



DDE Client/ TCP Server System Software

Software v1.0
Documentation v 1.2

J. Kuah and J. Hollar
ATLAS Michigan
26 July, 1999

Overview:

The Server/Client program is designed to operate with a minidag program and the ATLASinterface library to distribute events over TCP/IP to remote clients. In order to avoid the overhead of running a TCP server with multiple connections directly in the minidag program, this program runs independently, with only a single DDE connection to the minidag. The program includes four components:

1. A [DDE \(Dynamic Data Exchange\) client](#).
2. A [TCP \(Transmission Control Protocol\) server](#) for use with LabWindows CVI clients.
3. A [second TCP server](#) for use with UNIX and other non-CVI clients.
4. A [user interface](#) and control routines for [operating](#) the different parts together.

Data is sent by the minidag over the single Windows interprocess DDE connection to the Server/Client program. From there it is sent over TCP/IP to as many as 255 Lab Windows CVI based clients and 255 UNIX or other platform clients. The program is written in C and compiled under LabWindows CVI 4.0.1. The user interface is written using CVICALLBACK functions, and cannot be operated under different software packages without modifications. Since DDE is designed specifically for Microsoft Windows, this program cannot be run on other platforms.

In addition to the Server/Client program, a sample [CVI based TCP client](#) and a [UNIX based TCP client](#) are available.

Operation:

The Server/Client program is operated by a CVI based user interface, which allows independent control of the DDE client and TCP server parts of the program. It also records simple event statistics and connection status for the DDE client and TCP servers.

Any minidag program making use of the ATLASinterface library is automatically registered as a DDE server. The DDE server is therefore subject to the same requirements for use of the ATLASinterface library laid out in [3] and [4]. All that is required in the minidag program code is that the function:

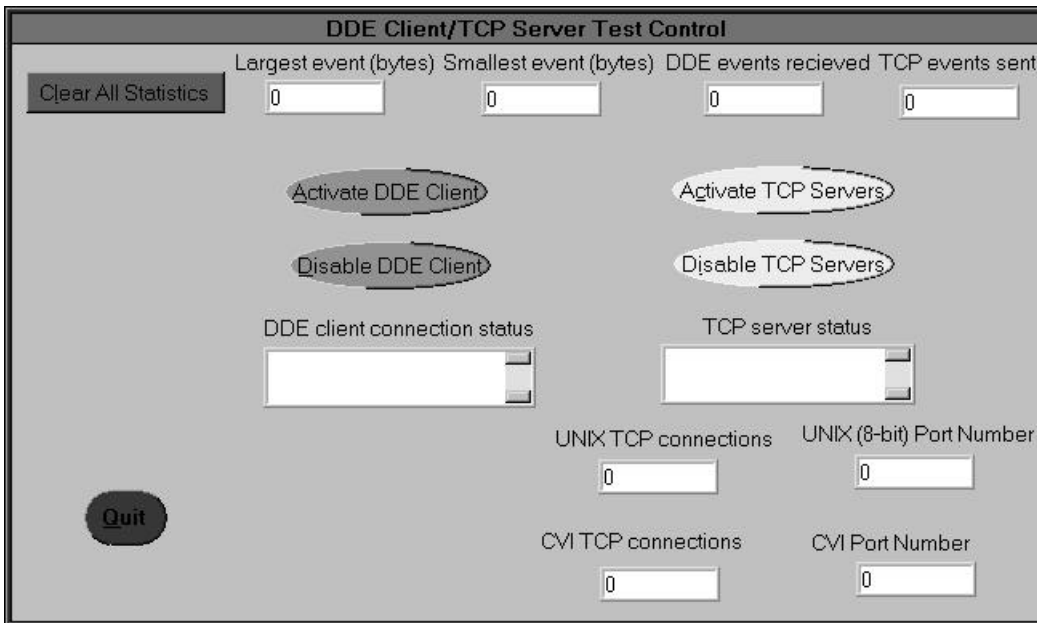
```
SendData(int *data, sizeof(data));
```

where data is a pointer to the new event data, and the second parameter its size in bytes, be called whenever new data is acquired. Calling SendData with the Server/Client program connected will send data over DDE to the Server/Client and then over TCP

automatically. If this function is called when a DDE connection has not been made, the send request will be ignored. The DDE server will automatically shut down when the user quits the minidag program.

DDE works by posting messages to the Windows message queues of the applications involved in the conversation. These queues can become overloaded when frequent DDE transfers are made while multiple other events, such as mouse and user interface commands, are being processed at the same time. When this occurs, DDE data can become lost or corrupted. Any unnecessary processes or mouse events should be avoided when making continuous frequent transfers over DDE.

