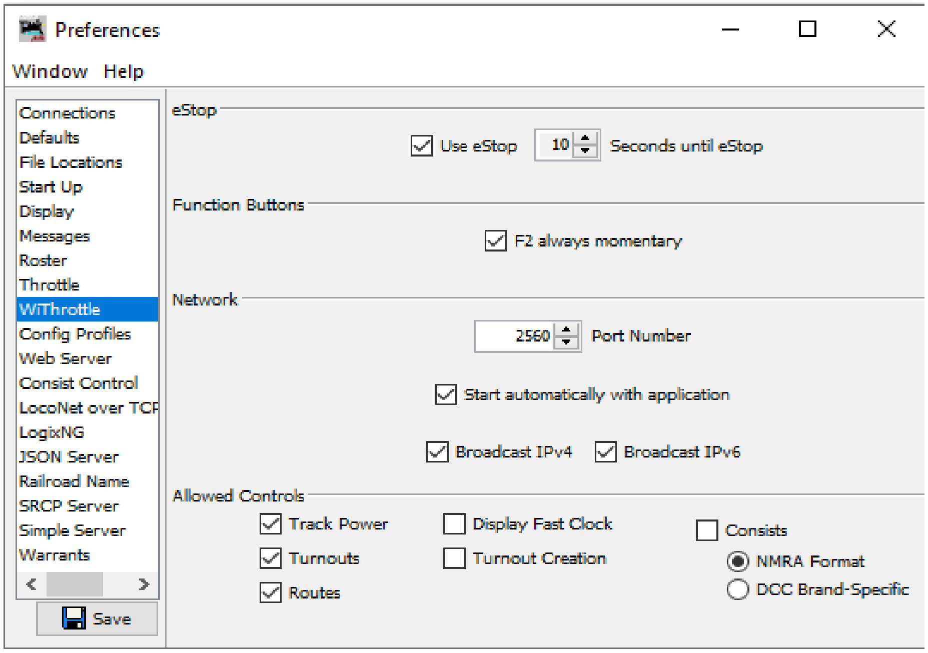
User Files Location C:\Users\Kevin\JMRI\KCS_JMRI_Railroad.jmri\

Jython Script Location C:\Users\Kevin\JMRI\KCS_JMRI_Railroad.jmri\

Preferences > WiThrottle Server

This provides a WiFi connection between JMRI and a handheld smartphone as a wireless WiFi Throttles. DCC-EX WiFi Config
Set the Port Number you wish to be the default that a wireless WiThrottle will enter in order to attach to the JMRI WiThrottle Server.
Example; when a Smartphone WiThrottle App like Engine Driver or WiThrottle, or DCCpp CAB wants to sign onto JMRI as a client, the WiThrottle Server pane will display an IP address and port number like this: 192.168.1.103:2560   IP : Port #
You would then enter this IP:Port# into the smartphone throttle app and it connects to JMRI WiThrottle Serve as a WiFi Throttle.
NOTE: *If you are going to Operate Two CS Profiles simultaneously then make DCC++EX port 25****60*** *and make DC+CAB port 25****80***
This works the same way if you have a DCC++EX DecoderPro Programming Station and a Proprietary DCC system for Operations.

WiThrottle Server



eStop

☑ Use eStop 10 Seconds until eStop

Function Buttons

☑ F2 always momentary

Network

2560 Port Number

☑ Start automatically with application

☑ Broadcast IPv4 ☑ Broadcast IPv6

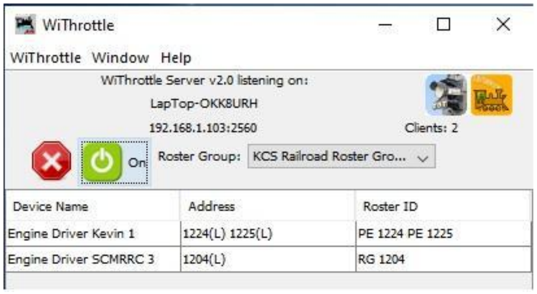
Allowed Controls

☑ Track Power □ Display Fast Clock □ Consists
☑ Turnouts □ Turnout Creation ● NMRA Format
☑ Routes ○ DCC Brand-Specific

If you wish to have Turnout names, Sensor names and Route names to be passed to and used in remote WiFi Throttles that are attached to JMRI WiThrottle Server then check x the boxes and press Save.
Note; Configuring Sensors, Turnouts & Outputs are accessed through the main Decoder Pro Roster panel by clicking on 'DCC++' then 'Configure Base Station' tabs. There you will add & assign pin# of Sensors, Turnouts and Output devices and then File > Save.
For additional information on adding these please see JMRI DCC++ Sensors and /DCC++ Turnouts & Outputs.

