



Automotive Communications API and Library

In-Vehicle Data

Gathering data is an important part of modern vehicle development, manufacturing, testing, service, fleet management, and motorsports. Today's vehicles rely on accurate factory sensors to manage the powertrain, ABS, steering, suspension, and airbag systems. These sensors communicate on the in-vehicle network, and the signals can be accessed externally in a scaled format with the proper hardware and software.

Accessing in-vehicle network data requires hardware capable of communicating over the protocols used by that vehicle, and software that has address locations for manufacturer-specific parameters and scaling information.

What is JVCI?

Drew Technologies developed JVCI, a software API and data Library, to get data off the in-vehicle network and deliver it to 3rd party developers in a simple and easy-to-use format. JVCI works with all J2534 hardware from Drew Technologies. JVCI automatically detects the protocols in use on the vehicle, determines what parameters are available for data acquisition, and delivers the data thru the API in a scaled format.

Supported JVCI Parameters

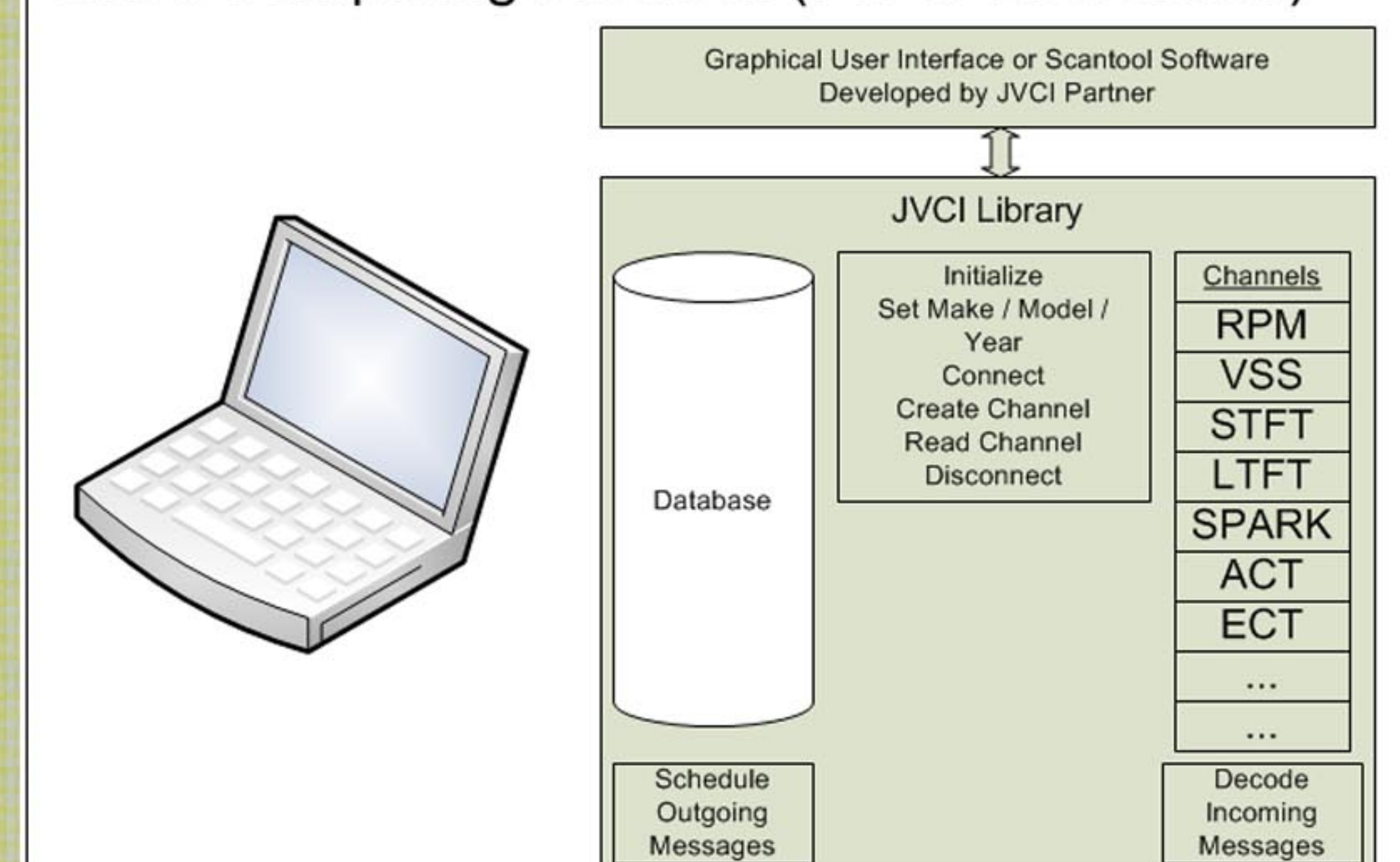
JVCI supports generic OBD on any vehicle 1996 or newer. Enhanced parameters are currently available for Ford Motor Company SCP/CAN vehicles and General Motors GMLAN vehicles. Planned future releases will extend enhanced parameter support to GM Class2, Chrysler, Toyota, Nissan, Honda, Mercedes, Subaru, and Mitsubishi.