Server/Client Control Panel



DDE Client:

When the Activate DDE Client button in the Server/Client user interface is enabled, a DDE connection and hotlink will be established with the minidag program. The minidag program must be running before a DDE connection can be established. Due to a problem with passing variable sized data over DDE in Windows, all events are currently sent and received in a 1 kilo -word buffer. The structure of this buffer for each event is assumed to have a 10-word header followed by the actual data:

Header:

1. Total length of this event, including itself
2. Run number
3. Event ID number
4. Reserved 1
5. Reserved 2
6. User defined
7. User defined
8. User defined
9. User defined
10. User defined

Data:

1. Total length of data array, including it self
2. First real data word
3. Second real data word
4. Third

Word one of the data section immediately follows word 10 of the header section. Any unused elements in the buffer will be sent and received as zeros. All words in this structure are 32-bit integers

The DDE client is shutdown by pushing the Disable DDE Client button on the user interface. It will also disconnect automatically if the minidaq program is exited while the DDE connection is still active.

CVI based TCP/IP Server:

This TCP server makes use of LabWindows CVI's ready-made routines for operating a server. Pushing the Activate TCP Server button on the Server/Client user interface activates this server. This must be done before any client programs attempt to connect to it. Only client programs that are using CVI should connect to this server. The appropriate port number will be displayed on the Server/Client user interface. There is an upper limit of 255 simultaneous connections on this server. As soon as the Server/Client program receives data over DDE, it will be sent out over TCP to all currently connected clients.

Data sent by this server is transmitted as a one kilo -word buffer. The same header and data format received over DDE is kept intact as it is sent over TCP . Events can be received by the client as an array of 32 -bit integers.

UNIX based TCP/IP Server:

This TCP server provides a general connection for UNIX or other non-CVI based TCP client programs. The appropriate port number will be displayed on the Server/Client user interface. It is also activated when the Activate TCP Server button in the user interface is pushed. This must be done before any client programs attempt to connect to it. This server also has an upper limit of 255 simultaneous connections. As soon as the

Server/Client program receives new data over DDE, it will be sent over TCP to all currently connected clients.

This connection only allows transfers in 8-bit packets. Therefore, the 32-bit words in the buffer are broken up into four words each containing eight valid bits in the Most Significant Byte position. The actual buffer sent by this TCP server will then be 4 kilo-words long. It will be the responsibility of the UNIX TCP client program to reformat the received data in such a way to extract the appropriate 32-bit words in the header and data.

Both of the TCP servers are shut down by pushing the Disable TCP Servers button on the user interface. If this is done while TCP clients still have active connections, those clients will be automatically disconnected by the server.

CVI based TCP Client:

This sample client is designed for connecting to the CVI based TCP/IP server on the Server/Client program. It includes a user interface with connect and disconnect controls, a connection status box, and keeps a count of events received. The user must enter the IP address of the Server/Client host machine and the correct port number before connecting. The option to save to a file or display in standard I/O (default) should be selected before connecting. This client is compiled in LabWindows CVI 4.0.1 using CVICALLBACK functions for the user interface.

The data is received over TCP/IP as a one kilo-word array of 32-bit integers. The data is printed as a list of hex numbers whether it is sent to a file or to standard I/O. Any zero words after the 6-word event header are ignored. Any modifications to the storage format can be made in the functions `ClientCallback1` and `StorageControl` in the *TCPclnt.c* file.

UNIX based TCP Client:

This sample client is designed for connecting to the UNIX based TCP/IP server on the Server/Client program. The user must know the port number and the IP address or server name of the Server/Client host machine to connect.

Each 8-bit packet received is expected to be in the upper eight bits of a 32-bit word in an integer array. This can be changed to handle the lower eight bits by modifying the function `Sort_Output` in the `tcpclient.c` file. The program takes the valid eight bit data it receives over TCP/IP and reassembles the 32-bit data words before writing them to a file. To run the UNIX TCP client program, type:

```
tcpclient <servername> <port> <filename>
```

References:

- [1] LabWindows/CVI Standard Libraries Reference Manual. July 1996: National Instruments
- [2] LabWindows/CVI User Manual. July 1996: National Instruments
- [3] ATLAS MiniDAQ Interface Library Software, Version 1.3. July 1999: E. Munson and B. Ball
- [4] ATLASinterface Library Configuration Issues. December 1998: E. Munson