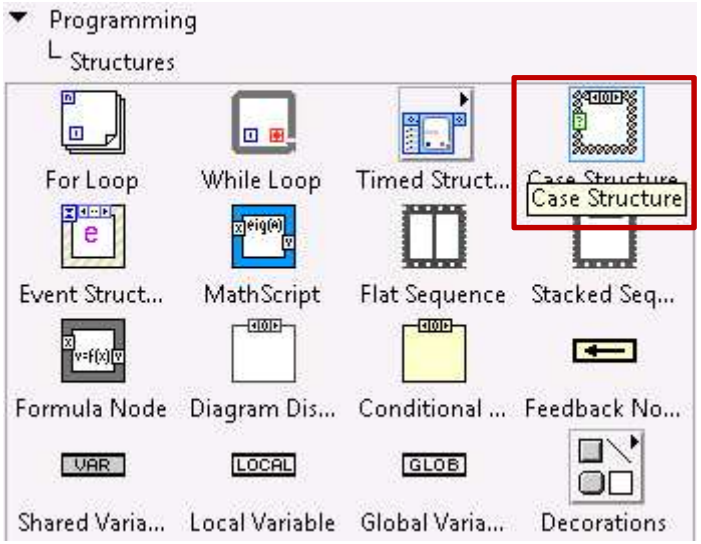
LabVIEW Lecture 2

Ertugrul Karademir

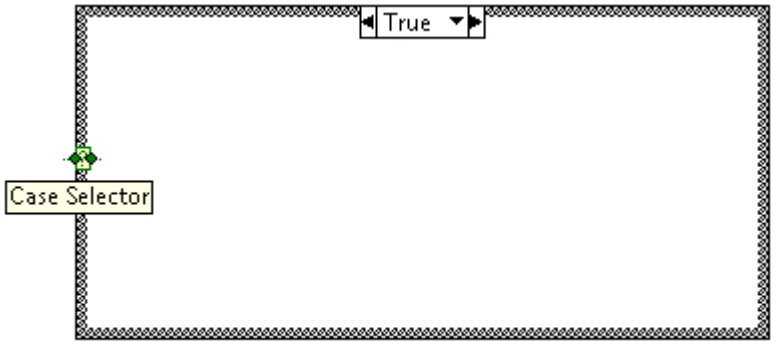
Case Structure

- Just like if...elseif...else

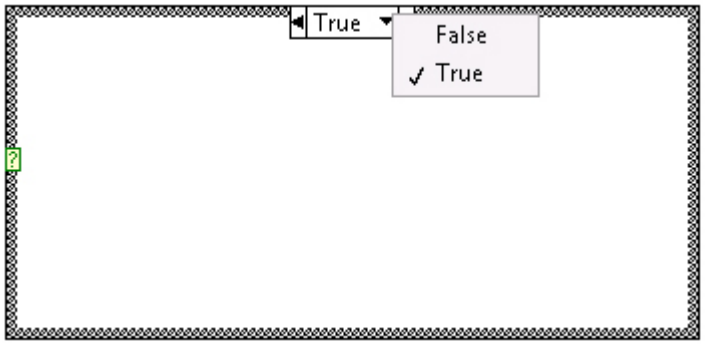


Case Structure

- Connect the case port

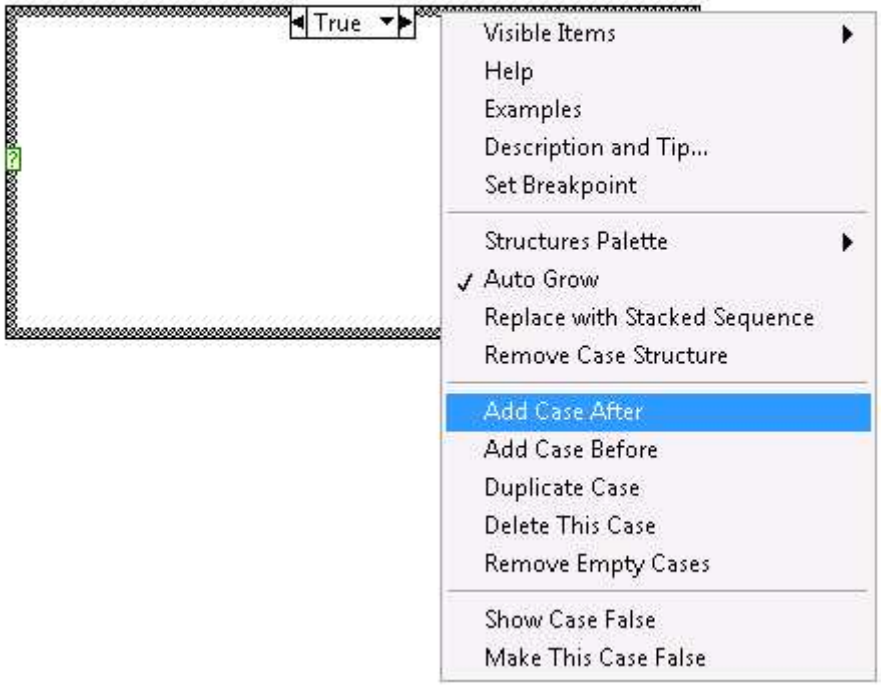


- Choose the condition you want to program



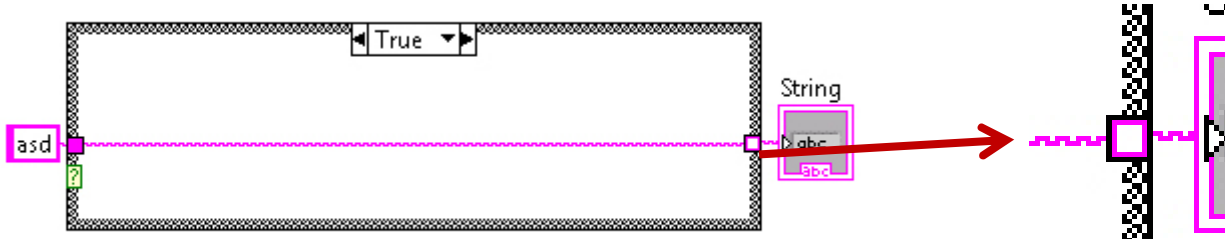
Case Structure

- You can add more cases by right clicking

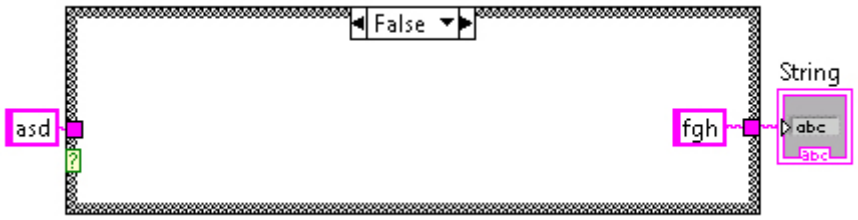


Case Structure

- All terminals for all cases must by connected