WiThrottle Server

After starting WiThrottle Server either through DecoderPro menu 'Actions' > 'Start WiThrottle' or through the 'Preferences' > 'Start Up' > 'WiThrottle' pane, it shows you the IP address sent by your Router and the Port # you set in the 'Preferences' > 'WiThrottle'.



WiThrottle Server v2.0 listening on:
LapTop-OKK8URH
192.168.1.103:2560
Clients: 2

Device Name	Address	Roster ID
Engine Driver Kevin 1	1224(L) 1225(L)	PE 1224 PE 1225
Engine Driver SCMRRC 3	1204(L)	RG 1204

2 Engine Driver Withrottles are signed on and running Eng PE 1224 & 1225 and RG 1204 respectively

By clicking on the Roster Group: arrow bar you can choose either All Entries or a previously built Custom Roster Groups to send to your smartphone's WiThrottle Server Roster.

Roster Groups are created on the main DecoderPro panel menu, 'Settings' > 'Create Roster Group' tab.
Then assign them to a roster group through the main DecoderPro panel menu 'Settings' > 'Roster Group Table Association' tab.
This allows you to shrink down the list of engines viewable on both the Main Roster panel and in the smartphone Rosters on the Engine Driver and WiThrottle and DCCpp_CAB Throttle smartphone Apps.

Note: Connecting to and receiving JMRI Engine Roster files on a WiFi Throttle is only available by first Starting Withrottle Server and then connecting to the same WiFi SSID "name" & "password" for both the PC and smartphone devices.
Alternatively, when connecting a WiFi Throttle Directly to a Standalone DCC++EX WiFi enabled Command Stations internal IP address with a Smartphone, it will only show the local Apps engine roster and default function keys to display in the WiFi Throttle. Not the engines in the DecoderPro Roster. Any questions should be referred to your specific WiTHrottle App provider.

WiThrottle Applications compatible with DCC++ EX 3.1

Use Android or Apple iOs, or Windows Smartphones as WiFi Throttles Apps to operate Engines, throw switches and run routes & accessories. To use Wireless Smartphone Throttles, you must download & install one of these Apps from an online play store.

WiThrottle Smartphone software is Free

Android Engine Driver 2.29 or higher for WiFi
iPhone WiThrottle Lite 3.0 or higher for WiFi
Android DCCpp CAB 2.7 or higher runs both WiFi and serial Bluetooth

Sign on a local WiFi Router or access point

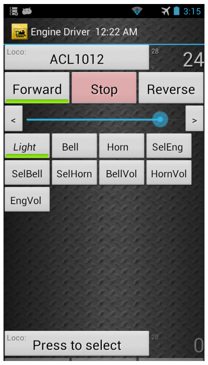
Look for and Sign into the same SSID WiFi signal connection and password for both your PC and Smartphone to share in common.

Then from the Decoder Pro Roster screen, click 'Action', then 'Start WiThrottle Server' to start and open the panel.
Note the IP address and Port number, look for this 192.168.x.xxx : xxxx. displayed in the WiThrottle Server panel.
Enter this number on your Smartphone throttle application.

After downloading the App to the smartphone, open it up and enter the IP number and Port number that was displayed in the JMRI Decoder Pro – 'WiThrottle Server' panel, example IP 192.168.1.103 port 2560 then press the Connect button.
Pick an engine from your JMRI Roster displayed on your phone, and have fun!

JMRI WiThrottle Server Help [WiThrottle Connected Throttles]() & Choosing a Controller (Throttle) [Throttle Controllers]()

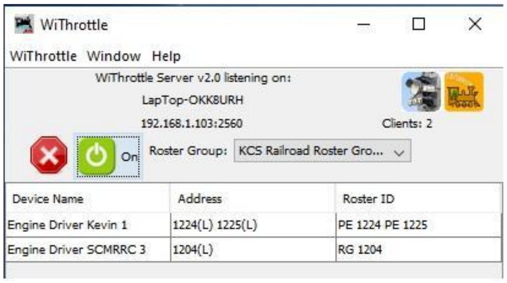
Engine Driver
Android



WiThrottle
iPhone

DCCpp CAB
Android

WiThrottle Server Panel



WiThrottle Server v2.0 listening on:
LapTop-OKK8URH
192.168.1.103:2560
Clients: 2
Roster Group: KCS Railroad Roster Gro...

