Intro to JMRI and DecoderPro

Dave Duchamp

Dick Bronson (Co-Presenter)

Bob Jacobsen (Co-Presenter)

Adapted by Scott Jay for
MFMR 2011 Convention

What is JMRI?

JMRI (Java Model Railroading Interface) is open source (free) software for connecting a model railroad layout to a computer, and performing various model railroading tasks via the computer.

JMRI was/is developed by a group of volunteer programmers under the leadership of Bob Jacobsen.

JMRI uses the Java programming language.

JMRI continues to grow . . .

How is JMRI Organized?

JMRI has of an extensive library of model railroading software, and several front-end applications focusing on different areas of model railroading.

All JMRI applications use this common library.

JMRI Applications include:

- DecoderPro - Programming DCC decoders.
- PanelPro - Layout display for running trains.
- Other applications (JMRIDemo, LocoTools, etc.)

What Computer Systems are Supported by JMRI?

Windows - XP, 2000, 98, 98SE

Macintosh - MacOS X, Classic

Linux

What Model Railroading Systems are Supported by JMRI?

Loconet - Digitrax (Chief, Empire Builder, Zephyr), Intellibox

Lenz - LI100,LI100F,LI101,LIUSB

NCE

C/MRI

EasyDCC

Wangrow

SPROG

XPA Modem

ZIMO MX-1

ZTC Controls (ZTC640)

Direct Drive (Serial)

TMCC (Lionel)

Oak Tree Systems

What Model Railroading Tasks are Supported?

Programming DCC decoders
Computer CTC Control Panel
Computer throttles
Consisting
Control of Turnouts
Routes (Controlling groups of Turnouts)
Control of Layout Lighting
Fast Clock
Control of Signals and More …

How do I get started?

1. Connect your computer to your model railroad layout.
2. Download JMRI.

   One download contains all JMRI applications.

   Large download - CD's are available.
3. Configure JMRI.
4. Test communications.

Detailed instructions for various computers and model railroading systems are on JMRI web site.

No computer programming is required.

Computer Connection Example

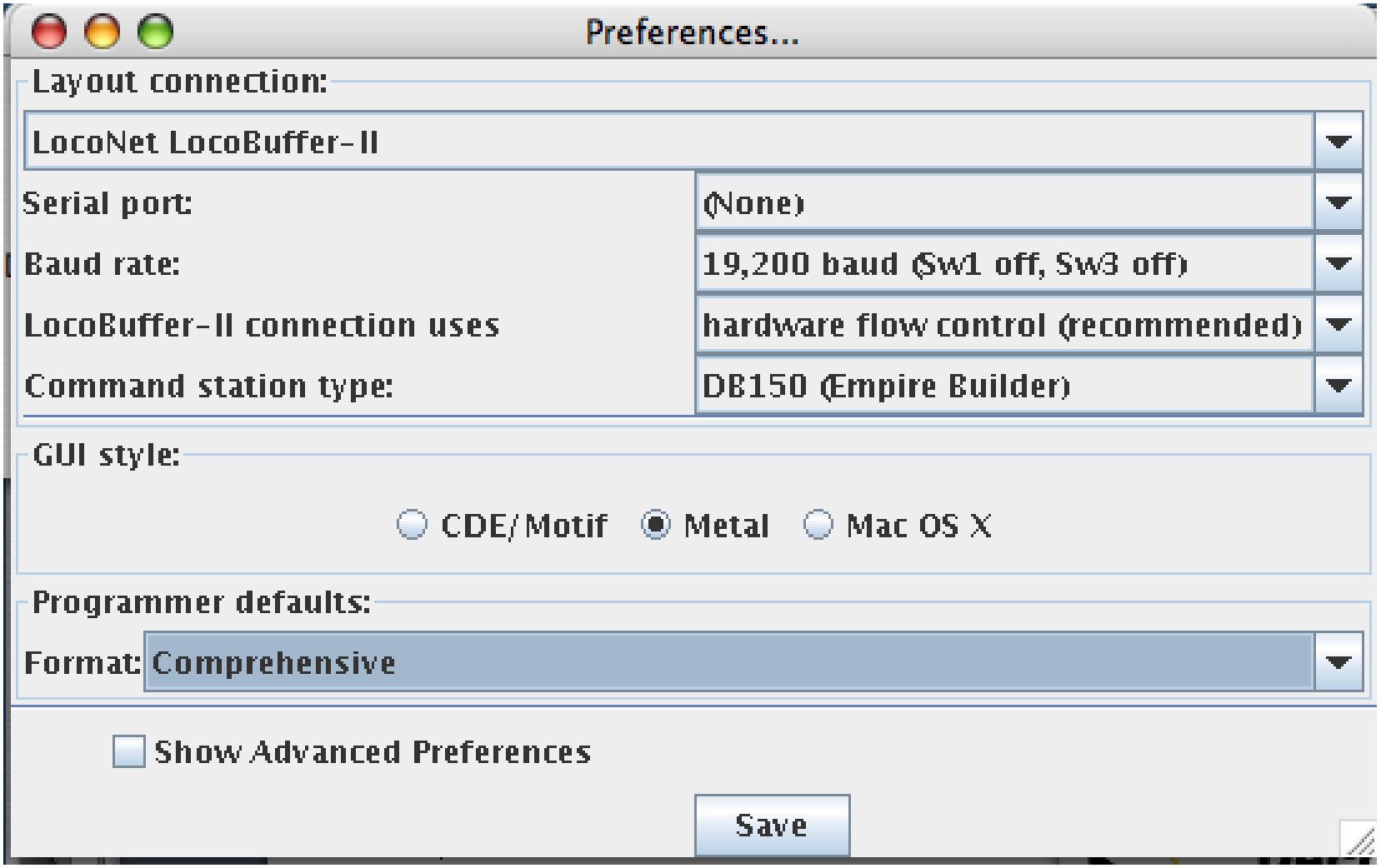
Workshop system:

Lenz LZV100

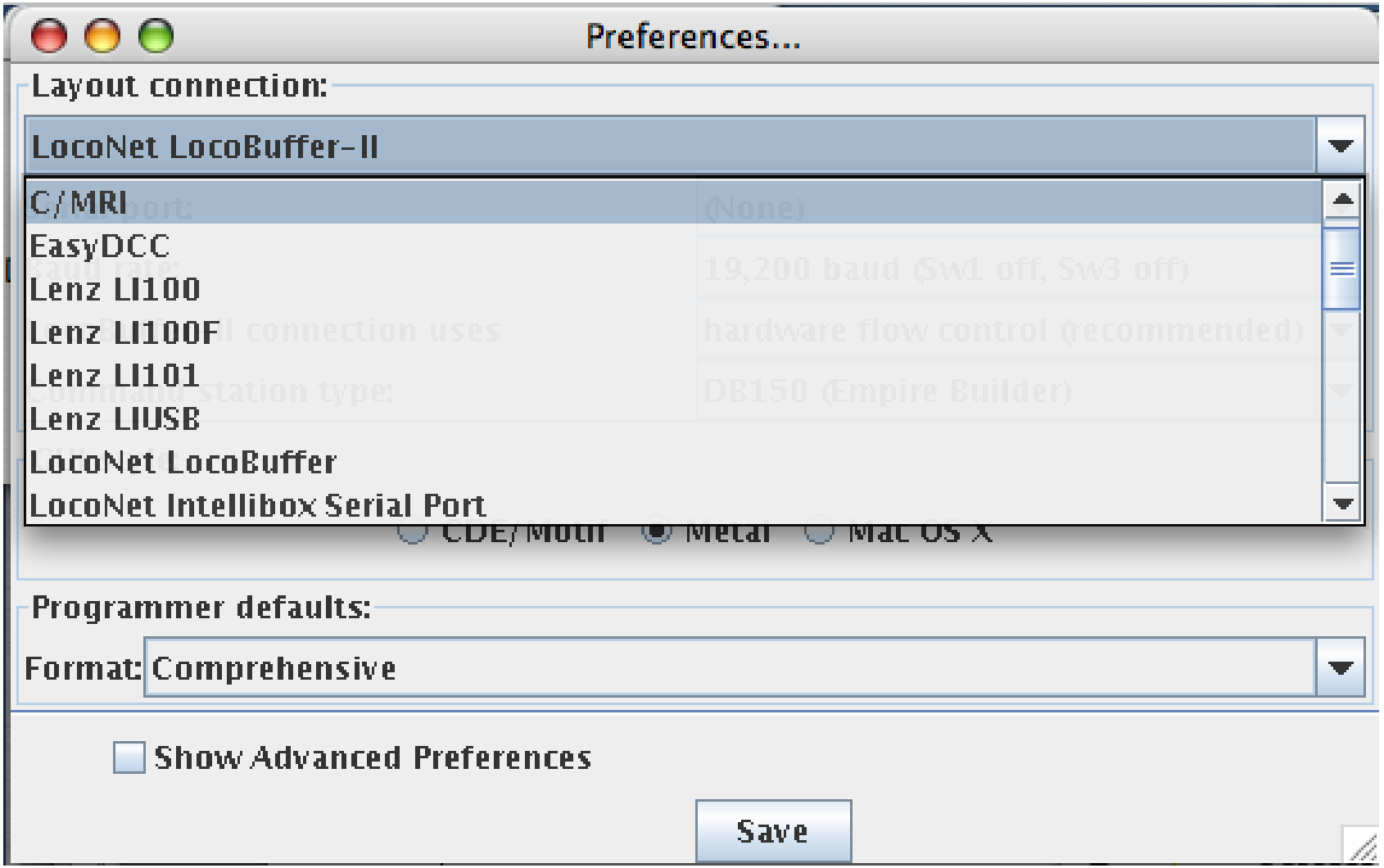
Lenz Li101F (with USB-Serial Adapter)

