DataSocket$^{\text{ä}}$ Simplifies Live Data Transfer for LabVIEW$^{\text{ä}}$

Many typical instrumentation solutions involve a single local application for acquisition, logging, analysis, and presentation. However, because of the rise in popularity of the Internet and company intranets, and the need to remotely monitor and control your data acquisition, you often need to exchange live data with other applications on different computers around the world. DataSocket for LabVIEW simplifies live data exchange between different applications on one computer or between computers connected through a network.

Although a variety of different technologies exist today to share data between applications, including TCP/IP and dynamic data exchange (DDE), most of these tools are not targeted for live data transfer to multiple clients. With TCP/IP, you have to convert your data into an unstructured stream of bytes in the broadcasting application and then parse the stream of bytes back into its original format in subscribing applications. DataSocket, however, simplifies live data transfer. It implements an easy-to-use, high-performance programming interface that is designed specifically for sharing and publishing live data in measurement and automation applications. In addition, DataSocket features interapplication connectivity, rich data types, and security to make sharing data easy. DataSocket is included with LabVIEW.

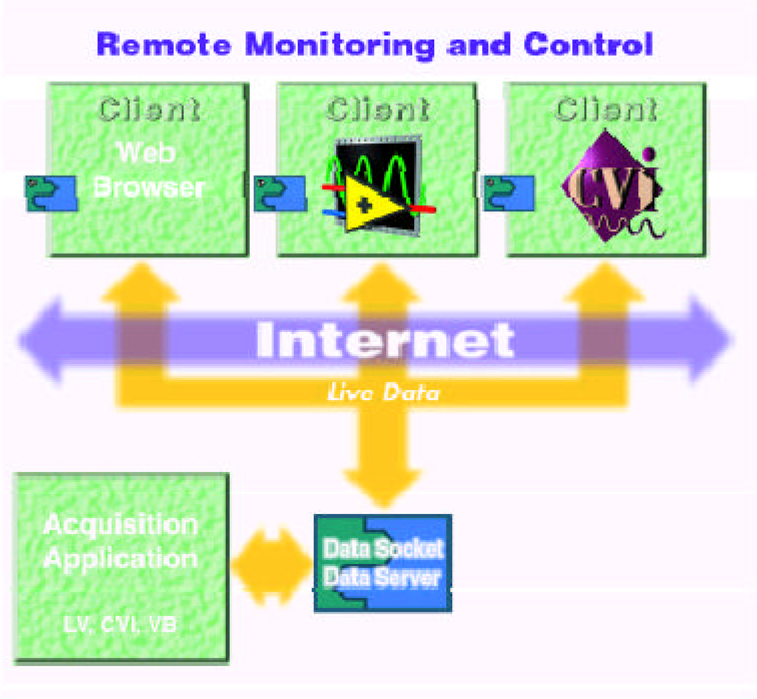


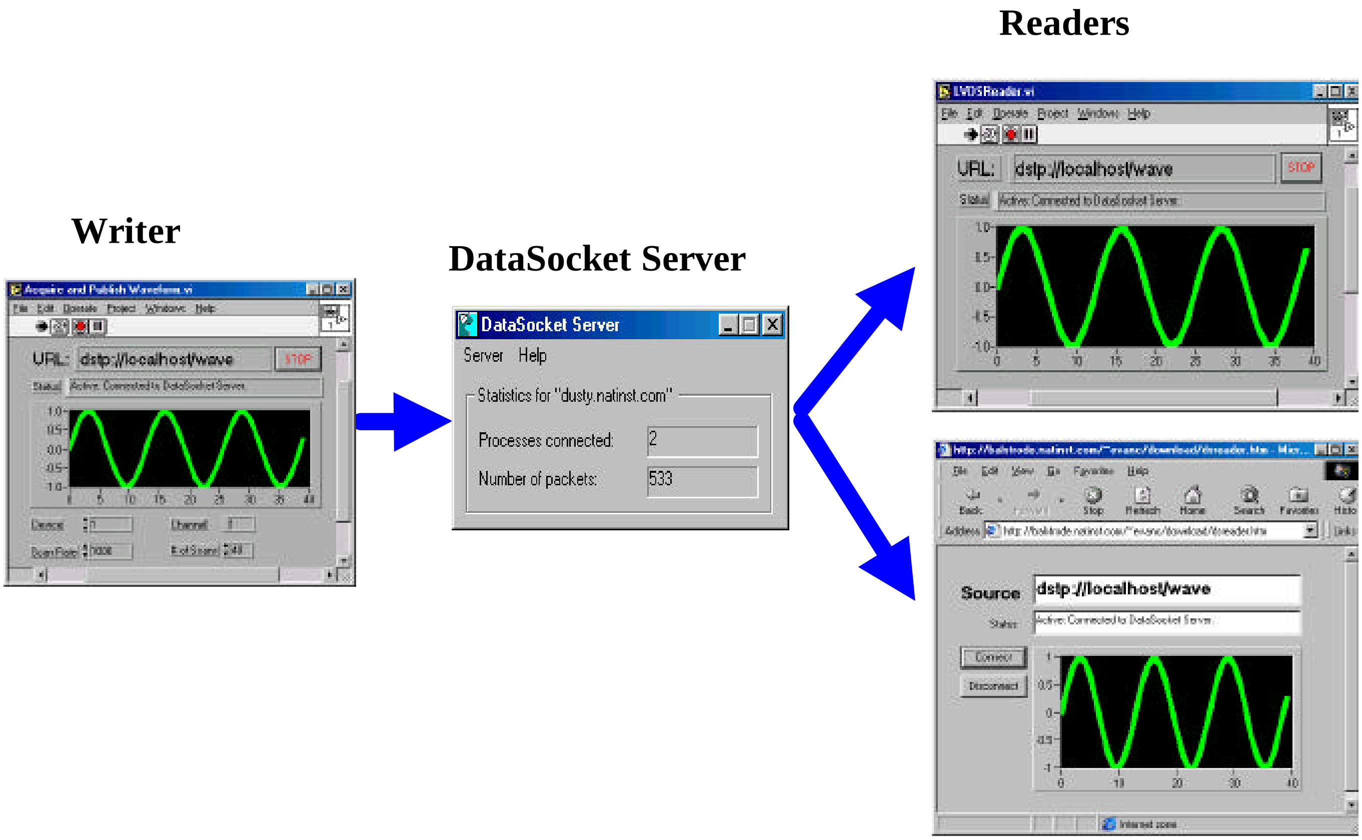
Figure 1: DataSocket shares data between different applications across the network using an intermediary server.

DataSocket Architecture

DataSocket is a single, unified, end-user application programming interface (API) for connecting to data from a number of sources – local files, files on FTP or Web servers, and data items on OPC Servers. A DataSocket application specifies the data location by using a familiar networking standard, the URL. Just as a Web browser uses a URL to connect to a Web page, a DataSocket application uses a URL to connect to data. By using an industry-standard URL, you can quickly and easily bring data into or share data from your DataSocket applications. In addition, the DataSocket Transfer Protocol connects a DataSocket application to live data by specifying a connection to a DataSocket Server. The DataSocket Server manages most of the networking tasks for you.