Device Name	Address	Roster ID
Engine Driver Kevin 1	1224(L) 1225(L)	PE 1224 PE 1225
Engine Driver SCMRRC 3	1204(L)	RG 1204

WiThrottle Server Panel from Decoder Pro and Panel Pro on your PC

Here is a list of some of the WiThrottle Apps currently available and or currently in testing with DCC++EX Command Station.
See DCC++EX WiThrottle supported and help pages for the latest on these and other Throttles;

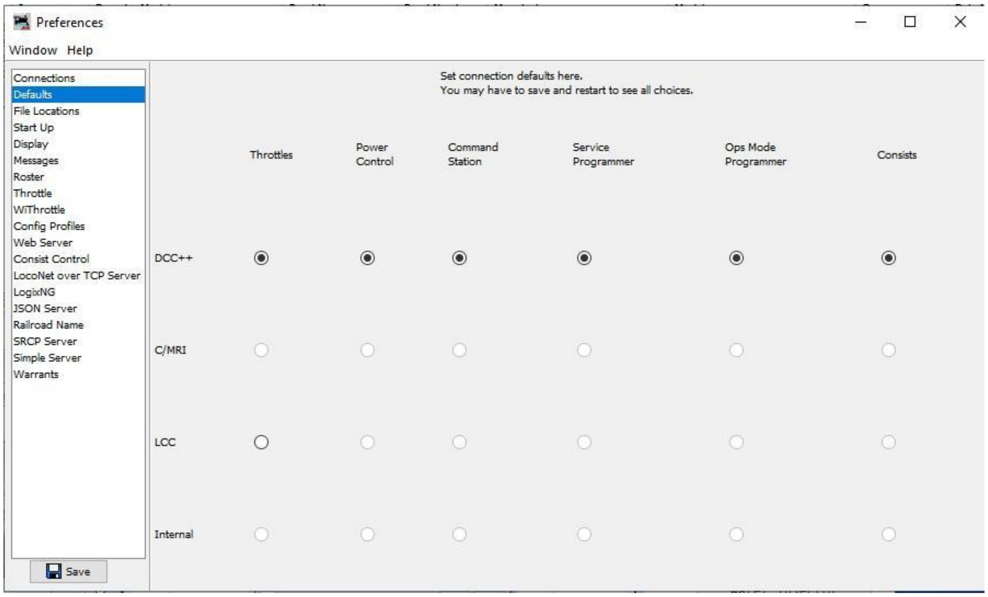
- [WiThrottle Server, Web Server, DCC++ API Explained]()
- [DCC++ EX Web Throttle]()
- [Engine Driver]()
- [WiThrottle]()
- [DCCpp CAB]()
- [Locontrol]()
- [Cab Engineer: DCC Throttle]()
- [DigiTrainsPro]()
- [SRCP Client for iOS]()

{Tips: inside a smartphone Engine Driver app you may want to setup Preferences – Throttle screen layout - Switching (*) throttle function}

Preferences >Defaults

If in the future when you set up additional connections be sure to come back to this screen and set the connection Defaults as shown to the DCC system you're connected to live. In this case they're all set to DCC++. Defaults can be set between more than one system, such as a DCC++ Service Programer Mode Station only, and a Proprietary DCC Command Station for Operations Mode only.
Be sure to verify what your settings are periodically, otherwise you'll be playing ping pong between systems.

Defaults



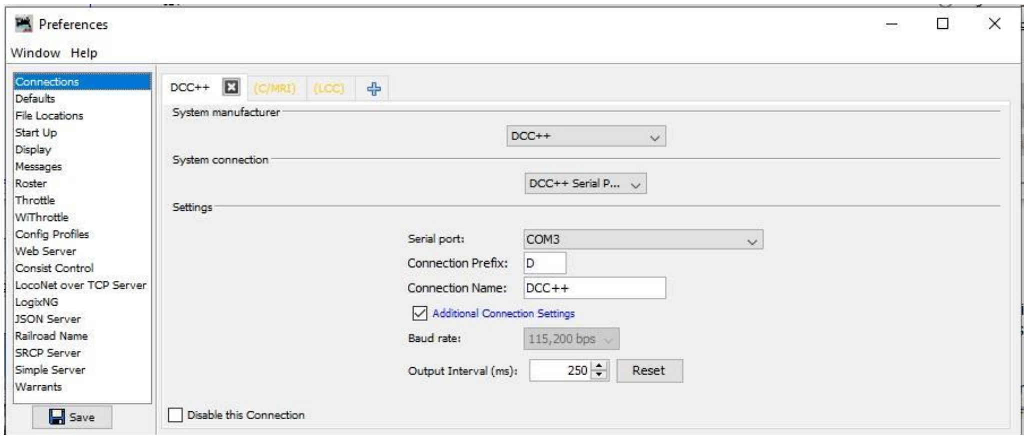
Set connection defaults here.
You may have to save and restart to see all choices.