Acer Aspire One Netbook

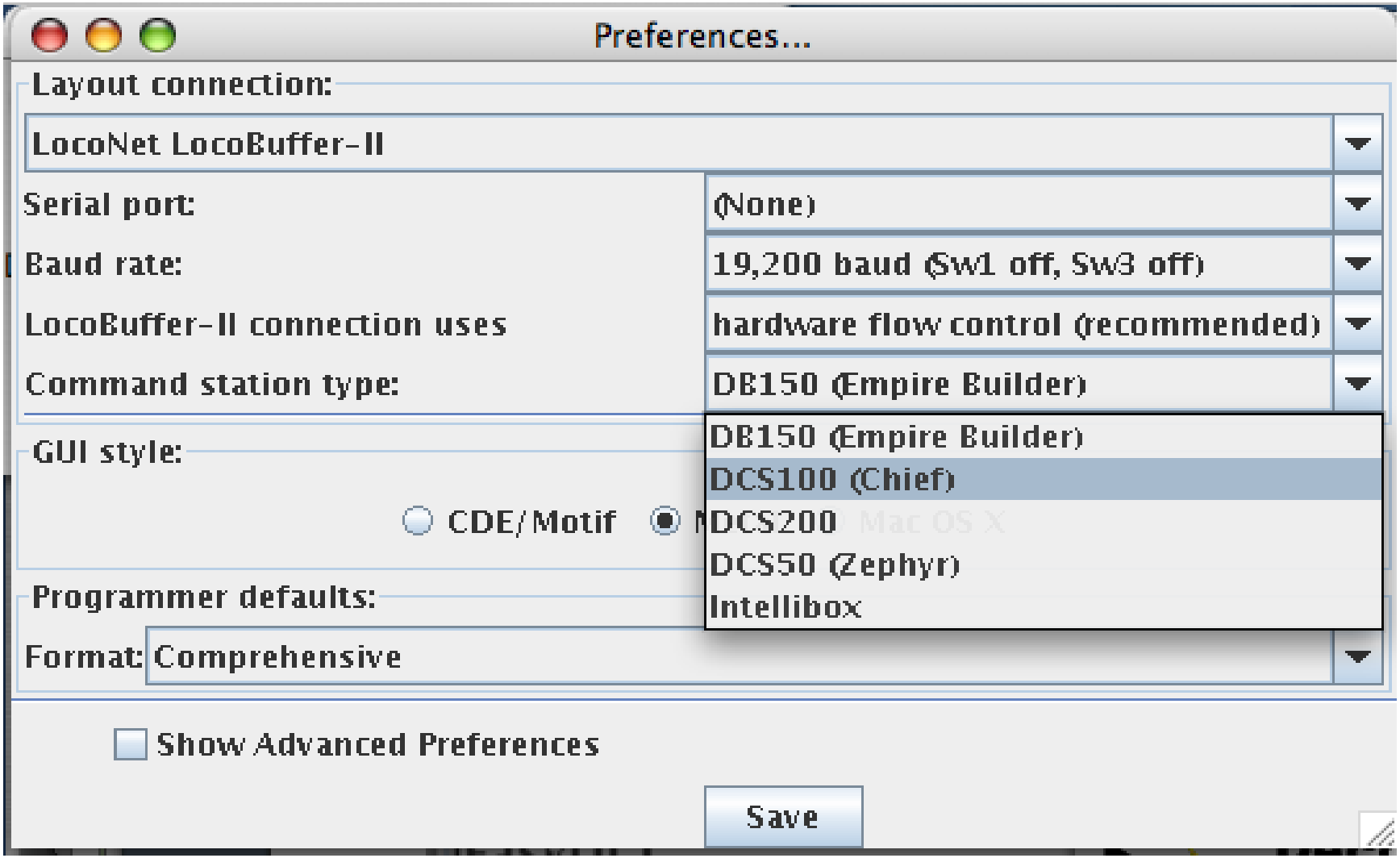
Configuration Panel



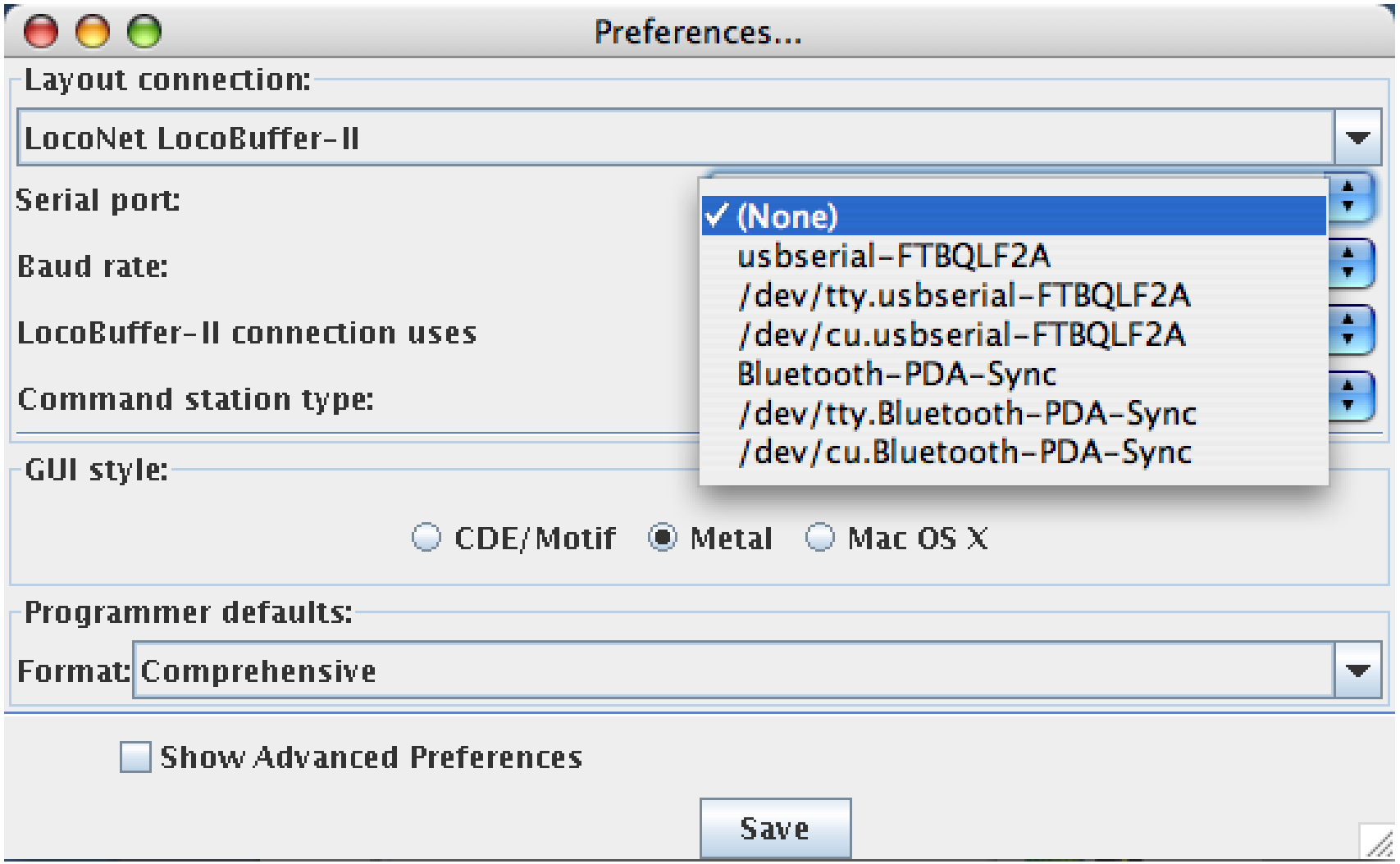
Select the type of layout connection from an extensive pull-down menu.



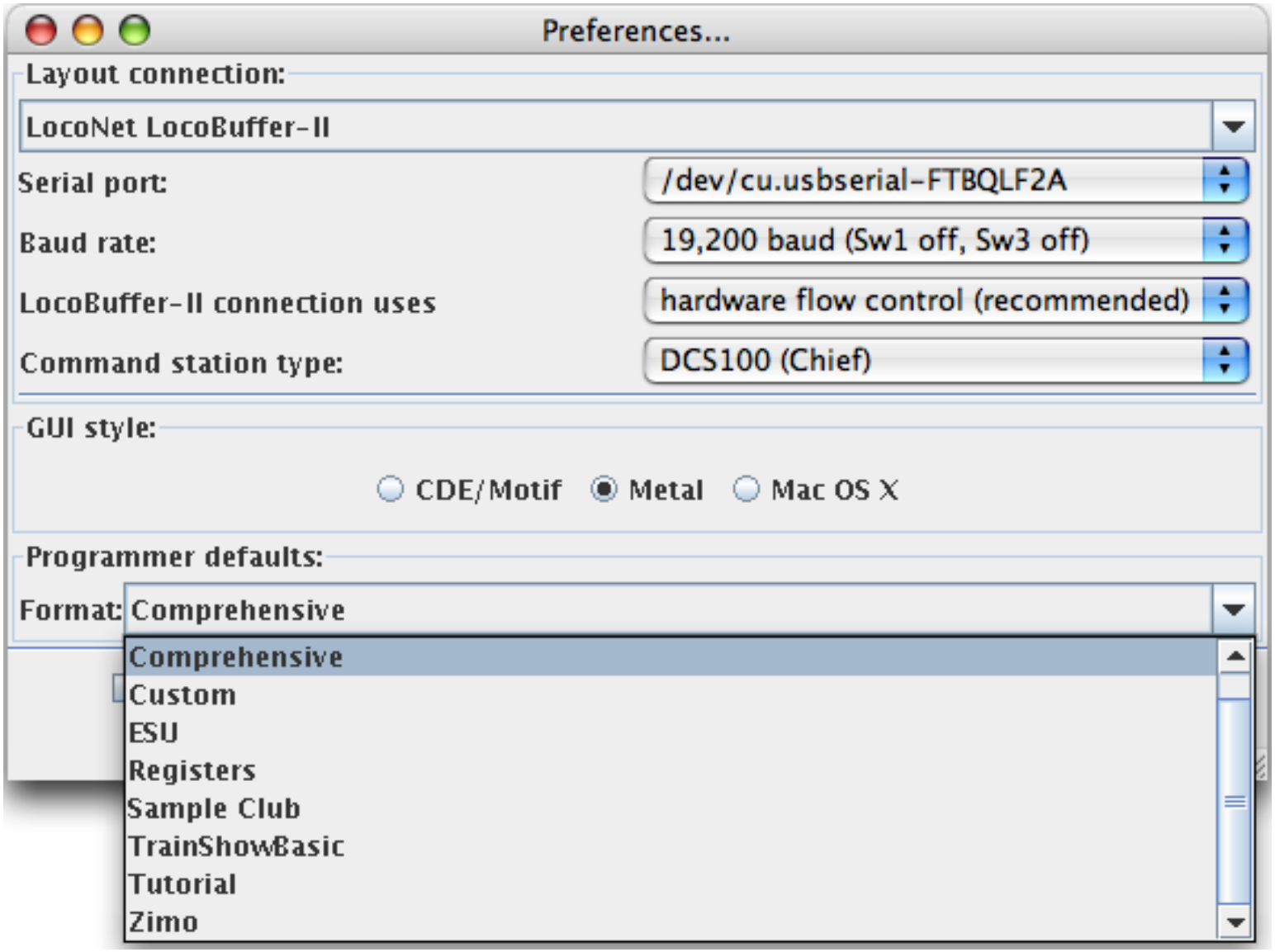
Select the command station type from the menu of types compatible with the layout connection.



