



10 Commerce Way
Norton, Massachusetts 02766
Tel: 508.946.5100
Fax: 508.946.9500
info@mccdaq.com
www.mccdaq.com

DASyLab Techniques

DASyLab DDE Examples

Introduction

Working with DASyLab DDE functions requires some knowledge of how DDE (dynamic data exchange) works. This obsolete Windows function allows DASyLab to send and receive data to an external program, and to send and receive commands from an external program.

Microsoft has advised that the DDE technology is obsolete and should not be used. They recommend that programmers move to the Microsoft Automation (OLE) model. DASyLab 2020 supports the ActiveX/OLE automation model in the Script module. Several Excel examples have been published for the Script module using that communications technique.

However, many DASyLab users prefer to continue to use the DDE functions as they are simple and straightforward. This technical note describes several examples for using the DDE functions with DASyLab 2020. Most of these techniques will work with earlier versions, but the examples were created with 2020 and will work with future versions of DASyLab until Microsoft completely removes DDE from the operating system.

The examples that are described in this technical note are:

- [Write to Excel at 1 samples/second](#)
- [Read from Excel sheet at high speed](#)
- [Write to Excel sheet at high speed](#)
- [Send Commands to Excel](#)
- [Send Global Strings and Variables to Excel](#)
- [Start Excel with a filename](#)

All of the examples require Microsoft Excel 2003 or later. They were created and tested with DASyLab 2020 and Excel 1903 32-bit on a Windows 10 64-bit operating system.

For most of the examples the Excel program must be open prior to operating and configuring the examples. Some examples include the Excel sheet. Read the description of the example, open it, and try it. Try making changes to see how the different settings work. When the example uses Global strings or variables, look in the Options menu, Global variables and strings to see the settings, including the Advanced settings.

There is no one location for Excel commands, but the Microsoft support site has an article on starting Excel from a command line. Other references are from the Excel VBA programming references.

These examples are based on the DASyLab DDE functions. Prior to using them, review the DASyLab Help and these articles in the Measurement Computing Knowledge Base, kb.mccdaq.com.

[DDE in DASYLab writing to Excel](#)

[Transferring Text from DASYLab to Microsoft Excel](#)

[Using DASYLab to Read Data from Microsoft Excel](#)

The DASYLab Help includes both the help for the DDE functions and the DDE Input and Output modules. There is a complete example shown in the Procedures section of the Help.

The screenshot shows the DASYLab Help window with the following content:

Send Data through the DDE Interface

Use the DDE interface to exchange data with other programs or applications on the same computer. In the following example DASYLab is the DDE client and sends data to Excel, the DDE server.

Note For this procedure you need Microsoft Excel on your computer.

Complete the following steps to generate data with the [Generator](#) module:

1. Open DASYLab.
2. Select **Modules»Control»Generator** from the browser and drag the module into the worksheet.
3. Select **Without modulation** and click **OK**.
DASYLab inserts the module in the worksheet.
4. Double-click the generator and select **Curve form»Noise**.
5. Click **OK**.

Complete the following steps to send data from the [DDE Output](#) module to the Excel program:

1. Open the **Excel** program.
2. Open DASYLab.
3. Select **Inputs/Outputs»DDE»DDE Output** from the browser and drag and drop the module into the worksheet.
4. Drag the input of the **DDE Output** to the output of the **Generator** module in order to connect the modules.
When you release the mouse button, DASYLab automatically connects the modules.
5. Double-click the **DDE Output** module.
6. Click the **Server** button to specify the DDE server.
7. First select **Excel** in the **Server (application)** column and then **[Book1]Sheet1** in the **Server topics** column.
8. Click **OK** to close the dialog box.
9. Select **Item»Definition»Generated**. This enables the **Item** button.
10. Click **Item**.
In this dialog box you compile the item name from three text fragments and two counters. The dialog box is predefined so that DASYLab writes ten values into ten columns respectively. Then DASYLab starts in the first line again and overwrites the numbers with the current values.

Note The system language set in Windows determines the names for the columns and lines. Address the first line in German with **Z1S1**, in English with **R1C1**, and in French with **L1C1**.