DataSocket Server

With the DataSocket Server, a lightweight, stand-alone component, programs using DataSocket can broadcast live measurement data at high rates across the Internet to multiple remote clients concurrently. These client applications use DataSocket to subscribe to the live measurement data. Because the DataSocket Server is a stand-alone component, it simplifies network (TCP/IP) programming by automatically managing connections to clients and automatically converting your measurement data to and from the stream of bytes sent across the network. You do not have to write the parsing code. And because the DataSocket Server can run on any machine on your network, it also improves performance and provides security by isolating the Web connections from your acquisition application.



URL: dstp://localhost/wave

Status: Active: Connected to DataSocket Server

Statistics for "dusty.natinst.com"

Processes connected: 2

Number of packets: 533

Source dstp://localhost/wave

Figure2: The DataSocket Server distributes live data to multiple applications, including LabVIEW, LabWindows/CVI, Visual Basic, and Web pages.

Publishing Live Measurement Data

We designed the DataSocket virtual instruments (VIs) for ease of use, modeled after the File VIs in LabVIEW. Because this is a familiar model to most LabVIEW users, you can begin using DataSocket without learning a new way to program. With just four commands, you can enable your LabVIEW application for remote data exchange. Just add the DataSocket Open Connection, DataSocket Write, DataSocket Close Connection, and DataSocket Status VIs to your existing LabVIEW application. Your application is now capable of publishing its data. To learn more about publishing your live measurement data, let’s look at a hypothetical example.