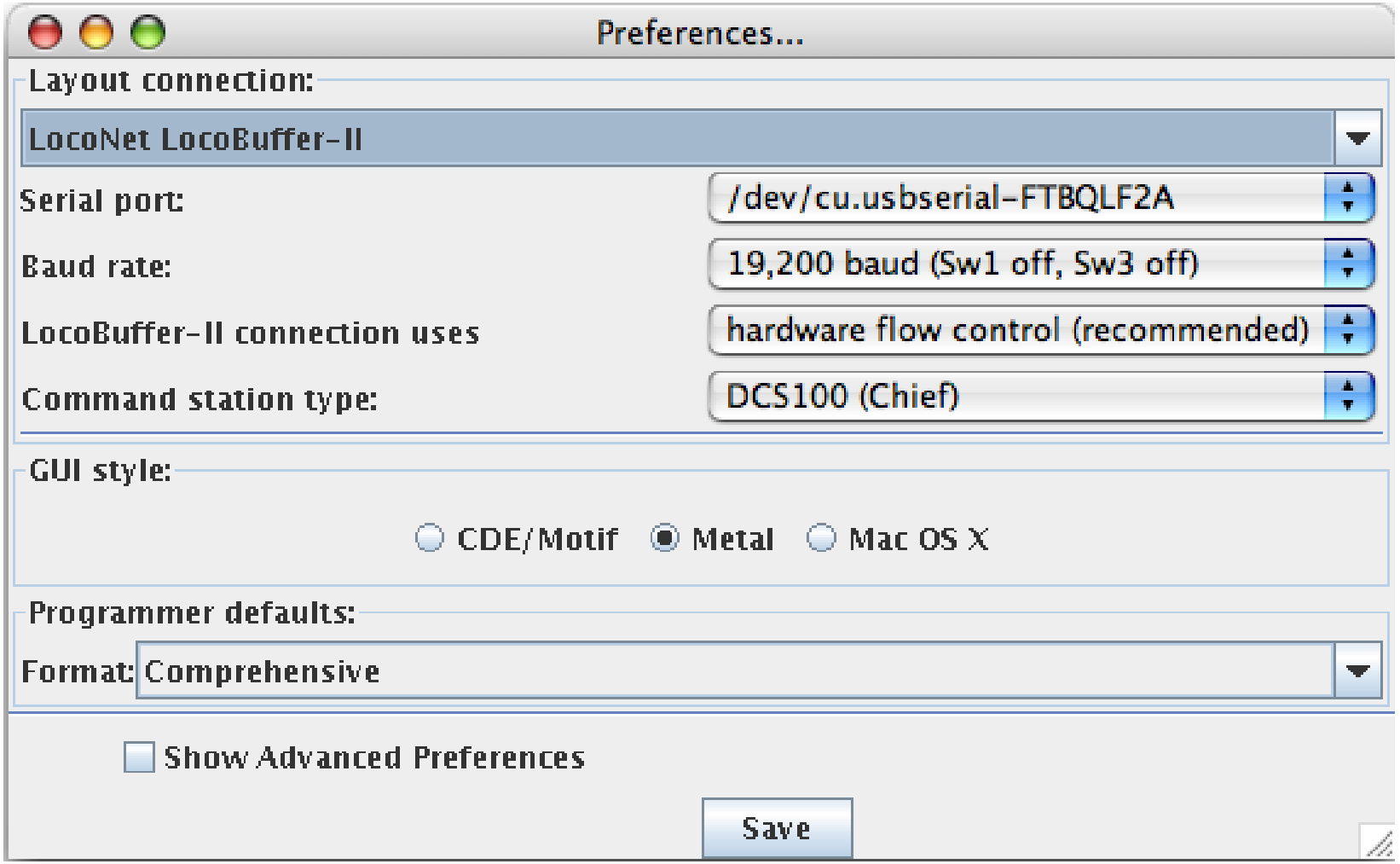
Select a serial port from the menu listing the serial ports that JMRI can detect.



Select a default programmer (Comprehensive is usually best).

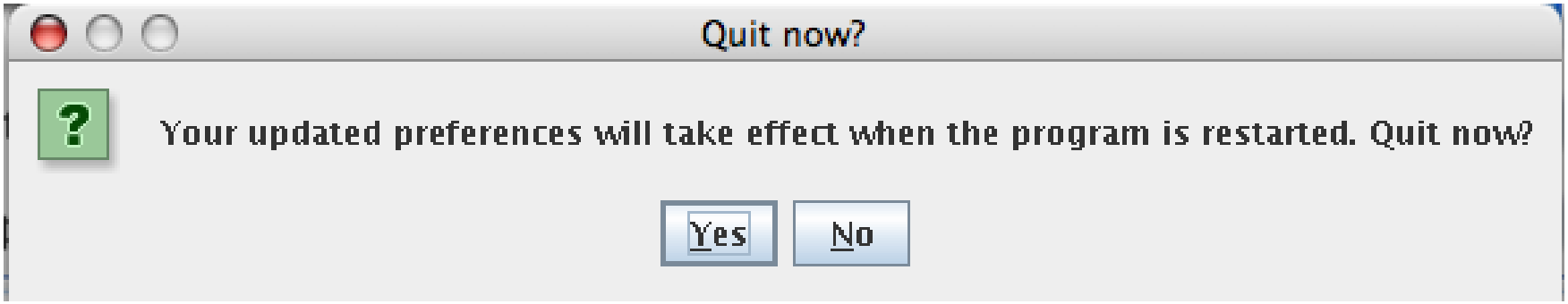


Click the “Save” button to write the connection configuration to disk.



Click the “Yes” button, to quit the program.

Restart the JMRI application.



Notes: Restart is required anytime preferences are changed for the preferences to take effect.

Preferences must be set for each JMRI application. They each have separate preferences files.

The program is set up according to the saved preferences.

Note: Startup window contains program version and Java version, in addition to connection information.

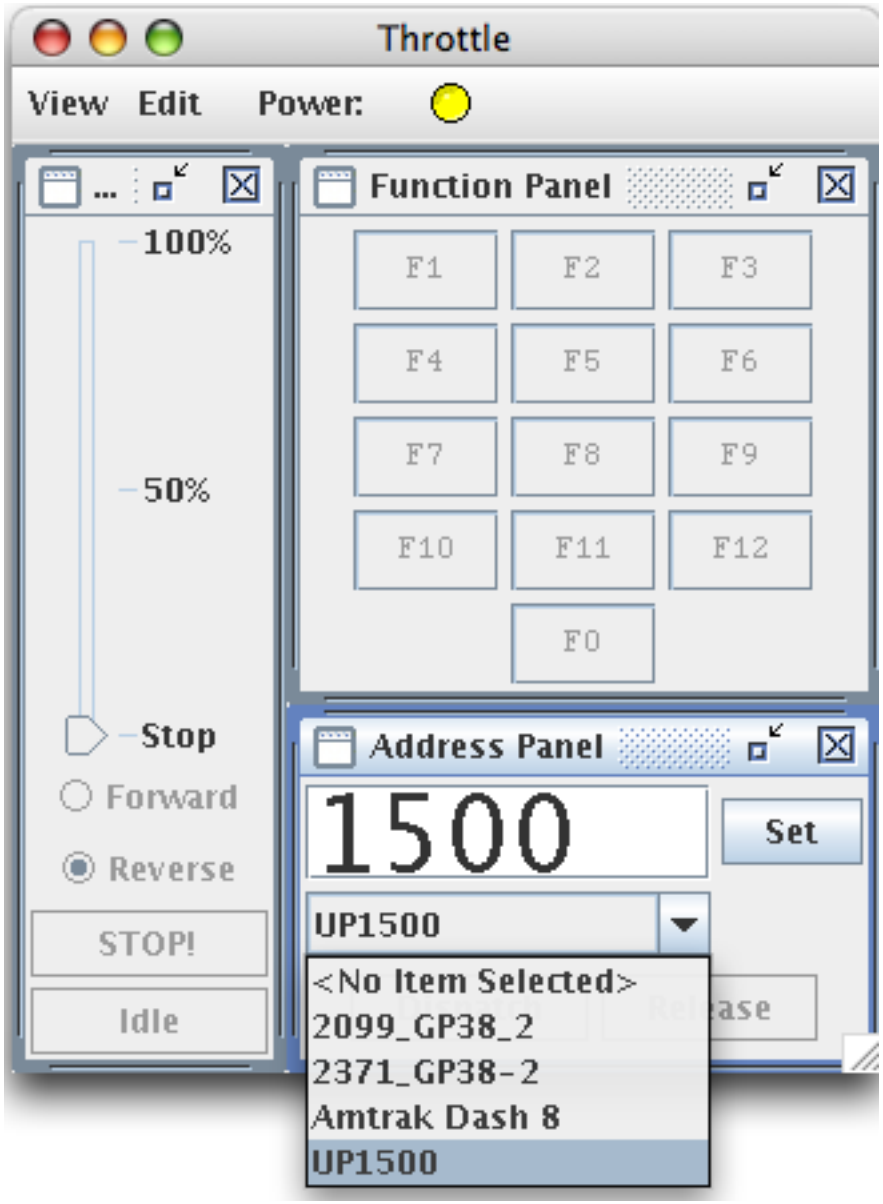
Connection Testing Example

Select a train from the Roster.