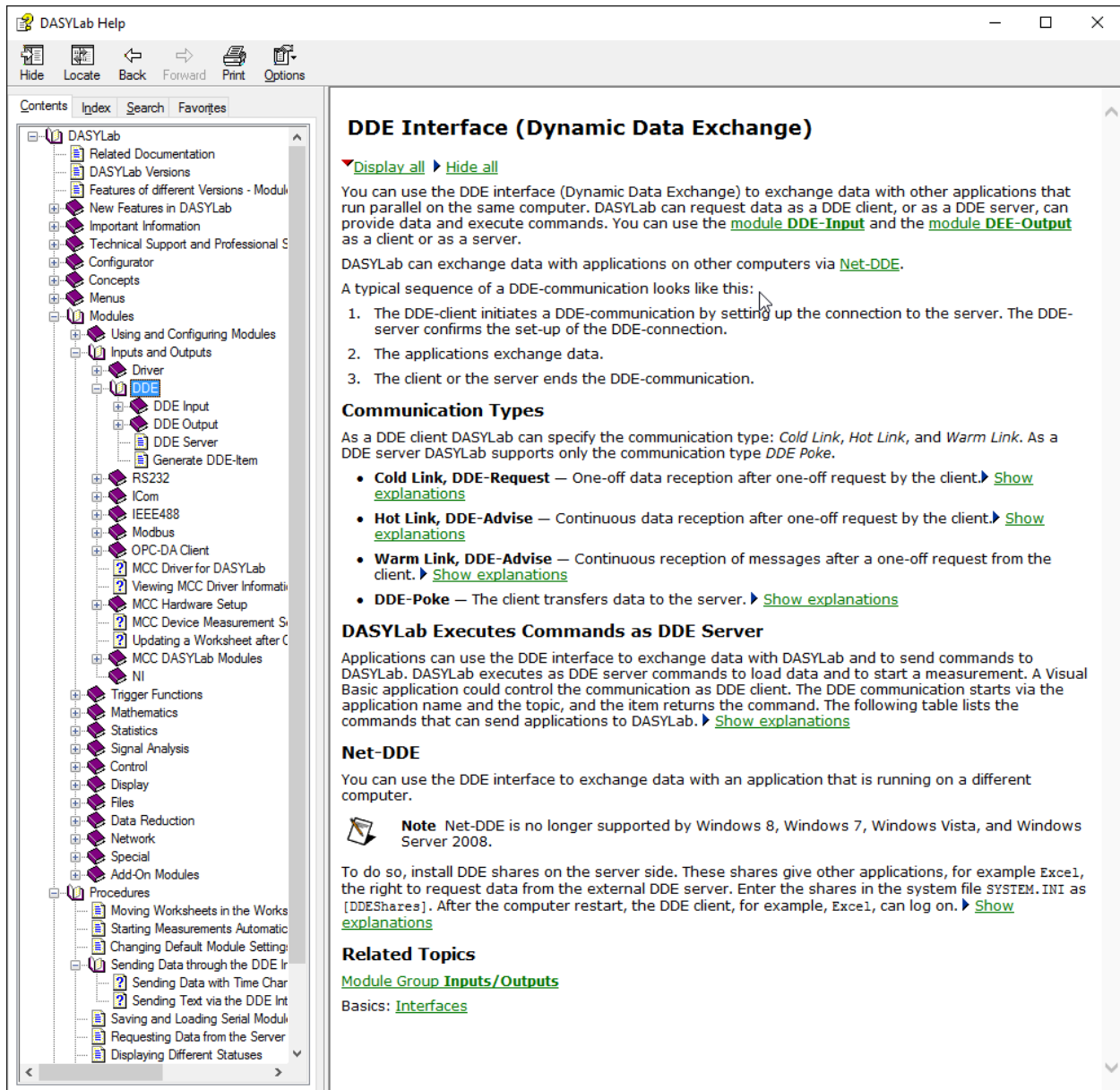
11. Click **OK** to close the dialog boxes.
12. Click **Start** on the group bar.
13. Click **Excel** on the status bar to move the application to the foreground. You can see how DASYLab writes the values into the first table.

Related Topics

[DDE Interface](#)

Basics: [Interfaces](#)

The DDE module help provides some overview information, as well as module specific help.