	Throttles	Power Control	Command Station	Service Programmer	Ops Mode Programmer	Consists
DCC++	●	●	●	●	●	●
C/MRI	○	○	○	○	○	○
LCC	○	○	○	○	○	○
Internal	○	○	○	○	○	○

Alternate JMRI Connections;

Preferences > Connections

Sample JMRI serial comm setup for DCC++, CMRI and LCC Systems



System manufacturer: DCC++

System connection: DCC++ Serial P...

Settings

Serial port: COM3

Connection Prefix: D

Connection Name: DCC++

☑ Additional Connection Settings

Baud rate: 115,200 bps

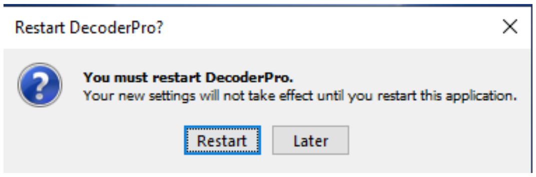
Output Interval (ms): 250

□ Disable this Connection

These other connections would be for your Secondary Accessory bus to communicate to the layouts Sensors, Turnouts, Signals and assorted ancillary accessories you may place on the layout in the future. This will be discussed more in depth in a future DCC++EX & JMRI Layout Command Control LCC Operations Guide. Or please visit JMRI.org and look at the Arduino setup for a CMRI Chubb Model Railroad Interface and other hardware setups for layout accessories. Hardware Support: Using Arduinos with JMRI

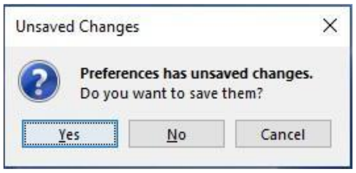
Save all your Preferences changes and Restart to load the new configuration into Decoder Pro.

Press the 'Save' button at the bottom left and a panel will pop up and ask if you want to restart now or restart later.
When you're done changing items in Preferences press 'Restart' & 'Yes' and your choices will start with Decoder Pro from now on
.


Restart DecoderPro?

You must restart DecoderPro.
Your new settings will not take effect until you restart this application.

and /or


Unsaved Changes

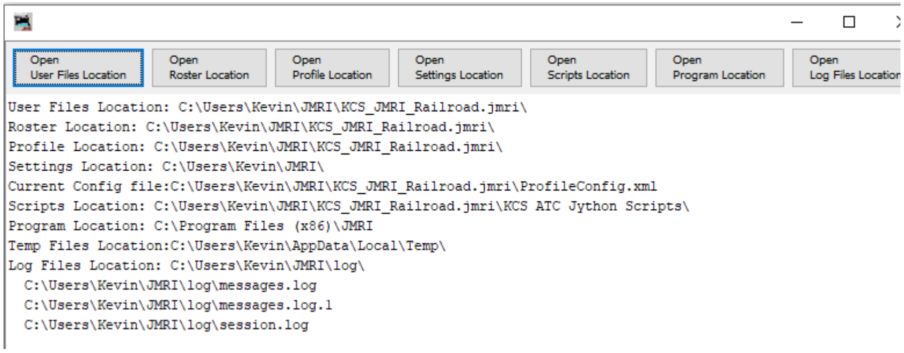
Preferences has unsaved changes.
Do you want to save them?

You Must Restart DecoderPro inorder for Preferences changes to take effect.

Utilities and File Management

DecoderPro Roster menu 'Help' > 'Locations'... files

Opens the Panel which shows you where your user profile files are stored and allows you to directly access user file folders.



```
User Files Location: C:\Users\Kevin\JMRI\KCS_JMRI_Railroad.jmri\
Roster Location: C:\Users\Kevin\JMRI\KCS_JMRI_Railroad.jmri\
Profile Location: C:\Users\Kevin\JMRI\KCS_JMRI_Railroad.jmri\
Settings Location: C:\Users\Kevin\JMRI\
Current Config file:C:\Users\Kevin\JMRI\KCS_JMRI_Railroad.jmri\ProfileConfig.xml
Scripts Location: C:\Users\Kevin\JMRI\KCS_JMRI_Railroad.jmri\KCS ATC Jython Scripts\
Program Location: C:\Program Files (x86)\JMRI
Temp Files Location:C:\Users\Kevin\AppData\Local\Temp\
Log Files Location: C:\Users\Kevin\JMRI\log\
  C:\Users\Kevin\JMRI\log\messages.log
  C:\Users\Kevin\JMRI\log\messages.log.1
  C:\Users\Kevin\JMRI\log\session.log
```

Simply click on the 'Open file location' button to open the file folder.