- or -

Type in a locomotive address.

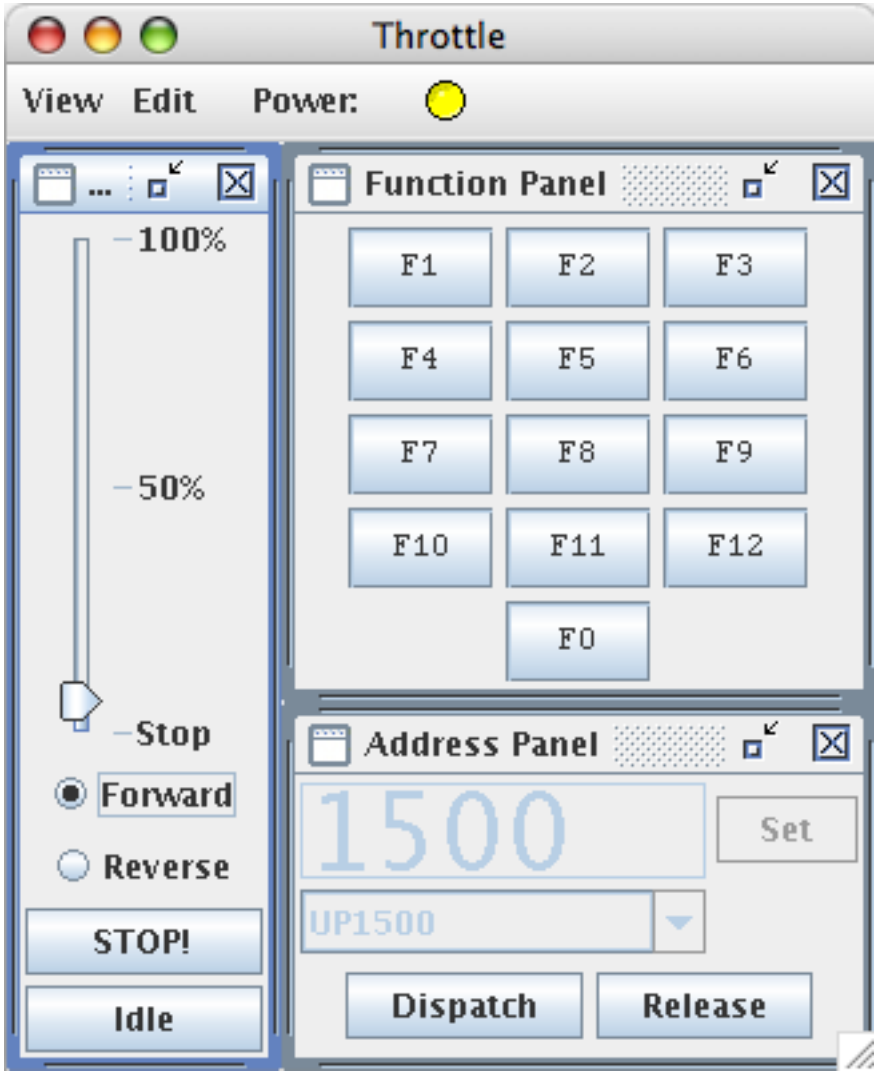
Click the “Set” button.



Connection Testing Example

Run a train from the computer.

If all works OK, then JMRI is successfully communicating with your command station.



Configuration preferences may be accessed at any time via the Edit menu.

Advanced Preferences

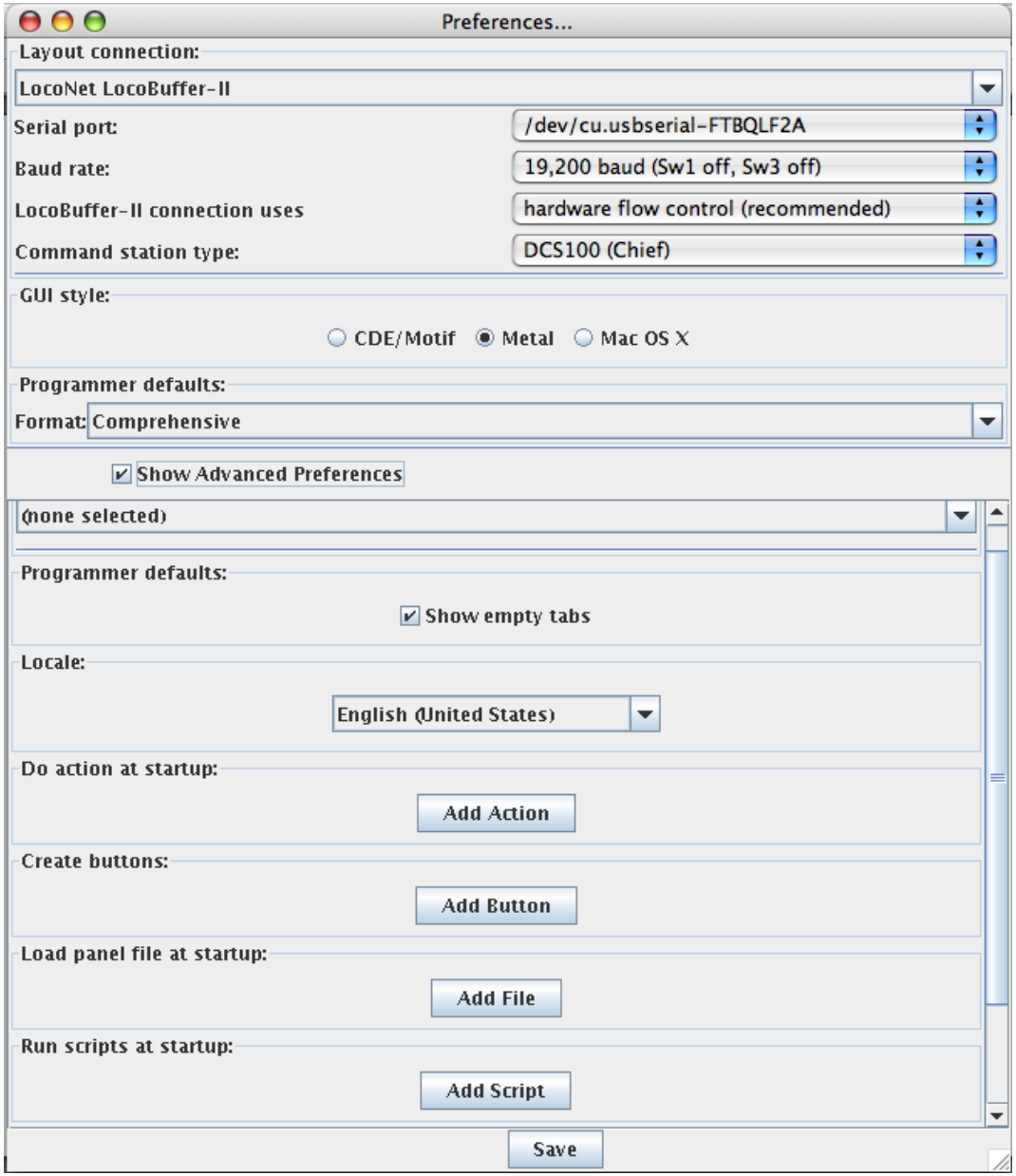
->

Allow many useful options including:

A second layout connection.

Automatic loading files at startup.

Running scripts at startup.



How do I get help?

1st - The JMRI web site - *http://jmri.sourceforge.net/*

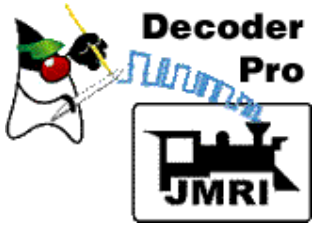
Documentation and detailed instructions

2nd - JMRI Yahoo discussion group.

jmriusers

Monitored by many JMRI ‘experts’, eager to provide help.

Information on JMRI web site on how to sign up.



What is DecoderPro?

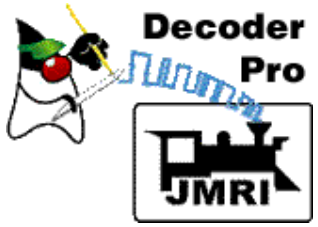
DecoderPro is a better tool for programming DCC decoders.

DecoderPro simplifies the job of configuring complicated DCC decoders.

DecoderPro is a JMRI application.

DecoderPro supports mobile decoders (decoders in locomotives).

DecoderPro supports some static decoders.



Basic Terminology

Decoder - small microcomputer based control unit

Mobile Decoder - Decoder in a locomotive, "decodes" DCC commands to control locomotive.

CV (Control Variable) - 8-bit data byte in a decoder that specifies user options.

Programming a Decoder - setting the values of the CV's to user's options.

