This chapter gives you an overview of CVI, including some of its prominent features and the operating environment. The main emphasis of this chapter is to show you some of the real-world applications presently being used in industry and academia and to explain the purpose and usefulness of CVI. You are shown how to access and run the first CVI project (application) to enable you to get familiar with the look and feel of CVI.
What Is CVI?

Before you get started, you would probably like to know what exactly CVI is and why you need to use it. CVI (C for Virtual Instruments) is a programming environment in which you create applications via a user-friendly graphical user interface (GUI) editor to interactively design virtual instruments. You do not use complicated graphics calls to build your user interface in CVI. You control your application using the full functionality of ANSI C and library functions.

Using the graphical user interface editor, you can select from a wide variety of controls designed specifically for instrumentation: knobs, meters, light emitting diodes (LEDs), gauges, dials, graphs, strip charts, and many more. These controls can be refined to customize their appearance and operation for your particular application.

In addition to the instrumentation controls, CVI lets you add menus, toolbars, text boxes, list boxes, and various kinds of ring controls. You can add sliders, command buttons, binary switches, check boxes, scroll bars, and decoration boxes to enhance the appearance of your GUI.

The power of CVI does not stop here. There are a myriad of built-in library functions ranging from standard library functions to special purpose libraries, such as data acquisition, General Purpose Interface Bus (GPIB), serial instrument control, Transmission Control Protocol/Internet Protocol (TCP/IP) communication, data analysis, data presentation, and data storage.

CVI was designed to minimize the development effort required to create engineering and scientific instrumentation applications and includes many timesaving features, which are shown below:

- **Function Panel Concept**—Function panels enable you to select the correct number, order, type, and attributes of arguments in the library function templates. The function panel lets you select from among the many attributes available for the function arguments to speed your application development. The library functions can be inserted into the source code via the function panels by just pressing a key.

- **Automatic Skeleton Code Generation**—This feature is a time saver as it creates the “skeleton code” in C from your GUI. “Skeleton code” is a program boiler-plate created from the GUI with the structure of the main and callback functions using the proper syntax.

- **Tight Integration Between Compiler and Linker**—CVI automatically inserts header files required for its internal libraries.
Run-Time Error Checking—CVI prevents you from overwriting memory locations and catches errors that other environments may not.

Interactive Execution of Library Functions—Function panels and code fragments can be executed interactively before including these functions in the source code.

Compatibility and Portability—CVI supports ANSI C and is available on several platforms. All CVI applications are upward compatible to newer releases of this tool. Code developed on Windows (Win32) environments can be accessed from other Win32 development environments.

Open System Architecture—CVI supports open software architecture, enabling you to reuse your existing programs within the CVI environment. You can incorporate standard ANSI C source code, object files, and dynamic link libraries (DLLs) with CVI. You can use the full power of CVI instrumentation libraries and GUI creation tools in a general purpose compiler of your choice from Microsoft, Borland, WATCOM or Symantec.

Add-On Enhancements—Instead of building your own applications from scratch, CVI has a number of add-on packages to perform various specialized functions that suit your needs. These packages include Test Executive, TestStand, SQL Toolkit, Statistical Process Control, Image Processing, PID (proportional-integral-derivative) Control, and many more.

Using the strength of the CVI environment, you can markedly increase your productivity. Programs that took weeks or months to write using the conventional programming languages can be completed in hours using CVI because it is specifically designed to take measurements, analyze data, and display the results in a user-preferred format.

Where Is CVI Used?

The convenience of design and ease of use has made CVI the product of choice in many industries, such as the automobile, aerospace, defense, energy, telecommunications, insurance, chemical, and test equipment industries, as well as many more.

The automobile industry uses CVI to test the Cadillac DeVille and DeVille Concours models for premium sound quality. The system used is called the
Automotive Audio System Tester (AAST), which conducts all the testing, data analysis, and storage of data for later use. The AAST includes microphones, a signal generator, a PC with CVI instrumentation software, an AT-DSP2200 board, and an AT-GPIB instrument controller board. This system can test operation and sound quality of eleven power amplifiers and eleven speakers.

