Getting Started Developing with the JSR Common API

Updated 2/16/2007
Version 1.1

1. Introduction

Thank you for choosing JSR Ultrasonics products for meeting your production, testing, or research needs. Using the JSR Common SDK will allow you to quickly and easily build your own GUI or text interface to several JSR Ultrasonics products or completely hide the instrument interface in a complex application controlling several devices and instruments. Your application can display or hide features as you need. The SDK allows developers to interface all supported JSR Ultrasonics devices via a standard interface, with optional features available on the more advanced instruments.

The SDK provides a Simulation mode which allows you to develop application code before you purchase or connect to actual instruments.

To use this development kit, you will need:

- Microsoft Visual C++ 6.0
- A PC with Microsoft Windows 2000 or Windows XP
  (As of this release, the SDK has not been completely tested on Vista)
- The CD or Zip file containing the SampleWorkspace and accompanying tools

Other documents that are part of the JSR Common SDK:

- JSR_API_ProgrammersReferenceManual.pdf
- JSR_API_PropertiesReference.pdf
- JSR_API_FAQ.pdf Frequently asked questions.
- JSR_API_ReleaseNotes.pdf

2. Installation of the Development Environment

These are step-by-step instructions for installing the SampleWorkspace for development/testing the JSR Common SDK

1. If you are using a PRC50 Instrument, run the JSR Control Panel Installer (JSRControlPanelInstaller.exe). Since the PRC50 is a PCI device, before operating it you must install the PCI drivers for the device.

2. Unzip the SampleWorkspace project in the directory of your choice.

3. Contents of SampleWorkspace

The SampleWorkspace developer's package once installed and unzipped should contain the folders described below.

Bin

- This folder contains precompiled example projects, as well as the JSR_Common.lib and JSR_Common.dll files. NOTE: All of the VC++ projects in the package compile their .obj and .exe files in separate folders and then copy the .exe into this folder.

Docs

- This folder contains reference documentation, and information on how to develop software with the SDK.

Include

- This folder includes the header (.h) files you need to include in your project to use the JSR Common API data structures, typedef's, and enumerations.

Loader:

- This folder includes an example of how to load the JSR_Common.dll at run-time. It is used in some sample code projects, such as SampleCode02.

SampleCode01

- This folder contains a very simple command line executable that uses the JSR Common API, as well as separate VC++ projects, for building the program specifically for PRC50 and the DPR500 instruments.

SampleCode02

- This folder contains a basic command line executable that uses the JSR Common API, as well as separate VC++ projects, for building the program specifically for PRC50 and the DPR500 instruments.

SampleCode03

- This folder contains an example of using JRS_Status error codes, with developer-defined application error codes.

SampleCode04

- This folder contains an example of formatting output using listID's with associated properties. This also gives an example of using the ID JSR_ID_PulserEnergyIndex (which is a classic use of listID).

SampleWorkspace.dsw

- This is a Microsoft Visual C++ workspace definition containing all of the various builds and projects that are included in the zip file.

4. Starting Development

This is the suggested order of doing things, to avoid errors or problems in development.

1. Run the JSR Control Panel application.

   The JSR Control Panel application, (JSRControlPanel.exe) makes calls to the JSR Common SDK, just as your application would. Running the JSR Control Panel will give you a feel for how the objects and properties work with the JSR Common SDK. Every controllable field on the JSR Control Panel represents a Property in the library.

   The JSR Control Panel can verify drivers are properly installed and instruments properly connected. If the JSR Control Panel can not connect to an instrument, it is very unlikely your application will. Note: only one application is allowed to connect to an instrument at one time.

   The JSR Control Panel can operate in Simulate mode without connection to an actual instrument, just as your application can.

2. Run a sample program from the bin directory.

   Select a sample program to test and run from the bin directory. Programs that start with PRC50 work with PRC50 Instruments only. Programs without an instrument type prefix allow the user to select what type of instrument to use.

3. Open the SampleWorkspace

   Using Visual C++ 6.0, open the SampleWorkspace.dsw in the SampleWorkspace folder.

4. Build and run a SampleCode projects in debug mode

   Executable from the SampleCode will be put in the \bin folder to be in the same directory as the lib and dll files. For your early testing, you should be running programs out of the \bin directory.

5. (Optional) Build and run release versions of SampleCode projects

   Executable from the SampleCode will be put in the \bin folder to be in the same directory as the .lib and .dll files Release versions of all SampleCode executables in the bin folder are write-protected. If/when you choose to overwrite these files, you will need to change the files' read-only attribute.

6. Contact Information

If after reading this and other documentation you still have questions or problems, the JSR Common SDK development team can be reached at:

TechSupport@Imaginant.com