Decoders have many CV's. Most CV's follow **NMRA Standards**, but some are vendor specific.

Each mobile decoder has an **Address** - a number that allows the locomotive to be uniquely identified.

Setting up an Address

Decoder (locomotive) addresses can be 2 digits or 4 digits on modern decoders and DCC throttles.

Usually set the address to the locomotive number.

Most decoders are set to address 03 on arrival.

A locomotive will respond to speed control and function commands that bear its address.

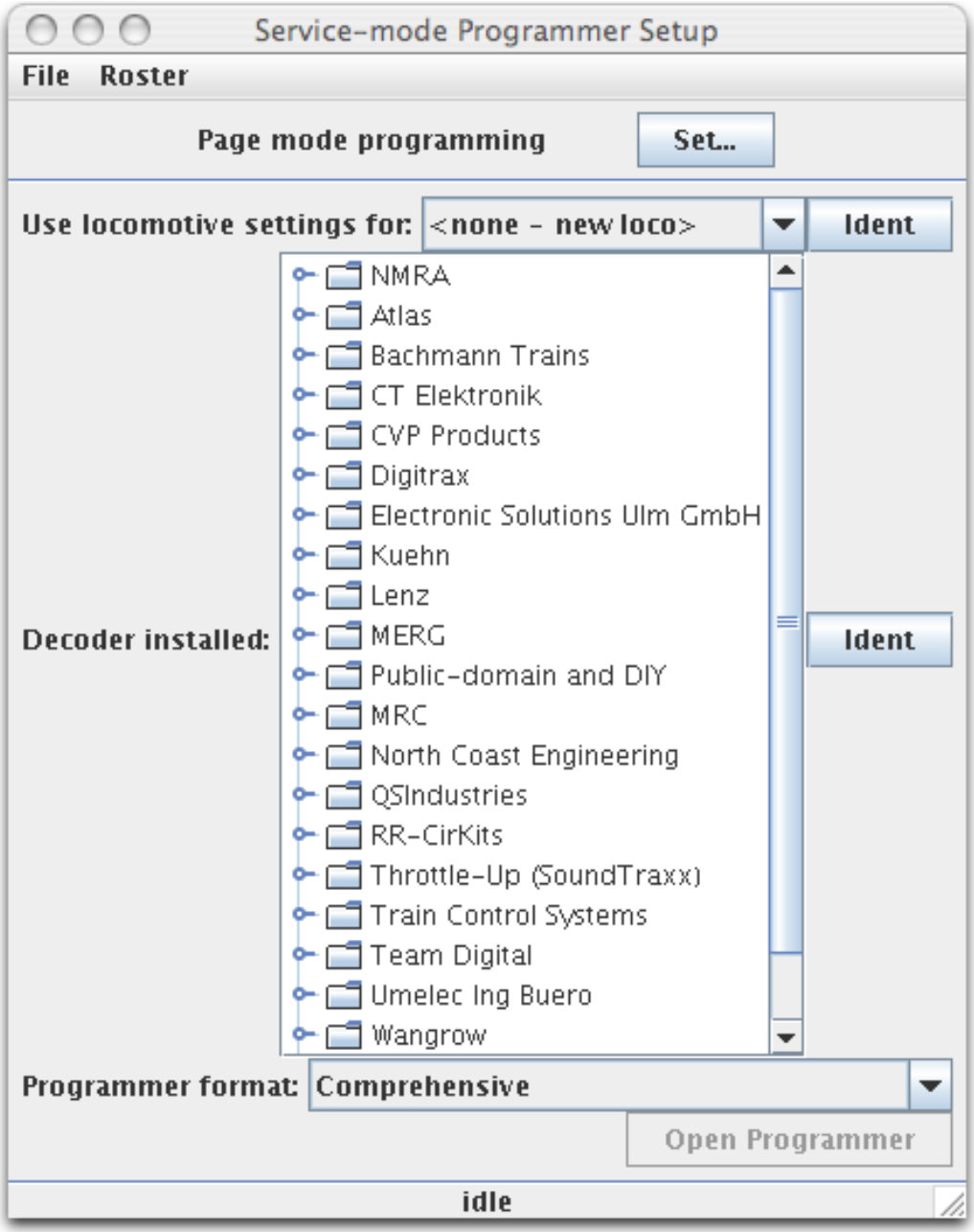
Setting the address is usually the first (and sometimes the only) programming needed.

It's easy to set up an address in DecoderPro.

Example - Setting the address of a new decoder

Put the locomotive with the new decoder on the programming track.

Start Decoder Pro. When the window below comes up, click on "Use programming track …".



The NMRA standards have two CV's to identify a decoder:

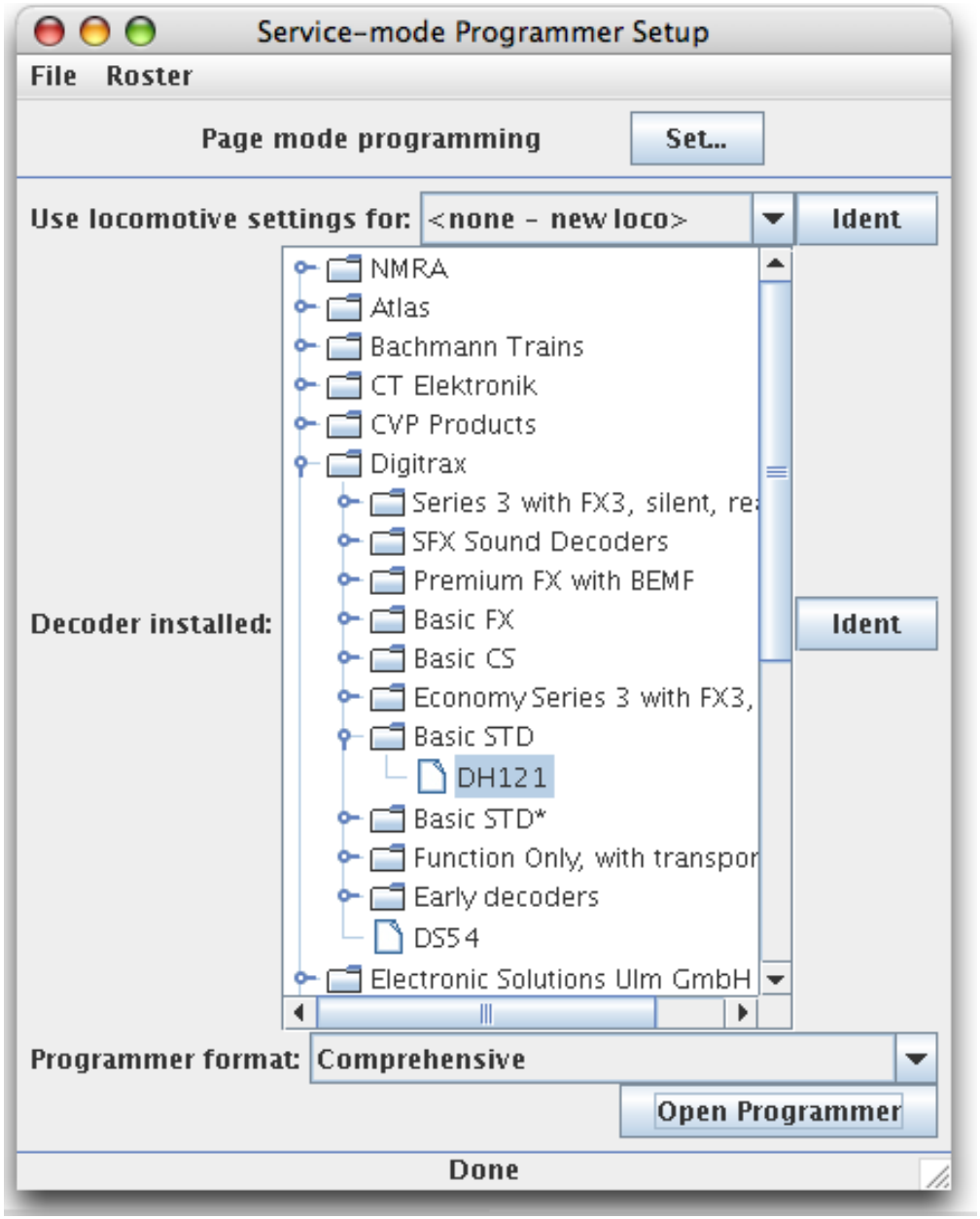
CV8 - Manufacturer ID

CV7 - Manufacturer Version Number.

Both are read only.

<- Click here to have DecoderPro attempt to identify the decoder by reading these CV's.

Note: Some command stations cannot read CV's! For these, select the decoder in the list manually.