The aerospace industry is using CVI to detect intermittent faults of 757 aircraft flap/slat controls during thermal and vibration cycling. Commercial Avionics Systems—Irving Operations (CAS-IO), a division of Boeing Defense and Space Group, has designed an automatic test system called the Boeing Mature Programs Environmental Stress Screening (ESS) Monitor. The ESS Monitor is a PC-based automatic test system using VXI and GPIB instrumentation, including the National Instruments VXIpc-486™ Model 550 embedded controller and CVI development software. The test runs constantly initiating the internal Built In Test Equipment (BITE) of a 757 Flap/Slat electronics module while subjecting it to thermal and vibration cycling. All the necessary input signals are provided, the BITE is activated, and the test results read. The module data is read back in a standard eight-bit (ASCII) format over the RS-232 serial bus. There are plans to develop the tests for 737, 747, 757 and 767 modules.

CVI is used in many facets of the defense industry and national security organizations. In particular, it is used successfully in the testing of scaled models of aircraft, missile, and space systems. These models are tested in conditions from vacuum pressure to many fold pressures and from low speeds to speeds up to Mach 6.

The U.S. Department of Energy uses CVI to investigate and map the conditions of the outer walls of the storage tanks containing radioactive waste materials using DAQ boards to control a remote robotics systems. These containers, which are 30 ft. tall and 80 ft. in diameter, are buried underground in a four foot layer of concrete. The only access to the outside of these underground tanks is through a very narrow space where the pipes feed the tanks. CVI was able to easily integrate the computer systems and create a GUI with operator interface menus that duplicate the indicators and controls of the instrument. Other control systems were used previously, but CVI proved to be superior for performing motion control.

In the world of developers of test systems, True Diagnostic Systems (TDS) uses CVI to test high-density matrix switching systems. These test systems use a mixture of analog and digital systems as well as GPIB and VXI instruments. Using CVI as a programming language for TDS test systems has made a major improvement in its product line.
At the Radar and Counter Measures (RCM) division of Thomson-CSF in France, CVI is being used for testing and controlling the performance characterization of microwave transmitter/receiver modules, which are part of active electronic-scanning antennas used for airborne radar. The software controls the GPIB instruments and VXI-bus instruments and is used to develop increasingly complex units under test designs. The decision to use CVI was driven by the quality of GPIB and VXI instrument drivers available in the CVI Instrument Driver Library, its wide range of GUI Library functions, its mathematical and analysis functions, and its basis in industry-standard programming languages. C programming language remains the choice of test instrument manufacturers because it easily controls the hardware in use.

In the telecommunications industry, Alcatel used CVI to find the best possible position to build a new radio network antenna for clear signals in that area. CVI was used to solve this problem. With the help of the Global Position System (GPS), a spectrum analyzer, and a laptop computer running CVI, the Alcatel engineers drove around the new site and measured the signal strength from a proposed transmitter location.

CVI is used in testing deep-sea diving suits and high-energy heart defibrillators and in performing battle-tank audio analysis. It is also used to reduce the downtime on fiber spinning machines, to test rail-vehicle propulsion logic printed circuit boards, and to teach pharmacology to students in universities as well as for many additional real-world uses.

CVI is not limited to use with instruments only. It can also be used as a stand-alone tool to perform database manipulations, mathematical modeling, advanced analysis, algorithm testing, or whatever you need to achieve programmatically using exotic GUIs.

I hope that by now your interest in learning CVI has been aroused. So let us charge ahead to run your first CVI project!

Starting CVI

The easiest way to run CVI is to double-click on the desktop icon that you created after you installed the software as explained in Appendix A. If you prefer not to create a desktop icon, you can run CVI from the Programs option by selecting the Start button in Windows. See Figure 1–1 for an example of what you might see.

To run the program, select CVI IDE. You will see a copyright screen telling you the version of the CVI software and to whom it is licensed. After
that, you see a screen like the one in Figure 1–2. This is called the Project window.

A Project window is used to open a new or previously created project and to compile, build, and run the project. It also saves the files, sets the various options, creates stand-alone executables and DLLs, and performs debugging. From the Project window, you can setup the CVI environment and numerous other features that you will see as you build the projects in later chapters.

Projects are CVI applications that you build and execute. A project consists of various types of files that will be discussed as we go along. All the projects are built by starting with the CVI Project window.

If you have built and saved any projects previously, the CVI Project window will open with the last project files displayed and the project title on the Title Bar. If you are starting a new project, you will see a blank Project window template with no files listed, as shown in Figure 1–2, with Untitled1.prj shown on the title bar. Untitled1.prj is the default

Figure 1–1
Typical Start Button List under Windows
project name when a new project has not been saved with a user-supplied name.

This new Project window lets you start building the CVI applications.

