EDRE DDE

Eagle Technology
Dynamic Data Exchange
User's Manual

Eagle Technology – Cape Town, South Africa

www.eagle.co.za

Dynamic Data Exchange

Data Acquisition and Process Control

© Eagle Technology
31-35 Hout Street • Cape Town • South Africa
Phone +27 21 423 4943 • Fax +27 21 424 4637
Email eagle@eagle.co.za

TABLE OF CONTENTS

1. Introduction

Network DDE (Dynamic Data Exchange) allows applications to communicate with each other across a network.

A Brief Look at "Standard" DDE
When using DDE, at least two programs are involved. The DDE server program registers itself with the Windows DDE system as a DDE server and it becomes visible to other applications on the system. The program registers an application and topic name that DDE client programs use to refer to the DDE server. The DDE client program starts a conversation with the DDE server using the application and topic names to specify which program it wants to talk to. The Windows DDE system handles the task of routing messages between the applications.

Moving to Network DDE
For the DDE server, network DDE is no different than standard DDE. The DDE server receives and responds to requests from the Windows DDE system and the DDE system handles the task of routing messages between the two machines. Because of this, any program that acts as a DDE server can use network DDE even if you can't modify the program.

For the client program, only the parameters of the DDECLIENT function or equivalent function call need to change. Instead of calling DDECLIENT with the application (service) and topic names, the function is passed the computer name and share name to connect to. You must be able to modify the application and topic name that the client program uses in order to use network DDE.

The share name is the thing that makes network DDE work. In standard DDE, the DDE client connects to a certain application and topic. In network DDE, the application (service) and topic are set up on the server computer as a DDE share. Instead of requesting a connection to the application (service) and topic, the client program requests a connection to the computer and DDE share.

(See Programming Guide for server service and topic Constance)

Applications

Network DDE is a flexible way to transfer information across a network. Because you do not need to modify the DDE server to use network DDE, any DDE server can be used as a network DDE server.

Network DDE also works independently of the network protocols being used. A network DDE application can communicate using any network protocol supported by Windows, including IPX, NetBEUI, and TCP/IP. I have used network DDE without problems across the Internet.

Although network DDE is not the solution for all network communication tasks, it is an effective tool for communicating information between applications on a network. By learning to implement this protocol, you will have another choice of tools to use when a network-programming situation arises.

Contact Details

Below are the contact details of Eagle Technology.

Eagle Technology
PO Box 4376
Cape Town
8000
South Africa
Telephone +27 (021) 423 4943
Fax +27 (021) 424 4637
E-Mail eagle@eagle.co.za
Website http://www.eagledaq.com

2. Programming Guide

Configure DDE client

When creating a client to connect to the Eagle DDE server you will have to configure the DDECLIENT.

DDECLIENT	DDE Share	NETWORK Share
SERVICE	EDRE_DDE_SERVER	\\ccc\NDDE$
TOPIC	EAGLEDAQ	EAGLEDAQ$

The DDE server uses two ITEMs. The one is used for data being 'poked' to the server and the other for data returned from the server.

Parameter	Constants
DDE_SERVICE	EDRE_DDE_SERVER
DDE_TOPIC	EAGLEDAQ
DDE_NETWORK_TOPIC	EAGLEDAQ$
DDE_ITEM_SEND	SERVERSEND
DDE_ITEM_RECEIVE	SERVERRECEIVE

Change the Destination Application to Use Network DDE
The DDE client application needs to be changed to use network DDE. Only the DDECLIENT function call needs to be changed. Instead of using the application and topic names when calling DDECLIENT, you should replace them with the computer and share names like this:

The application parameter need to be replaced with the string '\\ccc\NDDE$' to tell the DDE system which computer to connect to. This string is in Universal Naming Convention format, which specifies network resources as \\computer_name\share_name. In this case, the computer name is ccc and the share name (this is a network share, not a DDE share) is NDDE$. NDDE$ is a reserved share name used by Windows to connect to the DDE system.

The topic parameter has been replaced with the DDE share name, including the trailing dollar sign. This tells the DDE system to connect to the EAGLEDAQ$ share that was created in the previous step. After the connection is made, the server computer uses the EAGLEDAQ$ share in the registry to route the DDE messages to the correct application and topic.

Using DDE

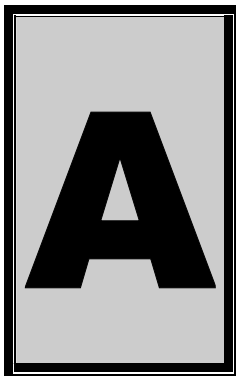
The EDRE DDE Server allows you to do simple data acquisition across networks. Because of speed constraints EDRE DDE Server has only got basic single return parameter functions.