Examples

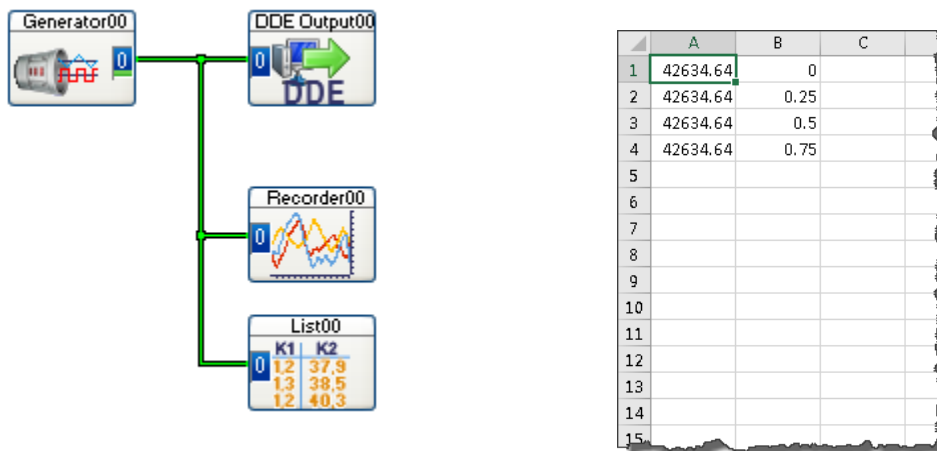
The following examples can be found at

ftp://ftp.mccdaq.com/downloads/example_programs/DASYLab_App_and_App_Notes/

Write to Excel at 1 samples/second

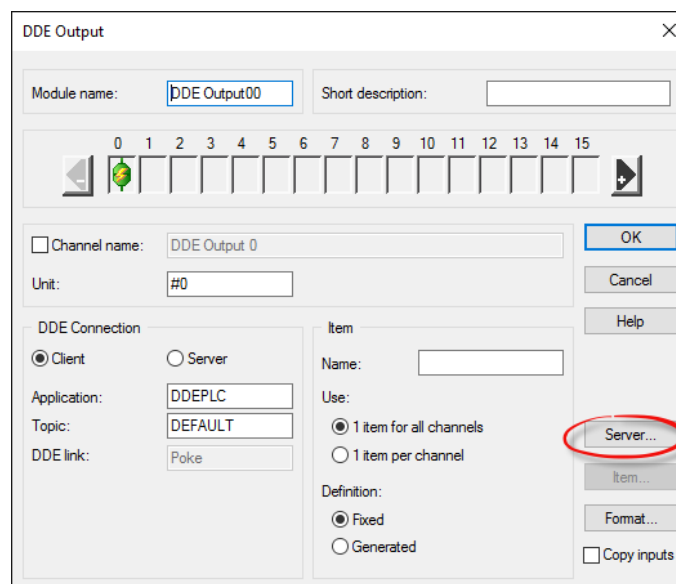
DASYLab 2020 Write to DDE 1s-second.dsb

This worksheet is the straightforward example for writing to Excel. It stores two columns of data, row by row. Column 1 ("A") is the time, and Column 2 ("B") is the data value. Each DDE Output module can write up to 16 columns. Use more than one module for additional channels.