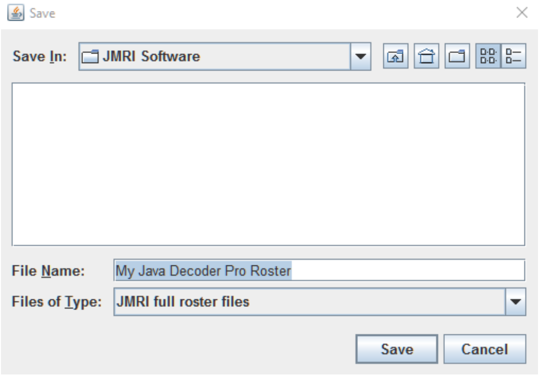
Back up 'Export Roster' & 'Import Roster File Procedures'

You can export and import Roster files from JMRI, allowing you to take them to another layout which has a JMRI software connected DCC Command Station and upload and run your engines with operations information and function key mapping of your engines onto their throttles & layout.

DecoderPro Roster menu click 'File', then

'Export Roster Entry' (for a single roster line, "ID")
or
'Export Roster' (for the Full Roster all entries.)

Give it a recognizable file name or your initials "ABC JMRI Decoder Pro Roster" then press Save.



Save In: JMRI Software

File Name: My Java Decoder Pro Roster

Files of Type: JMRI full roster files

Import Roster

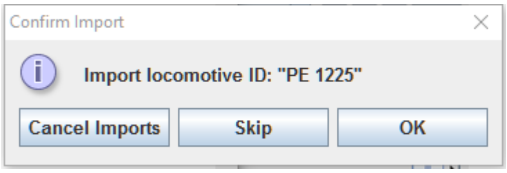
DecoderPro Roster menu

When you click 'File' > 'Import a Full Roster' it will prompt you to either;

Cancel Import

Skip this engine

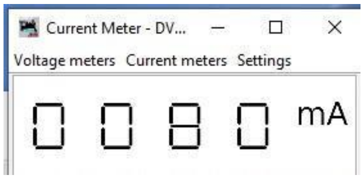
or OK import this engine.



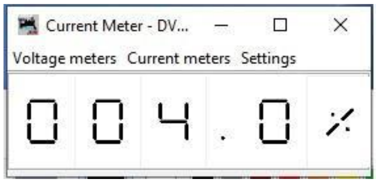
Import locomotive ID: "PE 1225"

Measuring Track Current

Main Track current may be measured by clicking on ‘DCC++’ > 'Voltage / Current meter' line from the DecoderPro Roster menu
This is the Current in Milliamps & as a Percentage of the Amp rating of your particular Motor Shield or external Booster.
i.e. a 10% on a 2 amp shield is .20amp draw. Various Current meters and decimal settings are available in the menu.options.



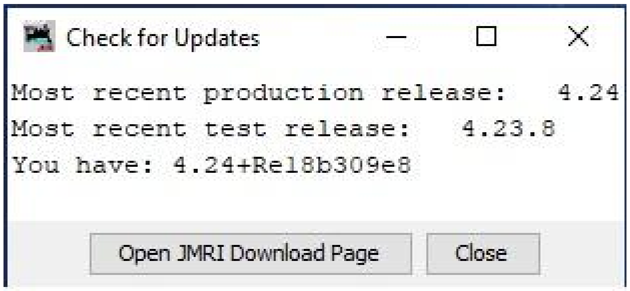
Current in MilliAmps



Current as Percent

Check for Updates

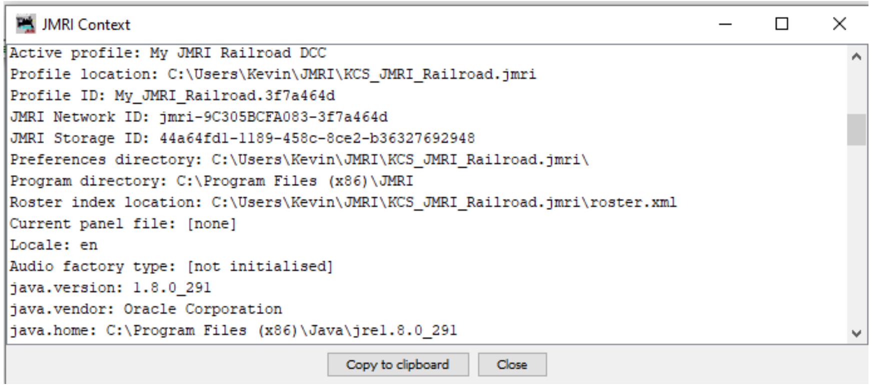
DecoderPro Roster menu *‘Help’* > ‘*Check for Updates’*



```
Most recent production release:  4.24
Most recent test release:   4.23.8
You have: 4.24+Rel8b309e8
```