This is empty because, False condition is not wired

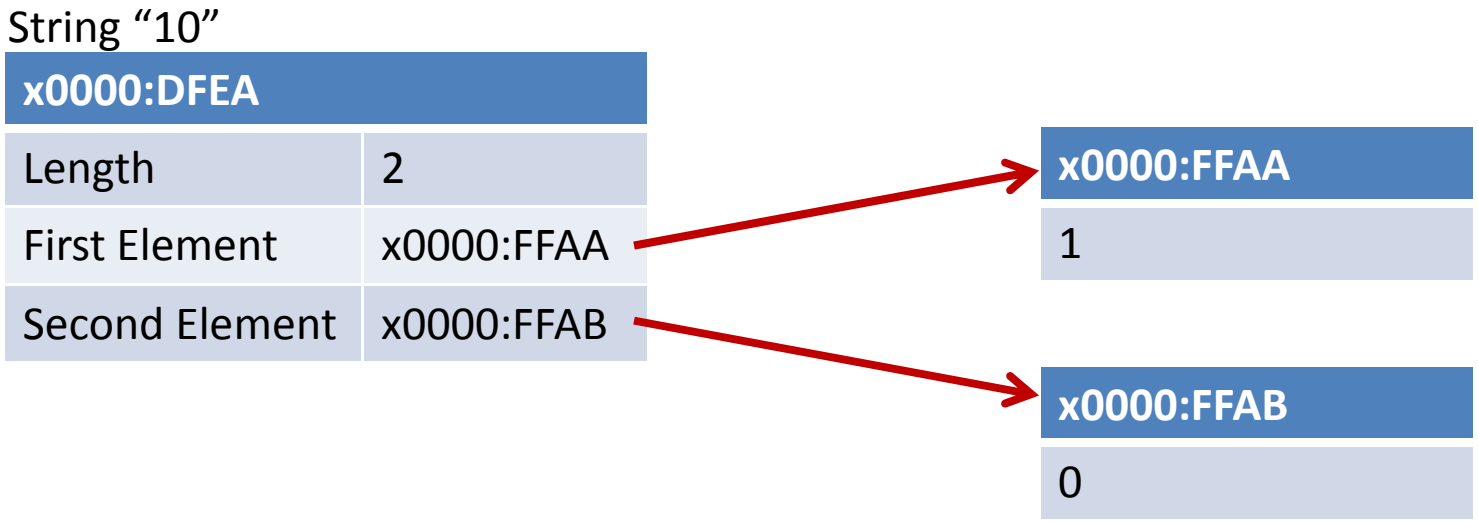


Variable Types

- There are basically three kinds of variable types:
  1. Integers: One memory register is enough to store. Bit value of the register defines the limits of cardinality.
  2. Floating point: more than one memory register is needed (one to store base, one to store significant digits, etc.)
  3. String: Textual information. It is stored much like Arrays. Actually a string is an array of characters (which are integers with special meanings).