DDE poke date list parameters

Parameter	Description
COMMAND	See command table
SERIAL NUMBER	Hardware Serial number
PORT /QUERY	Hardware port or Query code
VALUE / PARAMETER	Value sent to port or Query parameter

EDRE DDE commands

Commands	Description
“DioWrite”	Write DIO value
“DioRead”	Read DIO value
“ADRead”	Read analog input
“DAWrite”	Write to analog output
“Query”	See Configure Constants
“NumBoards”	Function returns number of boards installed in server computer
“GetBoard”	Function returns serial number of installed board



A.Configuration Constants

Query Codes

Name	Value	Description
APIMAJOR	1	Query EDRE API major version number.
APIMINOR	2	Query EDRE API minor version number.
APIBUILD	3	Query EDRE API build version number.
APIOS	4	Query EDRE API OS type.
APINUMDEV	5	Query number of devices installed.
BRDTYPE	10	Query a board's type.
BRDREV	11	Query a board's revision.
BRDYEAR	12	Query a board's manufactured year.
BRDMONTH	13	Query a board's manufactured month.
BRDDAY	14	Query a board's manufactured day.
BRDSERIALNO	15	Query a board's serial number.
DRVMAJOR	20	Query a driver's major version number.
DRVMINOR	21	Query a driver's minor version number.
DRVBUILD	22	Query a driver's build version number.
ADNUMCHAN	100	Query number of ADC channel.
ADNUMSH	101	Query number of samples-and-hold channels.
ADMAXFREQ	102	Query maximum sampling frequency.
ADBUSY	103	Check if ADC system is busy.
ADFIFOSIZE	104	Get ADC hardware FIFO size.
ADFIFOOVER	105	Check for FIFO overrun condition.
ADBUFFSIZE	106	Check software buffer size.
ADBUFFOVER	107	Check for circular buffer overrun.
ADBUFFALLOC	108	Check if software buffer is allocated.
ADUNREAD	109	Get number of samples available.
ADEXTCLK	110	Get status of external clock line – PCI30FG.
ADEXTTRIG	111	Get status of external trigger line – PCI30FG.
ADBURST	112	Check if burst mode is enabled.
ADRANGE	113	Get ADC range.
DANUMCHAN	200	Query number of DAC channels.
DAMAXFREQ	201	Query maximum DAC output frequency.
DABUSY	202	Check if DAC system is busy.
DAFIFOSZ	203	Get DAC FIFO size.
CTNUM	300	Query number of counter-timer channels.
CTBUSY	301	Check if counter-timer system is busy.
DIONUMPORT	400	Query number of digital I/O ports.
DIOQRYPORT	401	Query a specific port for capabilities.
DIOPORTWIDTH	402	Get a specific port's width.
INTNUMSRC	500	Query number of interrupts sources.
INTSTATUS	501	Queries interrupt system's status.
INTBUSCONNECT	502	Connect interrupt system to bus.
INTISAVAILABLE	503	Check if an interrupt is available.
INTNUMTRIG	504	Check number times interrupted

Error Codes

Name	Value	Description
EDRE_OK	0	Function successfully.
EDRE_FAIL	-1	Function call failed.
EDRE_BAD_FN	-2	Invalid function call.
EDRE_BAD_SN	-3	Invalid serial number.
EDRE_BAD_DEVICE	-4	Invalid device.
EDRE_BAD_OS	-5	Function not supported by operating system.
EDRE_EVENT_FAILED	-6	Wait on event failed.
EDRE_EVENT_TIMEOUT	-7	Event timed out.
EDRE_INT_SET	-8	Interrupt in use.
EDRE_DA_BAD_RANGE	-9	DAC value out of range.
EDRE_AD_BAD_CHANLIST	-10	Channel list size out of range.
EDRE_BAD_FREQUECY	-11	Frequency out of range.
EDRE_BAD_BUFFER_SIZE	-12	Data passed by buffer incorrectly sized
EDRE_BAD_PORT	-13	Port value out of range.
EDRE_BAD_PARAMETER	-14	Invalid parameter value specified.
EDRE_BUSY	-15	System busy.
EDRE_IO_FAIL	-16	IO call failed.
EDRE_BAD_ADGAIN	-17	ADC-gain out of range.
EDRE_BAD_QUERY	-18	Query value not supported.
EDRE_BAD_CHAN	-19	Channel number out of range.
EDRE_BAD_VALUE	-20	Configuration value specified out of range.
EDRE_BAD_CT	-21	Counter-timer channel out of range.
EDRE_BAD_CHANLIST	-22	Channel list invalid.
EDRE_BAD_CONFIG	-23	Configuration invalid.
EDRE_BAD_MODE	-24	Mode not valid.
EDRE_HW_ERROR	-25	Hardware error occurred.
EDRE_HW_BUSY	-26	Hardware busy.
EDRE_BAD_BUFFER	-27	Buffer invalid.
EDRE_REG_ERROR	-28	Registry error occurred.
EDRE_OUT_RES	-29	Out of resources.
EDRE_IO_PENDING	-30	Waiting on I/O completion