The First CVI Project

To run your first project, select File>>Open>>Project (*.prj) from the Project window. In the folder on the hard disk where you installed the project files from the CD-ROM, find and double click on project 1-1.prj. Depending on your folder settings option, you may not see the file extension; instead you may see a description of the file. The files associated with this project are listed in Figure 1–3 in the Project window list.

Notice that the pathname of the project is displayed in the Title Bar of the Project window. Throughout the text, your pathnames may be different and will indicate where you installed the project files.

Select Run>>Execute project1-1.exe, and the application will start to run. The GUI for this project, as shown in Figure 1–4, will be displayed.
Before you execute this project, let us discuss the project’s functionalities.

This project is used for monitoring the data from a rocket engine controller. The data monitored consists of voltages from two power supplies (five volts DC (direct current) and fifteen volts DC), open position for four engine valves, and pressures for gas generation and combustion chamber. Other data can be monitored for a rocket engine, but for this demonstration, the above data monitoring will suffice. Data can be acquired through a variety of communication interfaces: a data acquisition card, GPIB, RS-232, or other available instrumentation. For this demonstration, the simulated data is read from an ASCII file and run to demonstrate some of the capabilities of CVI.

The top sub-panel labeled POWER SUPPLY in Figure 1–4 records the data on a strip-chart for either the five-volt or the fifteen-volt power supply. Select which power supply to monitor by checking either the 5 Vdc or the 15 Vdc supply box. You can change the lower and upper limits of the voltages by clicking on the increment/decrement arrows to the left side of these control boxes. Move the mouse over the PS Timer knob (PS is short for Power Supply). Hold down the left mouse button over the indicator line on the knob and move it to set the time interval at which you want the data displayed. The same can be accomplished by clicking on the increment/decrement arrows on the left side of the box that is located below the timer knob.

To start monitoring, click the red ON/OFF command button on this sub-panel. The command button will be pushed in and its color will change to
green. The data will be plotted on the strip-chart at the time interval that you selected on the PS Timer control. A snapshot of the run is shown in Figure 1–5. The values for the lower and upper limits are also plotted on the same strip-chart. Click the ON/OFF command button again. It will pop up to OFF position, its color will become gray, and the data plotting will be stopped. Now change the time interval on the PS Timer to 0 to make it run at the fastest speed allowed by your system (1 millisecond for Windows). Click the ON/OFF command button again and notice the increase in speed with which the data is displayed on the meter and plotted on the strip-chart.

Notice that when the 15 Vdc supply is selected, the meter’s display maximum limit changes from 6 to 17. The maximum scale on the Y-axis of the voltage strip-chart also changes accordingly. The lower and upper limits default values are also changed to 8.0 and 14.25 for the 15 Vdc power supply. The label on the strip-chart changes to 15 Vdc PS to indicate the power supply voltages being plotted. These labels and values are changed programmatically.
To clear the strip-chart, click the **Clear** button below the strip-chart.

The middle sub-panel is labeled **ENGINE VALVES**. From this sub-panel, you can monitor the engine valve data and display it on the strip chart. Click on the pull-down menu shown below the valve gauge, and you will have an option to select one of the four engine valves. The selected valve name is placed in the strip chart title. Set the time interval on the **Valve Timer** knob and click the **ON/OFF** button to run the selected engine valve data. Notice that this **ON/OFF** command button is disabled until the engine valve is selected from the pull-down menu. The valve gauge will also move as each data value indicating the percentage open position of the valve is generated. The strip chart is plotted as before. As with the **POWER SUPPLY**, you can stop the monitoring and change the timer interval to make the monitoring run slower or faster. You can select a different valve from the pull-down menu and display and observe its simulated data.
The lower sub-panel is labeled **PRESSURE CHAMBERS**. In this sub-panel, the two pressure chambers are monitored and their data displayed simultaneously on the strip chart by clicking on the **ON/OFF** command button. The data is also “written” to the pressure tank controls to graphically depict the changes for each pressure value. The **Pressure Timer** knob can be varied appropriately to vary the display speed of the data.

To exit the project, click on the **Exit!** command on the menu, which is located on the top left corner of the menu bar.

**Summary**

This chapter introduced you to **CVI** and its many uses in real-world applications. It got you started with **CVI** and showed you how to run your first **CVI** project. A data acquisition application was run and its many features explained. The purpose of the first project was to show you some of the capabilities of **CVI** and enable you to get familiar with the way this tool can be used for your own needs.