Implementing JVCI

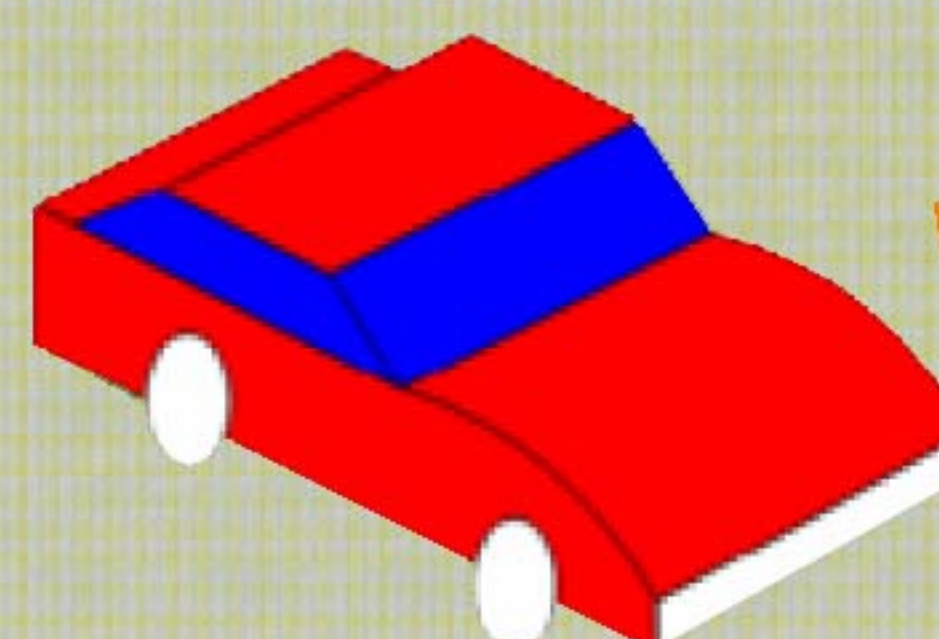
The library is provided as a C language DLL with an integrated database of enhanced parameters. JVCI requires experience with C programming, or experience using a C DLL in some other language. It's designed for a programmer that does not want to program the CarDAQ directly, does not need to learn how to talk to an ECU, or build a database of parameters.

JVCI may also work with C#, VB, LabVIEW, Java, and other programming languages. Consult with Drew Technologies for more information. It will work on Windows, Linux, OSX, and other standardized platforms.

Client Computing Platform (PC or Embedded)



In-Vehicle Network



Vehicle Interface



How Does it Work

The JVCI library encapsulates all of the different protocols with a simple software interface. The typical steps are:

1. JVCI identifies the vehicle automatically, or prompts the user to select make/model/year manually
2. JVCI automatically connects and sends the commands to enter the controller's diagnostic mode
3. JVCI queries the database for a list of supported enhanced parameters.
4. The user is prompted to choose any number of parameters from a supported list. Each parameter will get a separate buffer for scaled, time stamped data.
5. The Acquisition starts. The library schedules outgoing requests using high/low speed techniques as necessary to meet timing requirements.