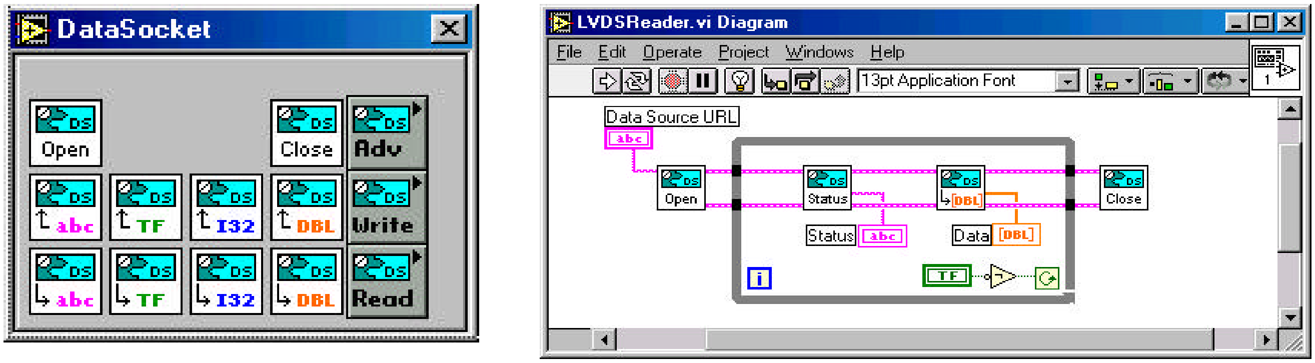
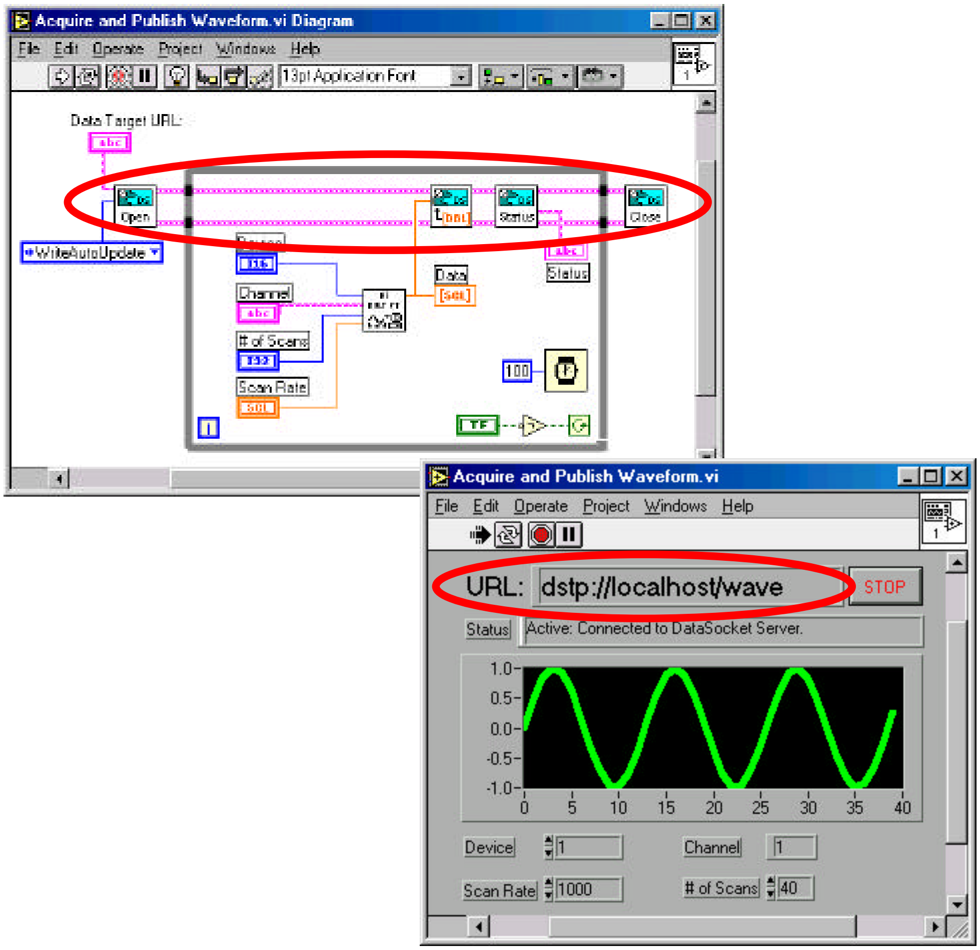