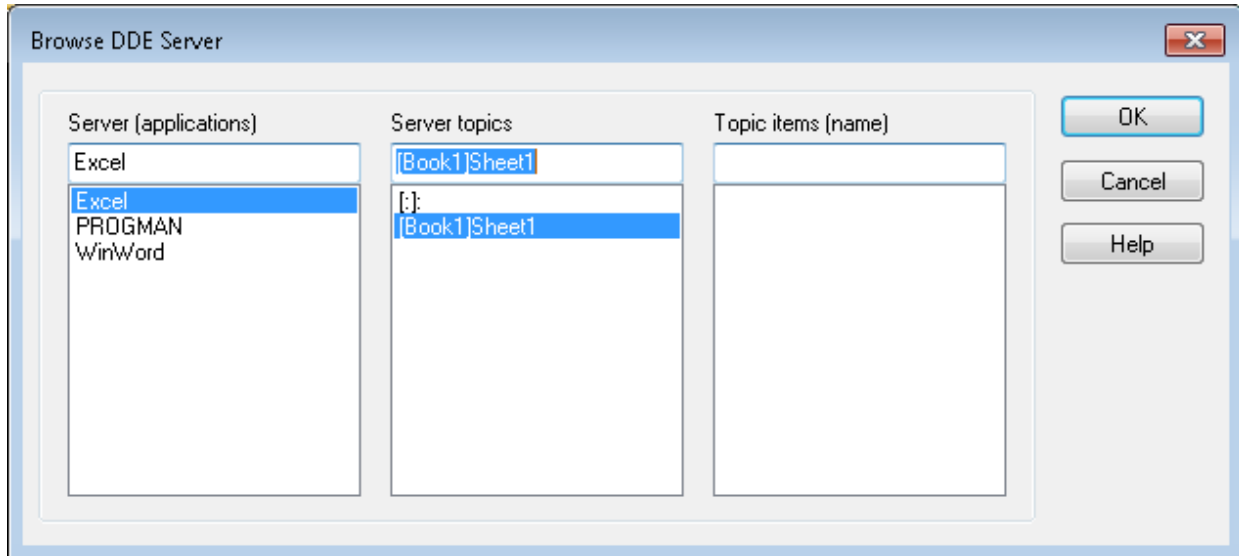


To set this up, create the Input/Output -> DDE -> DDE Output module.

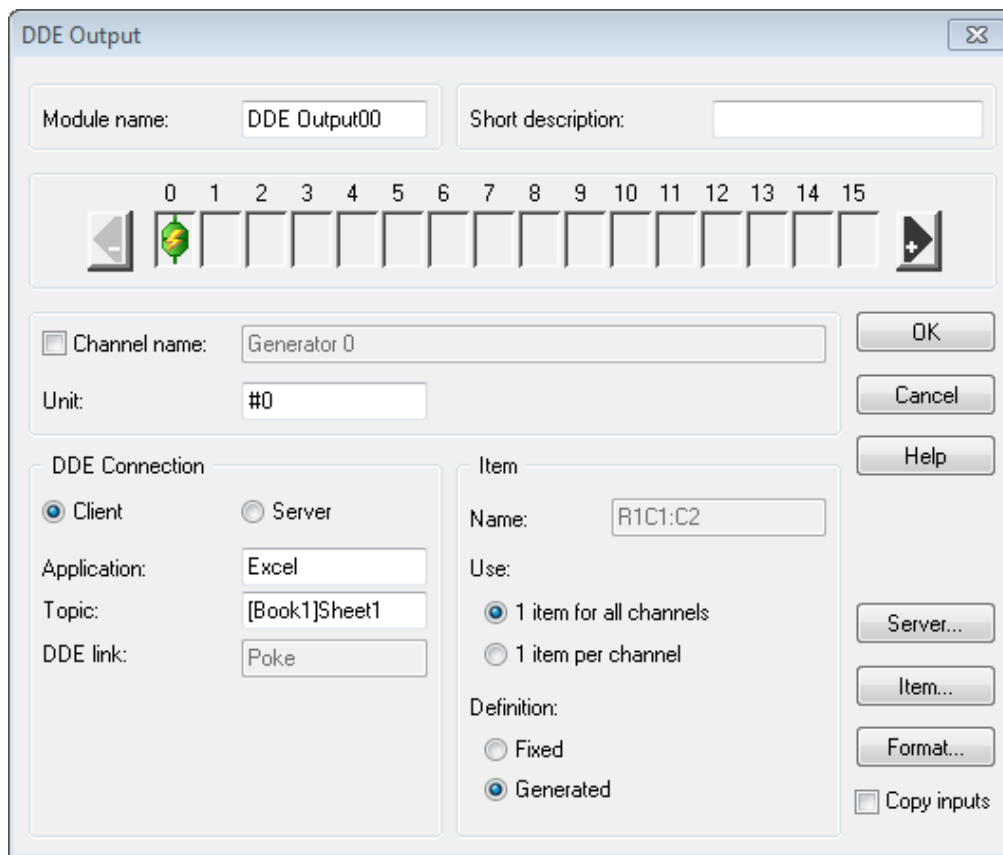
Configure the module to use Excel as the Server. Do this by clicking on the Server Button.



In the Browse DDE Server dialog, select Excel (it must be open), and then select the sheet. I have chosen to use a new workbook with the default names.



Next, select Generated for the definition field. This setting allows you to create a variable item name.