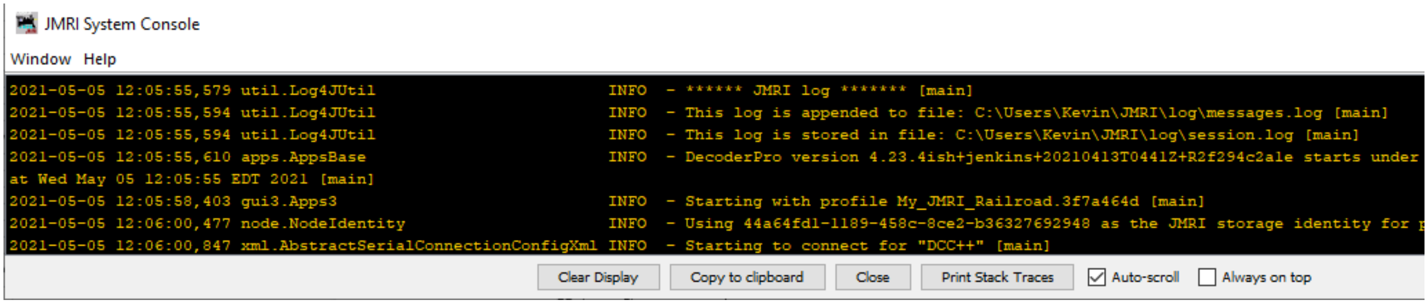
The Next two windows are JMRI serial data monitors for Startup and System operation & communications feedback
They both produce valuable information to provide JMRI support personnel to help determine the cause of any issues.

DecoderPro Roster menu ‘Help’ > ‘Context’ window



```
Active profile: My JMRI Railroad DCC
Profile location: C:\Users\Kevin\JMRI\KCS_JMRI_Railroad.jmri
Profile ID: My_JMRI_Railroad.3f7a464d
JMRI Network ID: jmri-9C305BCFA083-3f7a464d
JMRI Storage ID: 44a64fd1-1189-458c-8ce2-b36327692948
Preferences directory: C:\Users\Kevin\JMRI\KCS_JMRI_Railroad.jmri\
Program directory: C:\Program Files (x86)\JMRI
Roster index location: C:\Users\Kevin\JMRI\KCS_JMRI_Railroad.jmri\roster.xml
Current panel file: [none]
Locale: en
Audio factory type: [not initialised]
java.version: 1.8.0_291
java.vendor: Oracle Corporation
java.home: C:\Program Files (x86)\Java\jre1.8.0_291
```

DecoderPro Roster menu ‘Help’ > ‘System Console’ window



```
2021-05-05 12:05:55,579 util.Log4JUtil                       INFO  - ****** JMRI log ******* [main]
2021-05-05 12:05:55,594 util.Log4JUtil                       INFO  - This log is appended to file: C:\Users\Kevin\JMRI\log\messages.log [main]
2021-05-05 12:05:55,594 util.Log4JUtil                       INFO  - This log is stored in file: C:\Users\Kevin\JMRI\log\session.log [main]
2021-05-05 12:05:55,610 apps.AppsBase                        INFO  - DecoderPro version 4.23.4ish+jenkins+20210413T0441Z+R2f294c2a1e starts under
at Wed May 05 12:05:55 EDT 2021 [main]
2021-05-05 12:05:58,403 gui3.Apps3                           INFO  - Starting with profile My_JMRI_Railroad.3f7a464d [main]
2021-05-05 12:06:00,477 node.NodeIdentity                    INFO  - Using 44a64fd1-1189-458c-8ce2-b36327692948 as the JMRI storage identity for
2021-05-05 12:06:00,847 xml.AbstractSerialConnectionConfigXml INFO  - Starting to connect for "DCC++" [main]
```

So that's a quick introduction & setup of the DCC++EX Command Station and JMRI Decoder Pro software.
For more information on DecoderPro 4.24 and later see the JMRI web site and all the ***Other Useful Links*** below.

Troubleshooting

Arduino issue;

If you are experiencing intermittent faults with your DCC++ Command Station, please ensure that you are using a good quality regulated 12vdc 1amp for Z scale, 12-15vdc for N scale and 15-18vdc with 2 to 5amp power supply for HO scale.

When using Arduino Interactive Development Editor IDE and it Fails to compile or upload firmware, check your Command Station-EX.ino software edits and make sure you use // to comment out a line that is a comment and it is not mistaken for C++ code. Also, try a different USB cable from the Arduino to PC connection as yours may be faulty and may cause it to fail in uploading..