Figure 3. The DataSocket palette includes VIs to Open, Read, Write, and Close. VIs are provided for many common data types. Use DataSocket Open Connection to connect to a data source, DataSocket Read to get the data, and DataSocket Close Connection to disconnect from the data source. You can use the DataSocket Status VI for checking the validity of your connection.

Suppose you have a simple LabVIEW application acquiring a waveform and you want to view the waveform data from a remote machine. Start with the very simple AI Acquire Waveform. Add the DataSocket VIs to publish the data to multiple clients.

The example illustrated here, Acquire and Publish Waveform, publishes an acquired waveform to a DataSocket Server data item, using the following standard URL syntax, “dstp://localhost/wave”. Let’s examine the URL more closely. The “dstp” indicates that the DataSocket application is connecting to a DataSocket Server using the DataSocket Transfer Protocol for live data. The “localhost” indicates that the DataSocket Server is running on the local machine; if the DataSocket Server were running on another machine, you would replace localhost with the machine name or IP address. The “wave” is the data item name on the server. This is an arbitrary name, much like a file name or Web address, which identifies the location of your data on the DataSocket Server. A single DataSocket Server can handle numerous data items.



URL: dstp://localhost/wave

Status: Active: Connected to DataSocket Server.

Device: 1

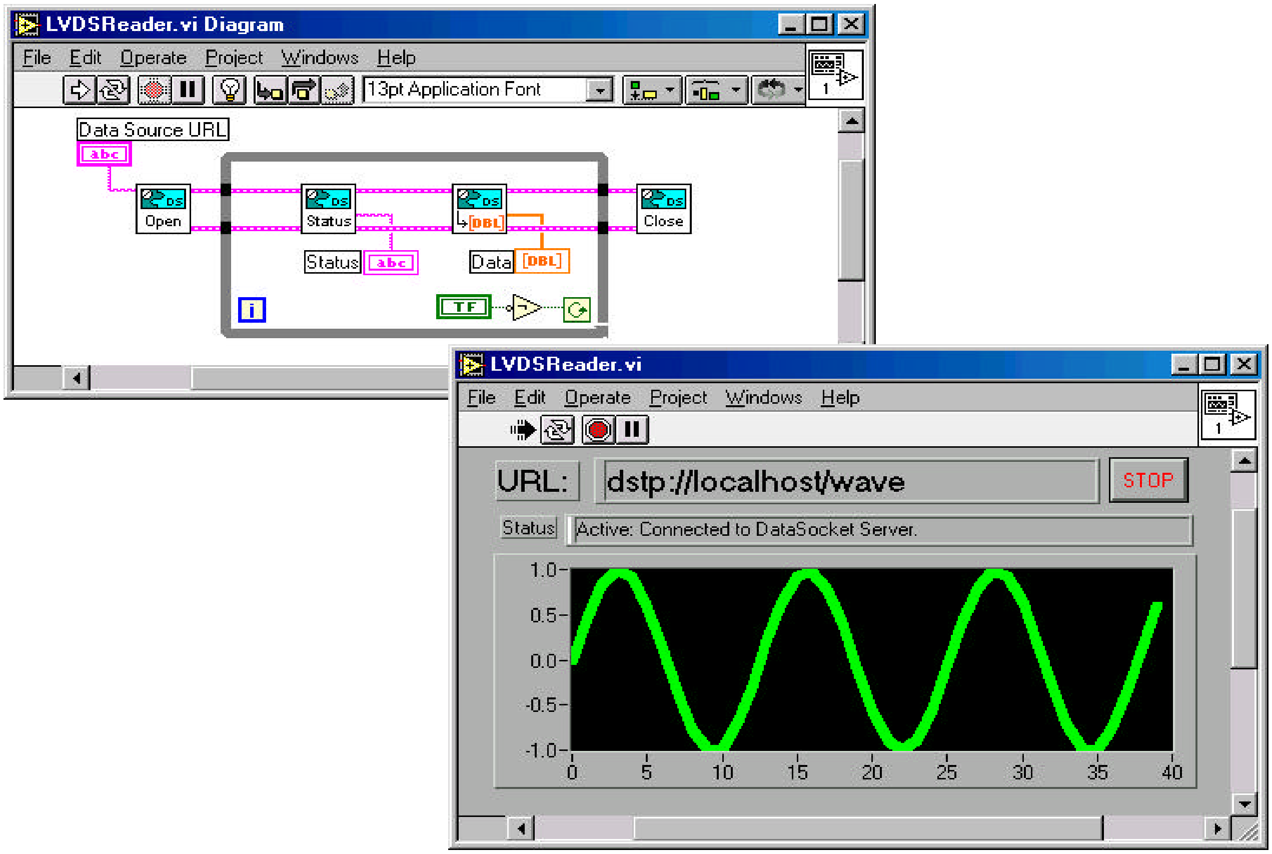
Channel: 1

Scan Rate: 1000

# of Scans: 40

Figure 4. Publishing data can be as simple as wiring an existing signal to a DataSocket write VI.

Once the example has connected to the DataSocket Server, a remote application can then view this live data using the same DataSocket VIs. You can use the LVDSReader.vi example to connect to this same URL, and using the DataSocket Read VI, read and display the data.



URL: dstp://localhost/wave

Status: Active: Connected to DataSocket Server.

Figure 5. Reading data from a remote server is much like reading from a file.

To run this example, start the DataSocket Server from your Windows Start Menu. Then, run the acquisition application in LabVIEW to begin publishing the data to the server. When you start the reader application, data is immediately displayed. In addition, notice that if you stop the writer, and then restart it, the data is again immediately displayed in the reader. There is no need to restart the reader. The reader applications and the DataSocket Server can run on any machine. Note: make sure that the URL in the reader and the acquisition application refer to the machine on which the DataSocket Server is running, not just "localhost."