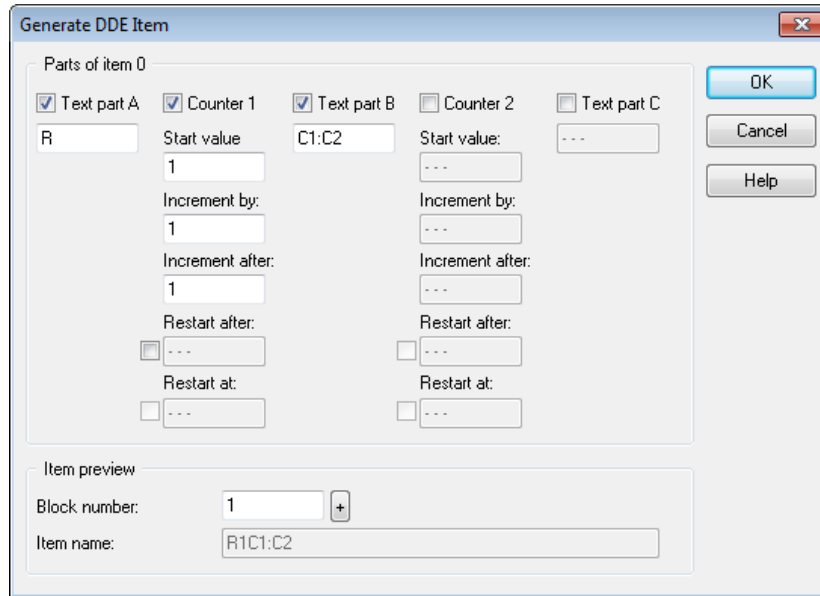


Click on the Item... button. To save two columns of data, row by row, enter the following

Text part A: R

Counter 1: Start value 1, Increment by 1, Increment after 1, no restart.

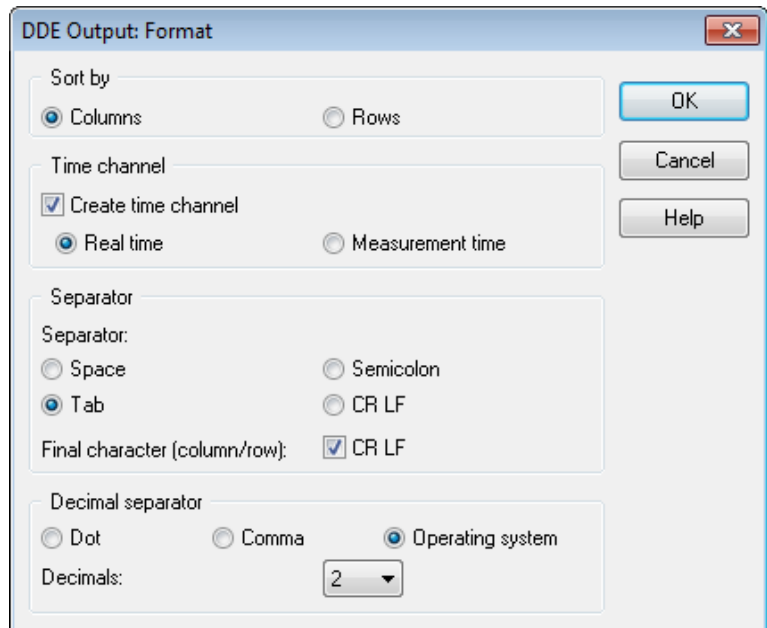
Text part B: C1:C2



These settings will create a valid Excel cell location, R1C1:C2, which would look like A1:B1 in normal Excel notation. DASYLab cannot easily iterate on the A1:B1 notation, but can easily iterate on the Row/column notation, R1C1.

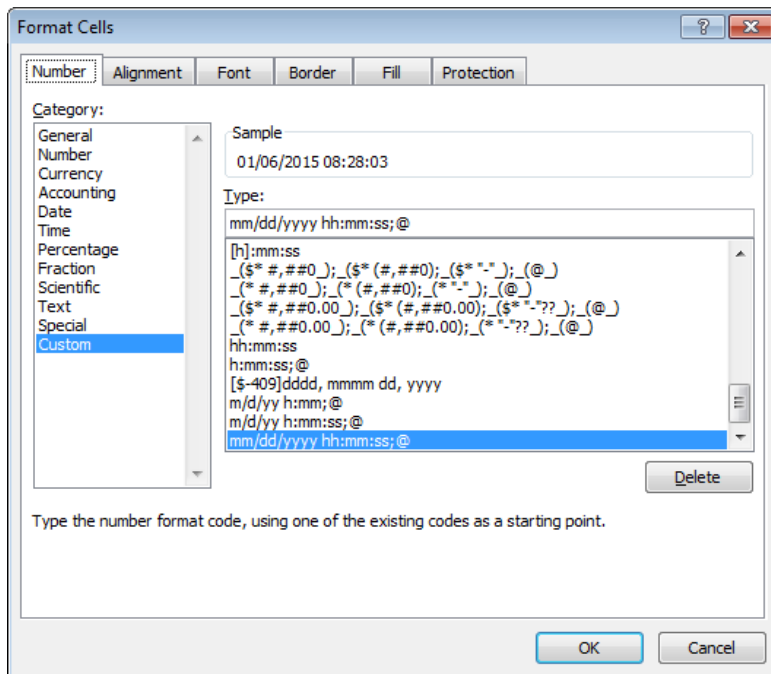
To include time, click Ok to close the Item properties, and click on the Format... button on the main DDE Output properties dialog box.