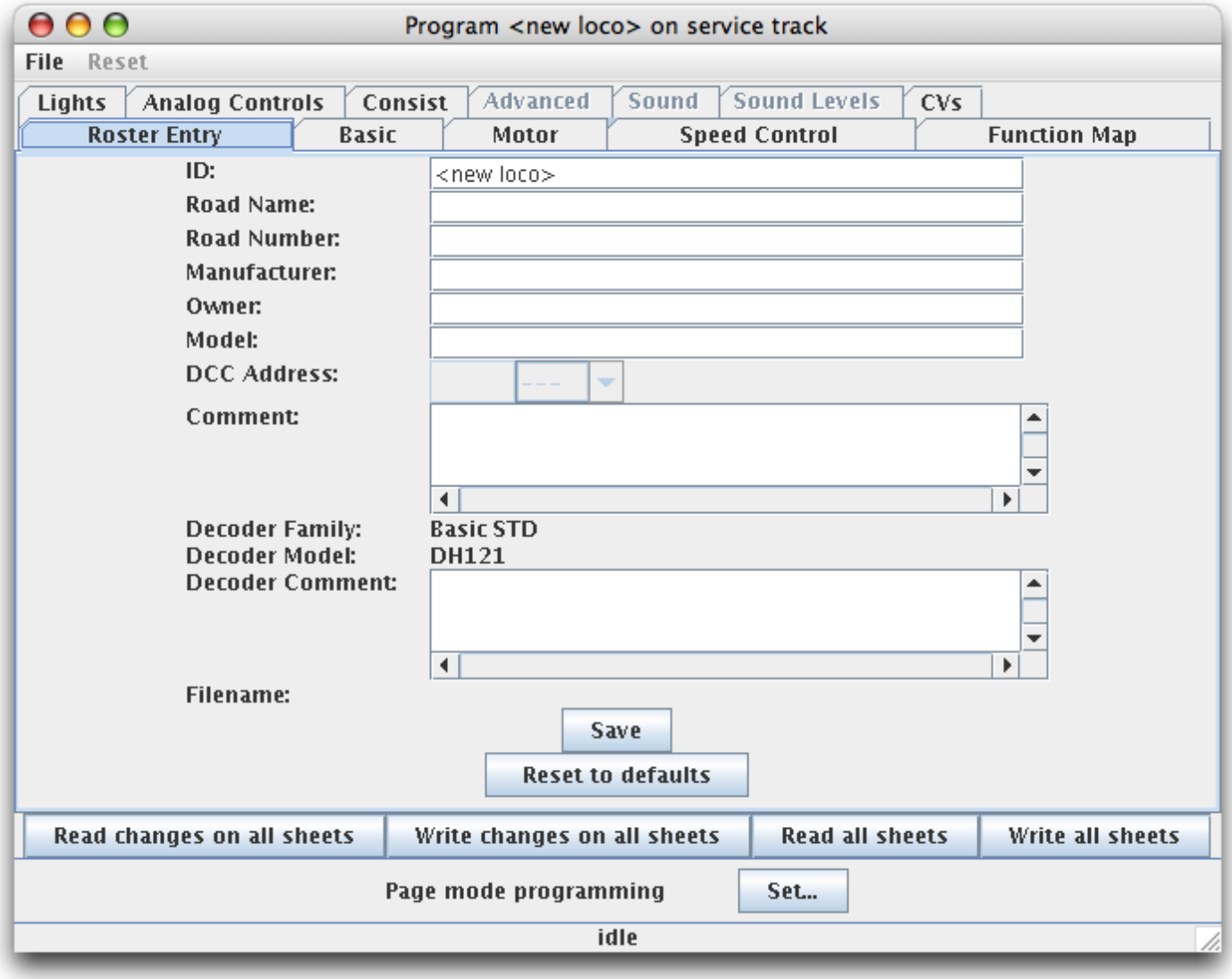
DecoderPro identified the decoder as a Digitrax DH121.

(Sometimes the user has to choose among several possibilities.)

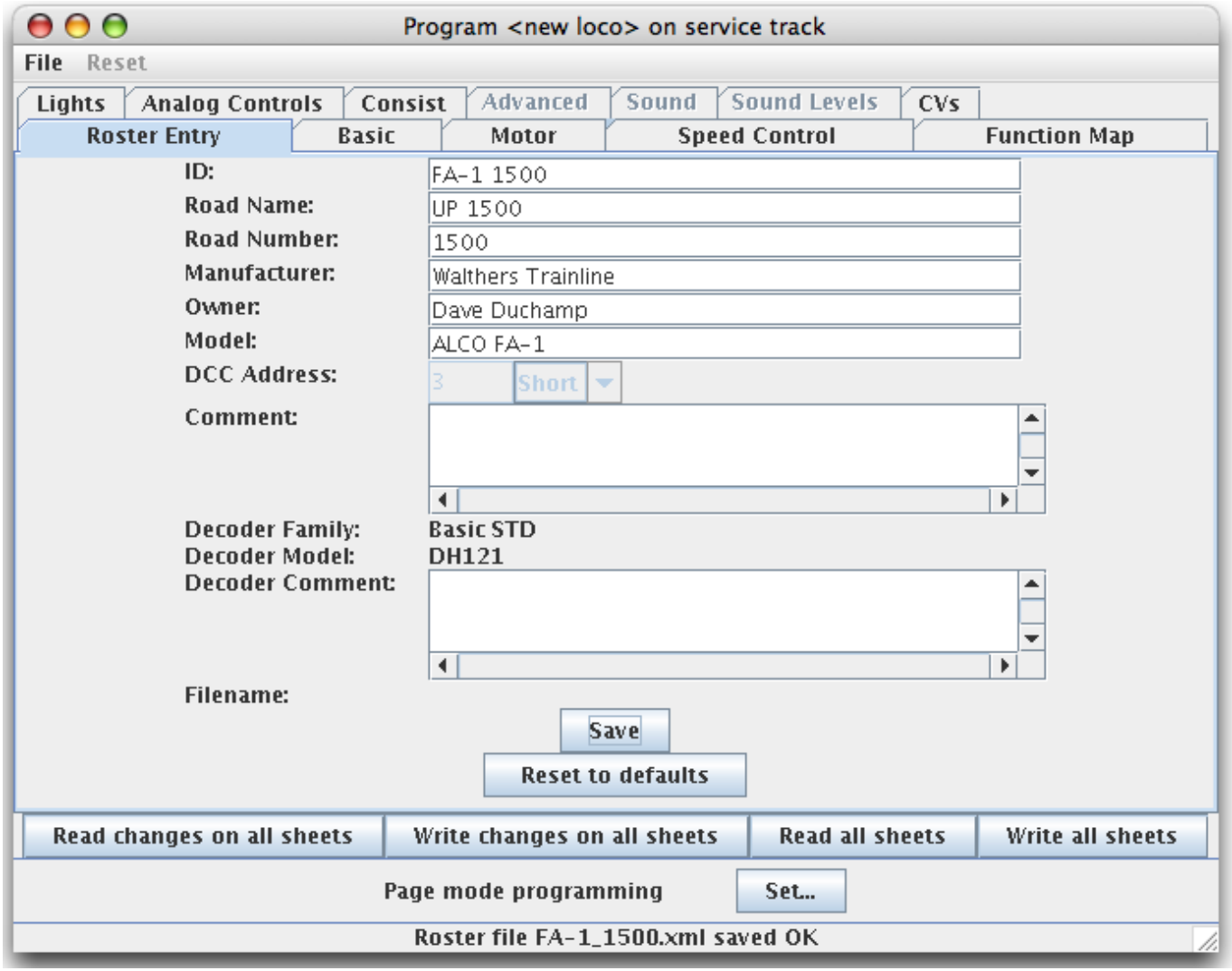
Check Programmer format, and click on "Open Programmer".

<-

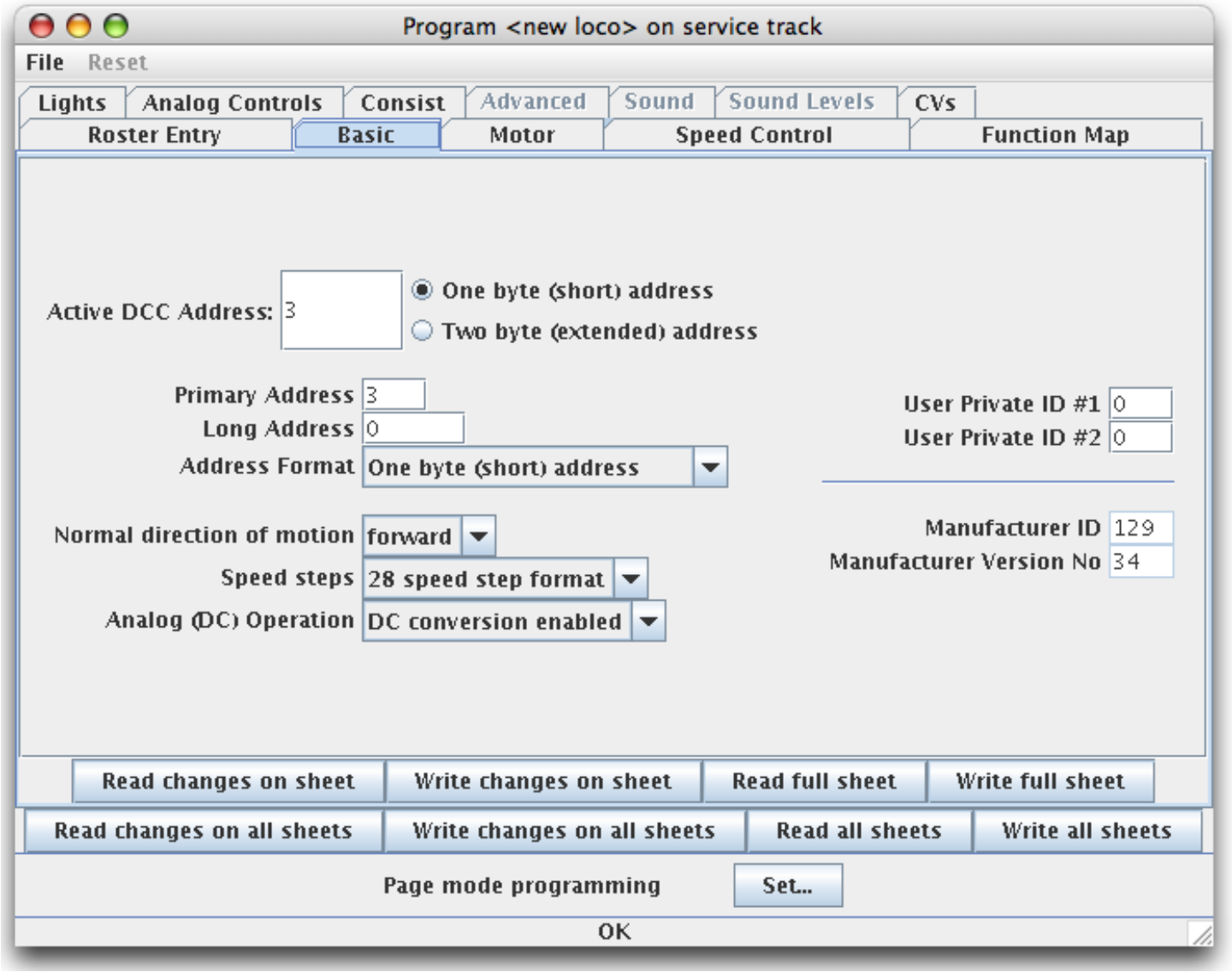
Fill in Roster information and click “Save”.