Using Attributes to Exchange Rich Data Types

In many cases, your applications need to exchange complex data. In addition to publishing a waveform, you may want to publish the waveform's timestamp, quality, or sample rate. DataSocket accomplishes this with data attributes. Attributes are bound to the data so no extra connection is needed. However, attributes do not affect the fundamental type of the data. Therefore, new attributes can be added without breaking the existing application. The LVDSWriterWithAttributes and LVDSReaderWithAttributes examples included with LabVIEW illustrate how to use attributes to exchange rich data types.

Interapplication Connectivity

Because DataSocket is a technology that is implemented as a National Instruments ComponentWorks™ ActiveX control and as a LabWindows™/CVI library, your LabWindows/CVI applications, Web pages, Visual Basic applications, and Excel spreadsheets can all exchange live data with your LabVIEW applications, using the same, simple API. So your LabVIEW applications can share live data with applications written in other development environments, and even Web pages! For instructions on how to construct a Web page that displays live data from a LabVIEW application, see the application note on Building an Interactive Web Page with DataSocket, available from the DataSocket Web page at www.natinst.com/datasocket

Security

The DataSocket Server provides security using the DataSocket Server Manager. With the DataSocket Server Manager, you specify which machines have the right to read, write, and create particular data items on a DataSocket Server.

The URL is the Path to Different Data Sources

As mentioned earlier, DataSocket handles connections to various types of data sources. With DataSocket, LabVIEW applications can directly connect to data items on OLE for process control (OPC) servers. Additionally, DataSocket automatically parses many types of files, so you can import data directly into your LabVIEW application. So whether you are reading live waveform test data from another application, reading historical trends from a file on an FTP server, or reading a measurement from a remote OPC Server, such as FieldPoint™, you can use the same simple API and switch between data sources just by changing a simple URL. DataSocket provides a common data connection, regardless of the type of data used or the source of the data.

Conclusion

DataSocket provides a powerful, easy-to-use, high-performance interface for sharing and publishing live data in your measurement applications. For more information on DataSocket, visit the DataSocket Web page at www.natinst.com/datasocket