Check the Time channel setting to select Real time. This will put the current clock time into the first column. If you only want the Measurement time, select that to store the number of seconds since the start of measurement.



Click OK to close both dialog boxes. Make sure that the worksheet is wired, and that the sample rate is slow (1 sample/second). Start the measurement, and observe the data being stored.

Note that the time value stored in the Excel sheet is a number, not a date or time. You will need to format the entire column to use a date/time format. The value does include the date, so you will need to use a custom Excel format to show both date and time with seconds. The format mm/dd/yyyy hh:mm:ss will yield a complete date - 01/06/2015 08:28:03 using a 24 hour clock.



DDE to Excel does have a major limitation:

- it will always start at the beginning, possibly overwriting your data.
- You must remember to save the worksheet by a new name, and reopen the original worksheet when storing new data.

See the example below to send commands to Excel to automatically save the Excel sheet, or see the example Script that is posted at the kb.mccdaq.com to use a Script module to open your Excel sheet before the measurement starts. The example includes a module that you can import and use without understanding the Script module.

Search kb.mccdaq.com for DASYLab+DDE+Script or click on this link

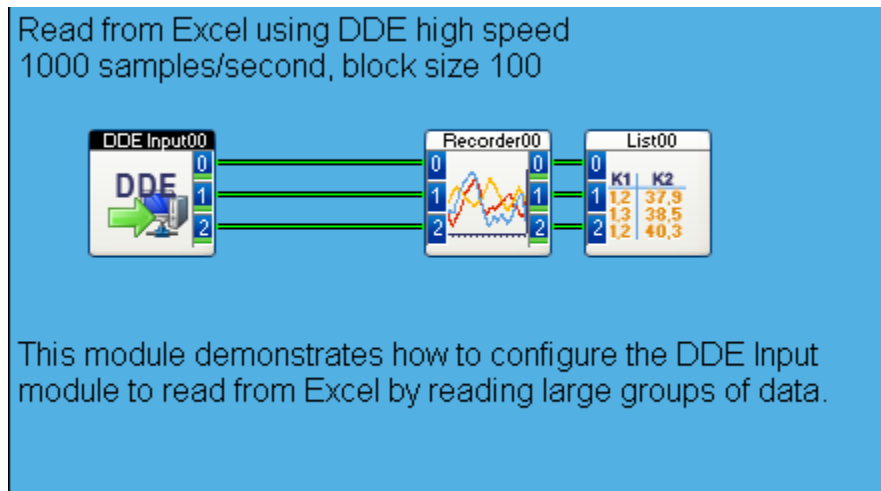
<http://kb.mccdaq.com/KnowledgebaseArticle50492.aspx>

Read from Excel sheet at high speed

DASYLab 2020 DDE Input 1000s-s 100b.dsb

This worksheet is configured to read from an Excel sheet at 1000 samples/second with a block size of 100. Treats DDE file as a table, where the row is the data. Uses DDE Cold Link to read at specified rate.

The DDE Input module uses a generated item list to read from Excel in blocks of 100. By using a large range, rather than reading one cell at a time, DASYLab can acquire the data from Excel at an effective sample rate of 1000 samples/second.



The DDE connection is a "Cold Link", so that DASYLab can control the timing of the read.

The DDE Input item generation is configured to read in blocks of 100.

DDE connection

Client Server

Application: Excel

Topic: [DDEBigExample.xls]

DDE link:

Hot Link Warm Link

Cold Link Poke-Link

Repeat data request

Time interval [s]: 0.10

Item

Name: R2C1:R101C3

Use:

1 item for all channels

1 item per channel

Definition:

Fixed

Generated

Generate DDE Item

Parts of item 0-2

Text part A Counter 1 Text part B Counter 2 Text part C

R	Start value	C1:R	Start value	C3
	2		101	
	Increment by: 100		Increment by: 100	
	Increment after: 1		Increment after: 1	
	Restart after: ...		Restart after: ...	
	Restart at: ...		Restart at: ...	