Click the Basic tab.

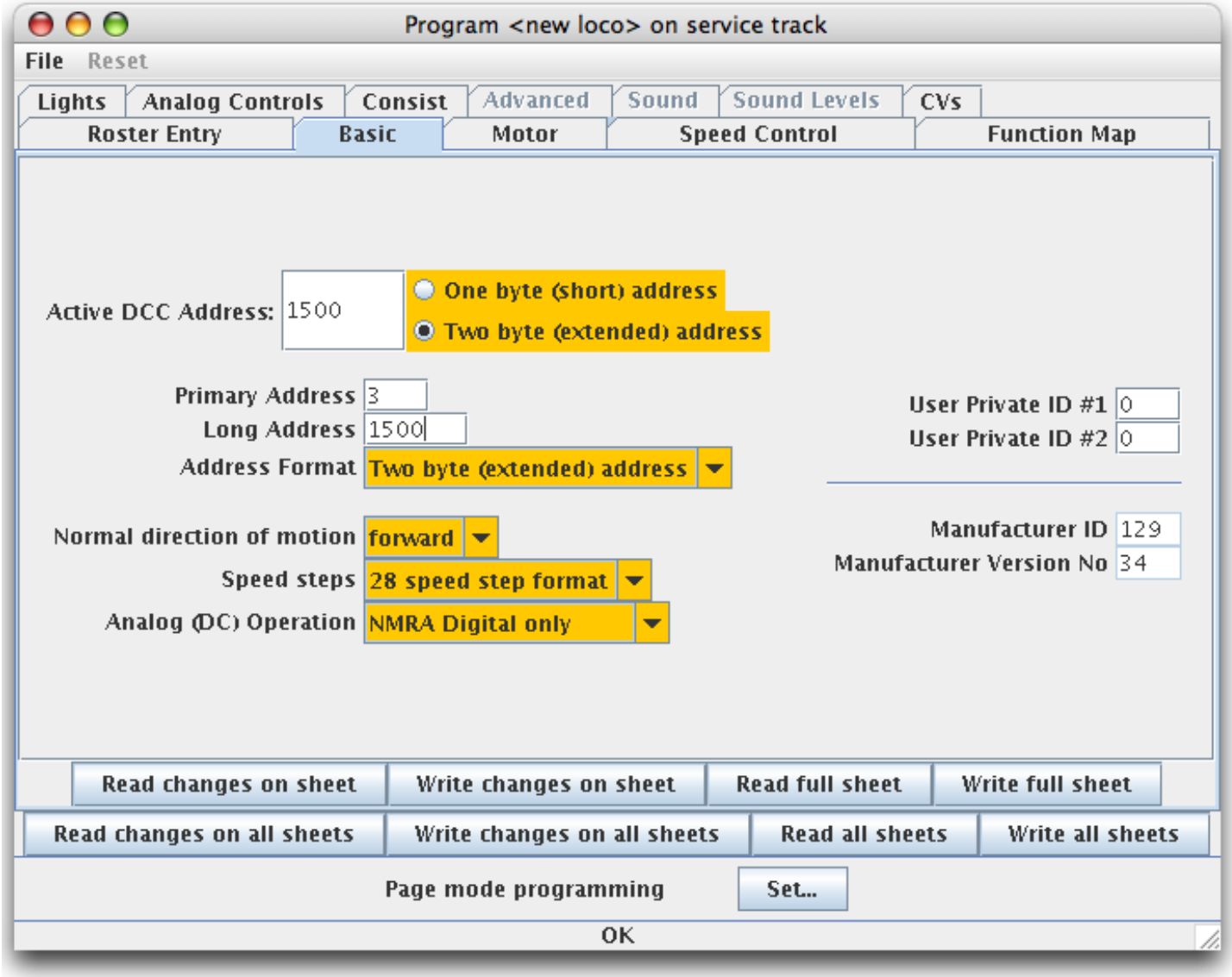


Click “Read full sheet”. Yellow items are replaced with factory default values.



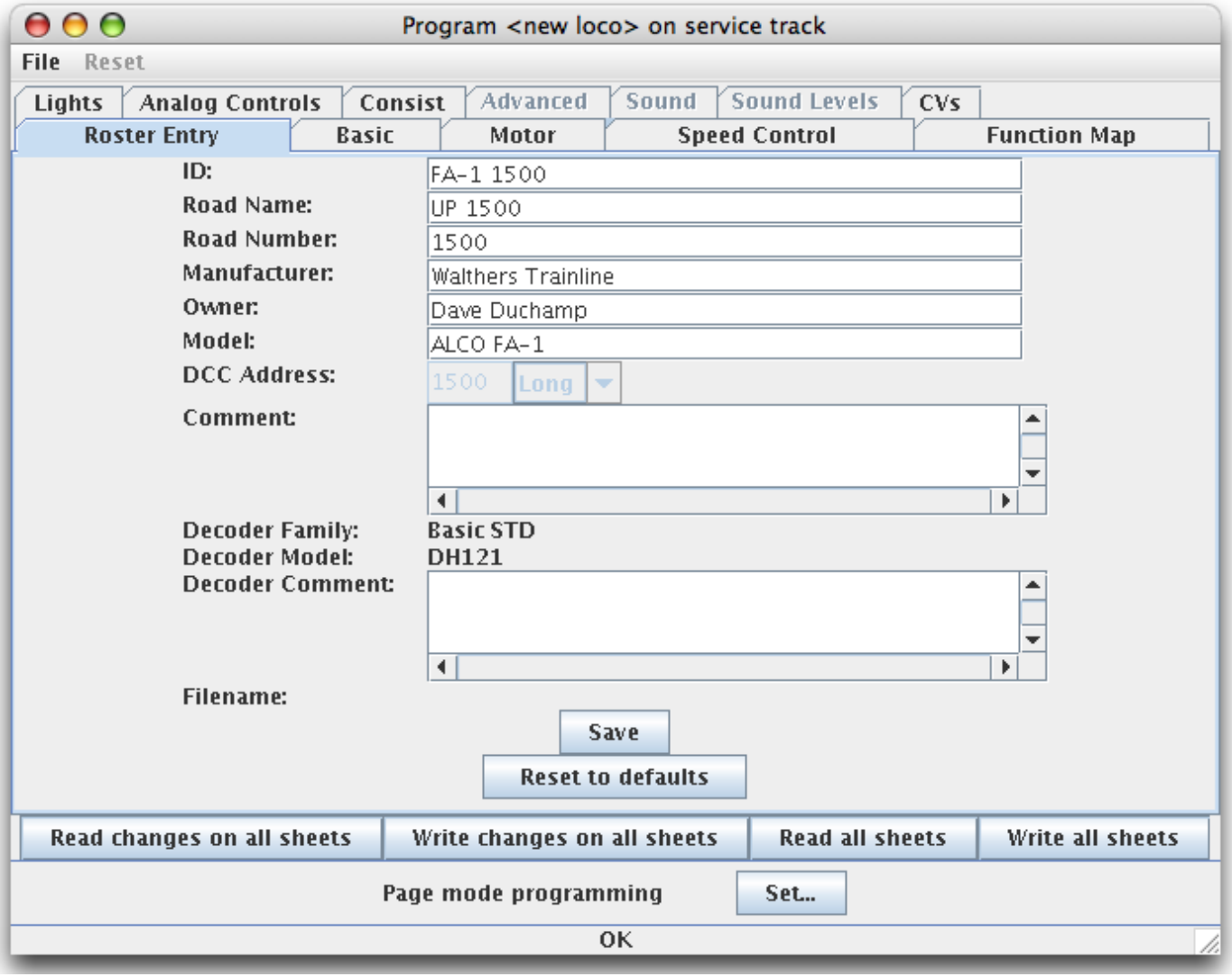
Switch off analog, and set new two-byte address.

Click “Write changes on sheet” to send to loco.



Return to Roster Entry and “Save” the Roster file to disk.

All done!