DCC++ Issue:

There is a DCC support page which will be updated to reflect the most common questions people have about DCC++.
DCC++ & DCC++EX Development & Support; DCC++ Development & Support on Discord Channel
Diagnostic testing Arduino and DCC++ Classic DCC++ Classic Diagnostics---D---Command
Diagnostic testing Arduino and DCC++EX DCC++ EX Diagnostic---D---Commands

JMRI issues;

One common issue is the configuration of the "Virtual COM Port" for USB. Check your PC Device Manager to make sure your COM x port is set to 115200 baud to match the baud rate in the DCC++ systems.
First try unplugging the USB cable from the Arduino and plugging it back in to see if that reconnects.

Please review the information in the section "Edit DecoderPro Preferences > Connections".
If this is not successful, as a last resort, it is often effective to delete the file containing the Preferences settings.
This file is in the user's data area on the computer and is located & named C:/Users/Name/JMRI /*DecoderProConfig3.properties.*
Be sure that Decoder Pro is not running, locate and delete this file, and then restart Decoder Pro once more.
Set preferences as before, Save and restart Decoder Pro once again.

Other Useful Links; **The links throughout this PDF document are clickable, Or by pressing Ctrl + Click mouse**

DCC++ & DCC++EX discord general **Support on discord channel**
Getting Started with Decoder Pro See the DecoderPro website; JMRI Java Model Railroad Website for any updates and latest information. Join the JMRI user group jmriusers@groups.io | Home for help from DecoderPro users and the team who develop it.

Java Model Railroad Interface JMRI

Java Model Railroad Interface DecoderPro Symbolic Programmer
JMRI Supported on DCC++ JMRI Hardware Support - DCC++
Download & install Java Runtime 1.8 on Windows PC, Java for Windows - Offline Installation
Download & install JMRI 4.24 or newer JMRI Software Downloads
JMRI Install & DecoderPro video; JMRI Overview & Install DecoderPro video long intro to JMRI Suite of products
JMRI Support on Windows JMRI with Windows
Install JMRI on a Raspberry Pi instead of a PC, MSteveTodd how to guide: MSteveTodd JMRI-Raspberry Pi Image
DCC CV Calculator Digitrax CV Calculator
Manufacturer's ID: DCC_Shortcuts Card.pdf

DCC++ Command Stations

DCC++ EX Home Page DCC++ EX Model Railroading
DCC++ EX Command Station Setup DCC++EX Getting Stated & Advanced WiFi Config DCC-EX WiFi Config
DCC++ EX YouTube Channel. DCC-EX YouTube
DCC++EX DIY & Digital Railroad DCC++EX Game Changer
DCC++ Facebook page DCC++ & Arduino Facebook page
DCC++ EX TrainBoard DCC++ Update Project 2020
DCC++ EX Wiki page Wiki DCC++ EX Command Station
DCC++ Classic Wiki page Wiki DCC++ Classic Base Station

Thirdparty & Peripherals;

Engine Driver for Android Smartphone Engine Driver: Home
DCCpp CAB for Android Smartphone DCCpp CAB Throttle download
WiThrottle for Apple Smartphone WiFi Throttle: Home
Dr. Geoof Bunza, Master of DCC Animation & Accessories MRH Magazine Geoff Bunza Journal
DCC++ Dave Bodnar, Developer of DCC Throttles & Motor Boosters & Gadgets; My Experiments with DCC++

Arduino Microcontrollers

Arduino Microcontrollers Arduino Official Store
Download & install Arduino Interactive Development Environment IDE (editor) Arduino Software (IDE)
Arduino IDE Installation Instructions Install the Arduino Software (IDE) on Windows PCs
Arduino Online Frequently Asked Questions FAQ's Arduino Help Center
Arduino and JMRI & CMRI Bus Accessory interface Hardware Support: Using Arduinos with JMRI

Addition Technical Notes:

Arduino

Reference to Arduino DCC++ EX3.1 DCC++EX Getting Started Building & Setup

Specification/Operating Conditions

Arduino Uno 328P Revision 3 or Mega2560 Revision 3 microcontrollers
Input; 7 - 12vdc regulated with a Arduino Vin output supply current 3.3v to 5vdc.
(**Never** input DC power supply greater >12vdc in the Arduino female 2.1 plug)

Arduino L298P R3 Motor Shield dual full H- bridge driver acts as a booster (2amp per channel 4amp total)
Motor Shield signal pin jumper wires are no longer needed or used with a DCC++EX version software.
Input; 12 - 18vdc, 2a to 5 amp to the L298P Motor Shield (**Note cut Vin trace on bottom if > 12Vdc is used**)
Input voltage specifications recommend Not to exceed 18vdc.
Notes: Voltage output to main and programming should read 12vac to 16.5vac AC current for N scale to HO scale.