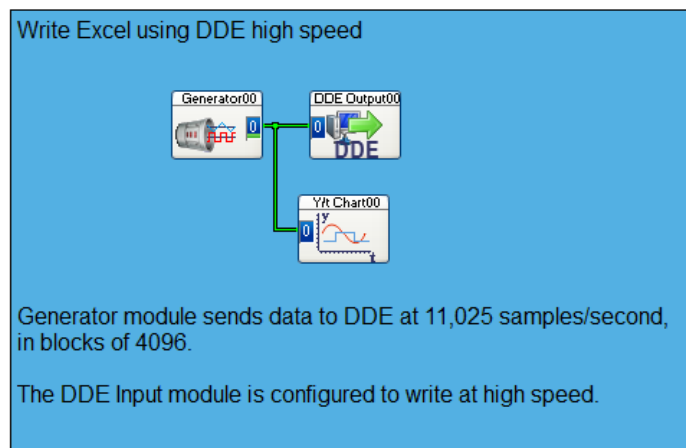
Item preview

Write to Excel sheet at high speed

DASYLab 2020 BigDDE.dsb

This worksheet is configured to write to DDE in blocks of 4096, writing the entire block as 4096 rows. The sample rate is 11.025Ks/sec (using the Sound card driver) with a block size of 4096.

Ensure that you have created an Excel sheet that is not saved in Compatibility mode . The Excel Compatibility mode limits the sheet to 65K rows. The maximum worksheet size for a current version of Microsoft Excel is 1,048,576 rows by 16,384 columns.¹



Configure the DDE Output module to write to the selected sheet. DASYLab will start with the beginning of the range each time, and will overwrite existing data.

This example works because the item generation creates a large Excel range for DASYLab to write into.

Generate DDE Item

Parts of item 0

Text part A Counter 1 Text part B Counter 2 Text part C

R Start value: C1:R Start value: C1

1 4096 4096

Increment by: Increment by:

4096 4096

Increment after: Increment after:

1 1

Restart after: Restart after:

... ...

Restart at: Restart at:

... ...

Item preview

Block number: 1

Item name: R1C1:R4096C1

OK Cancel Help

As noted in the Item name field at the bottom of the dialog box, the initial range is R1C1:R4096C1. It then increments by 4096, and the second block is R4097C1:R8192C1, and so on.

¹ <https://support.office.com/en-nz/article/Excel-specifications-and-limits-1672b34d-7043-467e-8e27-269d656771c3>

Send Commands to Excel

DASYLab 2020 DDE Messages.dsb

This example uses the Action and Message modules from DASYLab Full and Pro.

Use DASYLab and the Message module to send commands to Excel using the DDE command interface. Shows Minimize Excel Window, Restore Excel Window, as well as Save and Save As to the specified filename.

In setting up this example, I found that the syntax of the Excel commands was a bit finicky, and I had trouble finding a good description of available Excel commands. These commands were determined to work by trial and error. Some commands are to Excel itself (minimize, restore) and some are to the open Excel sheet (Save and Save As).

DASYLab Example for sending DDE Commands to Excel

Excel must be open with file named ddeexample.xls
User is prompted for the Save As filename



Action module sends commands to the Message module.
[APP.MINIMIZE()]
[APP.RESTORE()]
[SAVE()]
[SAVE.AS("\${STR_1}")] - name is in the global string #1



ExcelSystem Message module is the Excel SYSTEM topic for the minimize / restore commands



ExcelSheet Message module is the EXCEL specific worksheet topic for the save/ save as commands.
Change filename in this module (TOPIC)

Global Strings and Variables to Excel


DASYLAB 2020 Message_DDE_String.dsb

This example uses the Action and Message modules from DASYLab Full and Pro.


Use DASYLab and the Message module to send Global Strings and variables to a computed Excel cell range. This example concatenates variables and strings together and sends a single line with tab separated values.