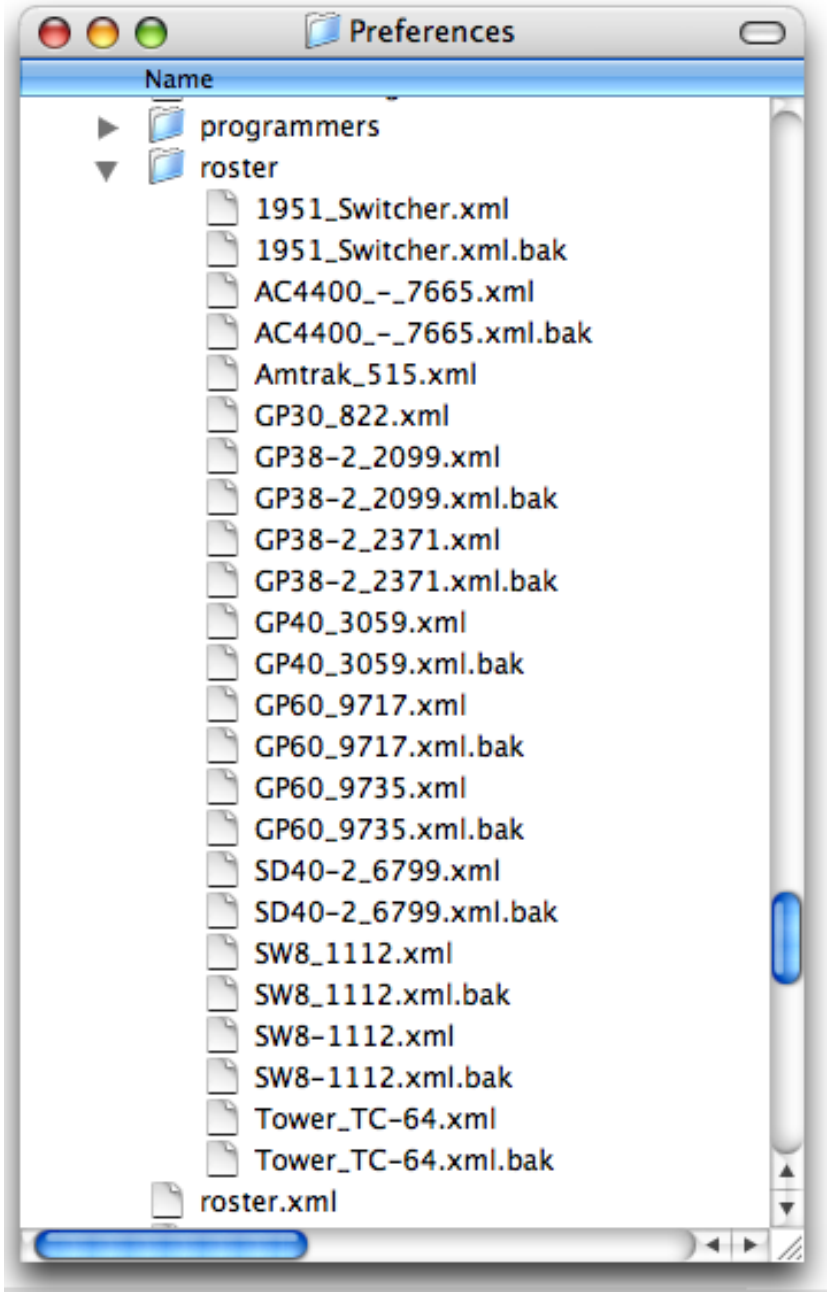
What are Roster Files?

DecoderPro stores the final information for each decoder in a **Roster File**.

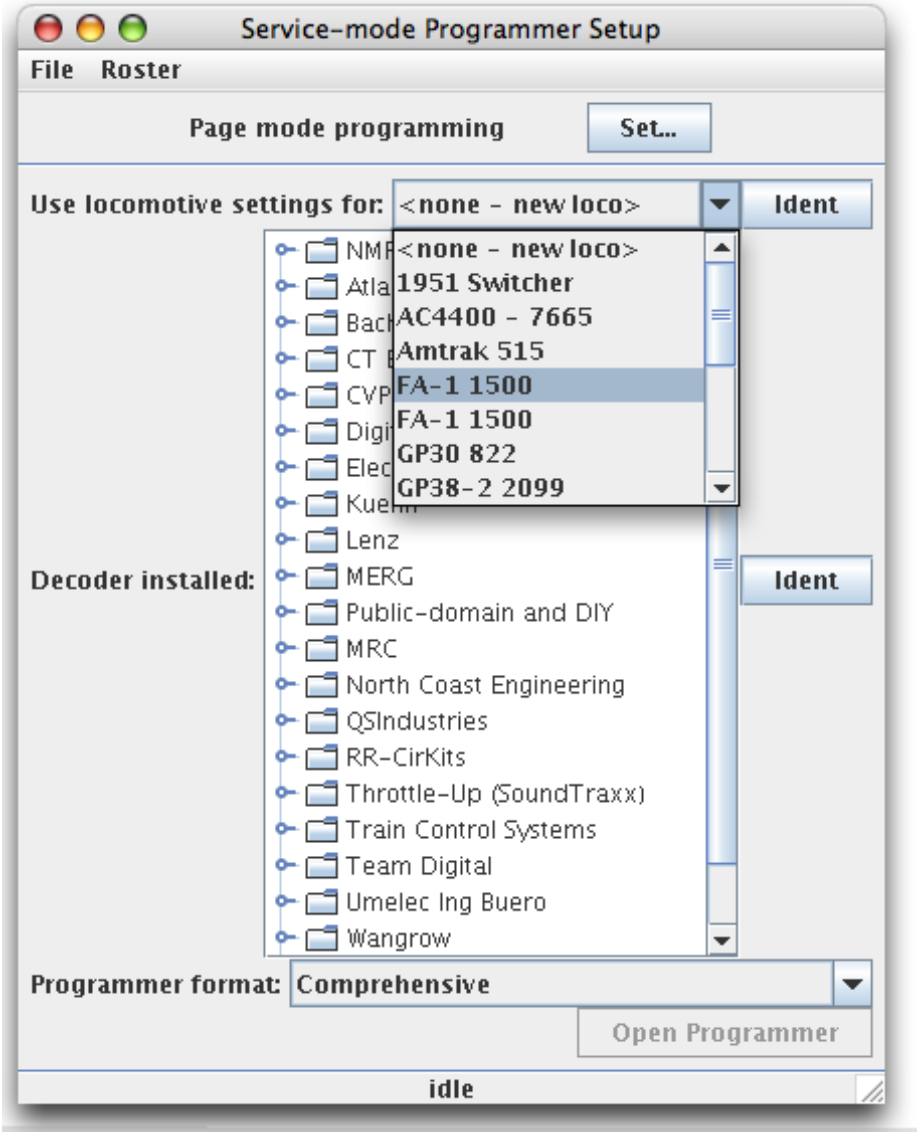
These Roster Files are used to construct a Roster for JMRI applications.

A Roster file allows easy reprogramming if decoder needs to be reset.

The Roster allows easy selection of a loco in JMRI tools-- decoder programmer, throttle, consist, etc.



Changing a decoder's programming



Select loco from Roster

- or -

<- Click "Ident" to have DecoderPro read the loco address and find it in the Roster.

After loco is identified, click "Open Programmer"

Note: "Open Programmer" is not active until a decoder is identified.

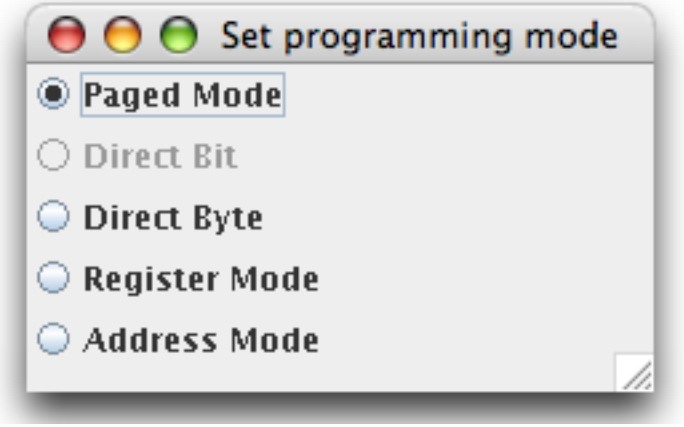
<-

Miscellaneous Info and Tips

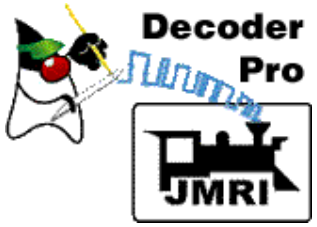
Support for new decoders is constantly being added to DecoderPro.

DecoderPro works through the command station, so it's usually limited to what you can do with your throttle.

DecoderPro supports other modes of programming. Access these other modes using the "Set…" button to get the dialog shown at the right. **Some decoders need a different mode for programming.**



Some new sound decoders need a programming track booster to communicate with some command stations.



Practical Examples

- Practical = Low Cost
  = Time Saver
- Automation can be quite costly and may not be practical for some layouts and owners
- Programming Decoders
  - Speed Matching Locomotives (Demo)
  - Complicated Sound Decoders (Demo)
  - Backup/Restore Decoder Settings (Demo)

Example Procedure for Speed Matching Engines for Consists

Object: To match the speed of two or more engines.

- Preliminary: Determine which engine runs slowest. Warm up engines (3-4 minutes). Make sure wheels and track are clean!
- Make sure all engines have DecoderPro roster files, and start speeds are matched.
- Make a consist with your slowest engine as the lead engine. Do not couple the engines.

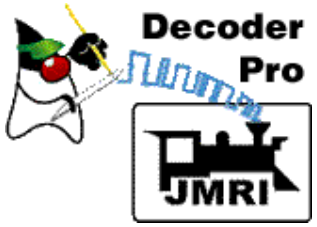
Example Procedure for Speed Matching Engines for Consists

Object: To match the speed of two or more engines.

- In Ops Mode (main track) Programming open the sheets for the slowest engine
- Option 1: Adjust min., mid., and max to get similar speeds
- Option 2: Adjust speed tables to get similar speeds

Programming Complicated Sound Decoders

- Simple sliders for volume controls
- Simple check boxes, drop downs, and radio buttons for other options
- Use Ops Mode (main line) programming for quick testing of new settings



Backup/Restore of Decoder Settings

- Get an engine running again after loss of decoder settings
- Save time during an operating session
- Restore complex settings in no time

Other Uses for JMRI Software

- On-screen throttle (not portable)
- Universal Wireless Throttles
  - WiThrottle – allows use of Android phone, iPhone, iPod, iPad as a throttle on any system
  - Web Interface – allows use of any WIFI device with a web browser such as Blackberry and other cell phones on any system
- Other Throttles
  - RailDriver – a desktop cab throttle designed to look like the controls in a real engine

Other Uses for JMRI Software

- Control Panels (video)
- Dispatcher Panels (video - video)
- Automated Train Control
- Control of Layout Lighting
- Fast Clock
- “Snooping” DCC packets to troubleshooting
- And a whole lot more…

Decoders can be programmed **without having to know anything about individual CV's.**

Isn't that a great way to program DCC decoders!