1. Minimum supply voltage depends upon the requirements of the decoder being programmed.
   In general, it is safer to use as low a voltage as possible in case of problems with a newly installed decoder.
2. DCC++EX will remove track power if output current exceeds standard as measured 100ms after applying power.
3. Arduino's are protected against reverse polarity connection but will not work unless the polarity is correct.

Note: Arduino DCC++EX Command Station has a new <1 JOIN> feature which logically joins the Programming track to operate as a Main track which allows a Loco to be Driven Away onto the main line. Care should be taken to make sure ' + - 'polarity is maintained when wiring the two insulated sections with dual nylon joiners in between and +v Right rail and -g Left rail.

Note of Caution; Connect the Programming Track
The programming track MUST be isolated from all other DCC or DC control systems and connected only to the DCC++EX CS. Damage will result to the Arduino Uno DCC++ Command Station or other equipment if this rule is not followed.

Connect the Programming track "B" and Main track "A" Terminals with the same polarity when used together in a Spur or Siding on a layout with right positive and left negative polarity. The DCC output voltage to the track will be approximately -1vac or less below the input vdc power supply voltage. Typically, a 15.5vdc input will show a 14.8vac on the track with the standard regulated power supply.

During programming the track current is sensed 100 milliseconds after the programming track is powered up.
If the current exceeds 250 milli amps then the programming track power is removed momentarily to avoid potential damage to an incorrectly installed or difficult decoder.

Tips on JOIN command & Programming Track

When the Command Station receives a program track command it does the following:
1 UnJOINs the prog and main track if it was previously joined
2.Turns on the prog track power if it was previously off
3. Performs the command
4. Resets the power and JOIN state to what they were before.

This is very convenient when handling single commands BUT the JOIN and power changes require a significant amount of extra time for the decoder to settle so it's much faster, when reading multiple CVs, to have the prog track power switched on with <1 PROG> before you start.

Getting additional on-line help DCC++EX support..

EX contains a comprehensive diagnostic trace of the CV handling process, this can be used to detect many kinds of issues such as power problems, reluctant decoders, out-of-spec responses etc.

If you are unable to get a decoder to behave, then the best way to get help is to run the following commands with a loco on your programing track:
<D ACK ON>
<R>
and then copy the resulting trace from the DCC++ Traffic Monitor and post it to the EX support discord support channel

This will give us important information on how to best proceed and help you with your decoder.
If you merely report that something "doesn't work" or "returns error -3" then the only answer you are going to get is "Something wrong," followed by a suggestion to do the above.

The <R> command is unique to EX, it examines the loco and returns the address which may be used to drive it.
This may be a long, short or even consist address and you don't have to read the CVs individually to work it out.
Bear in mind that EX (like DCC++) does NOT currently support decoders with a LONG address of <128."

The quickest way to change a loco address is to use the <W addr> command on the Programming track which will automatically manage the short/long settings and clear the consist address.

Advanced WiFi Configuration and WiFi Setup and usage of WiThrottle Server For JMRI and for DCC++EX Standalone CS. *For more detail see DCC++EX website* ***advanced-setup wifi-config***

WiFi Connectivity;
You have the option of allowing the Mega WiFi enabled setup to have a WiFi connection to either a;
JMRI WiThrottle Server software connection through either a Windows or Mac iOS PC, or a Raspberry Pi microprocessor.
Or, by Directly connecting to the DCC++EX Command Station with WiFI and the embedded WiFi WiThrottle Server feature.

The setup can be performed either through the automated exInstaller program or directly by using the Arduino IDE to edit the CommandStation-EX.ino sketch manually.

By opening and editing the 'config.h' file to configure the SSID "local name" and PASSwrd "local password" if you plan to use a router and setup in Station STA mode. Otherwise the Command Station will Default to Access Point AP Mode and then will default to the name 'DCCEX_xxxxxx' and PASSwrd "DCCEX_xxxxxx" were the xxx will the unique six digit device number either the Shield or Onboard ESP WiFi chip. For further information see DCC-EX.com Configuring WiFi and also WiFi Setup.

Fixed IP Addresses
There are 2 ways to setup a fixed address in Station mode:
1. configure the Mac ID in your home router to always give it the same address. ie., 192.168.1.xxx

2. #define DONT_TOUCH_WIFI_CONF true in config.h and then give the ESP chip a fixed address in STA mode with the native AT+commands. In DCC++EX you can give these by typing <+ > plus the appropriate AT command. See DCC++EX website.

-- E N D --