DASYLab Example for sending DDE data to Excel using Global Strings and Variables

Data will be saved to the [Sheet1] of the open workbook

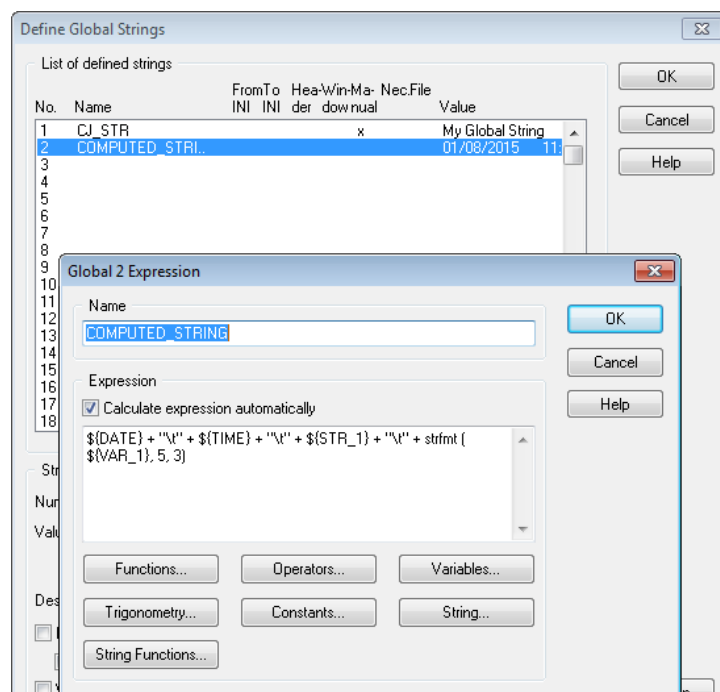


The Action tells the Message module to send $\$(STR_2)$
 The Message module uses DDE POKE to send the message using the computed TAB separated string
 Message uses a generated ITEM to determine the cells



This examples uses a Slider to save data to a variable.
 Review $\$(STR_2)$ to see how the date, time, string and value are combined into one string

The image below shows how the combined string was created in the Global String Advanced Definition.



Start Excel with a filename

DASYLab 2020 Startup for Win10-64.DSB

This example uses the Action and Message modules from DASYLab Full and Pro.

The example was created with Excel 2020 32-bit on Windows 10 64-bit. The location for the Excel program file is in Global string #1, and is "

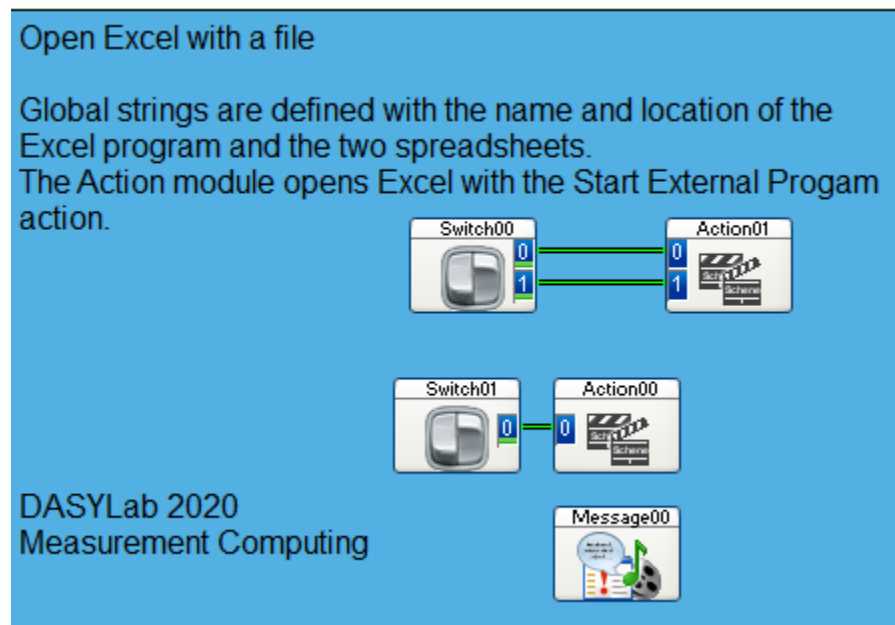
C:\Program Files (x86)\Microsoft Office\root\Office16\EXCEL.EXE"

in the worksheet. Change this string to reflect the correct program location on the computer.

The worksheet will start Excel with specific filename using the Action module, Start External Program. This must be run prior to starting a worksheet with DDE. It also has an Action to Quit Excel.

You cannot use this Action in a worksheet with the DDE module because the Action on Start Measurement will not start Excel quickly enough for the DDE modules.

This worksheet could be the first in a series of worksheets run by the Control Sequencer, where the this worksheet does the setup, starts up Excel, etc., and subsequent worksheets then use DDE to write to Excel.



The image below shows how the file name was created from the Excel program location and the Excel sheet location.