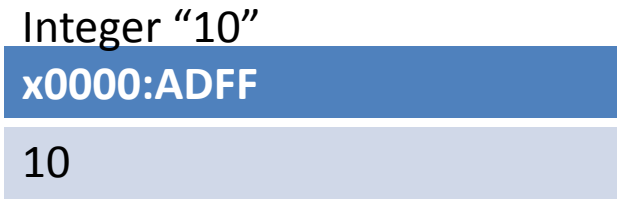
Variable Types

- When number 10 is stored as a String it is represented more or less like this:



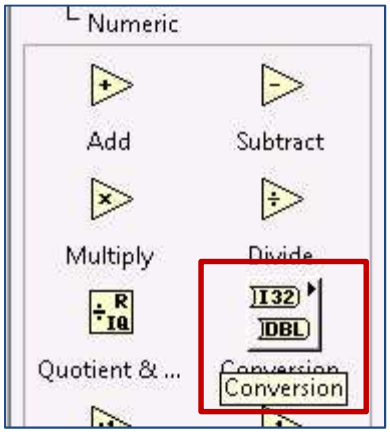
Variable Types

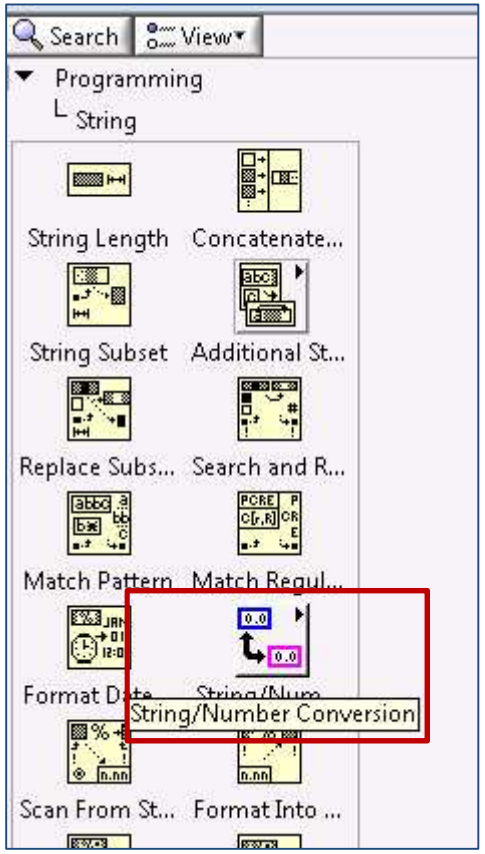
- When number 10 is stored as an Integer it is represented more or less like this:



Variable Types

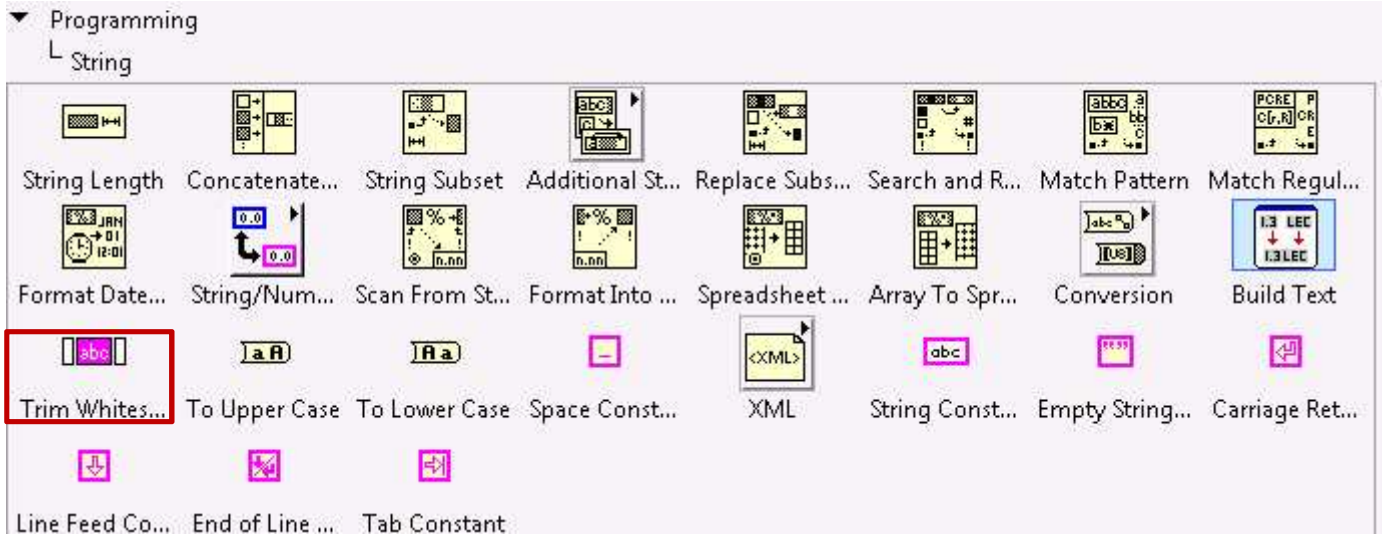
- Controls we use to communicate with instruments need and produce String data.
- Numerical values need to be converted approriately (type casting).





String Operations

- Find your way in string palette
- It's intuitive
- Don't forget to trim whitespace out of returned data from the instrument

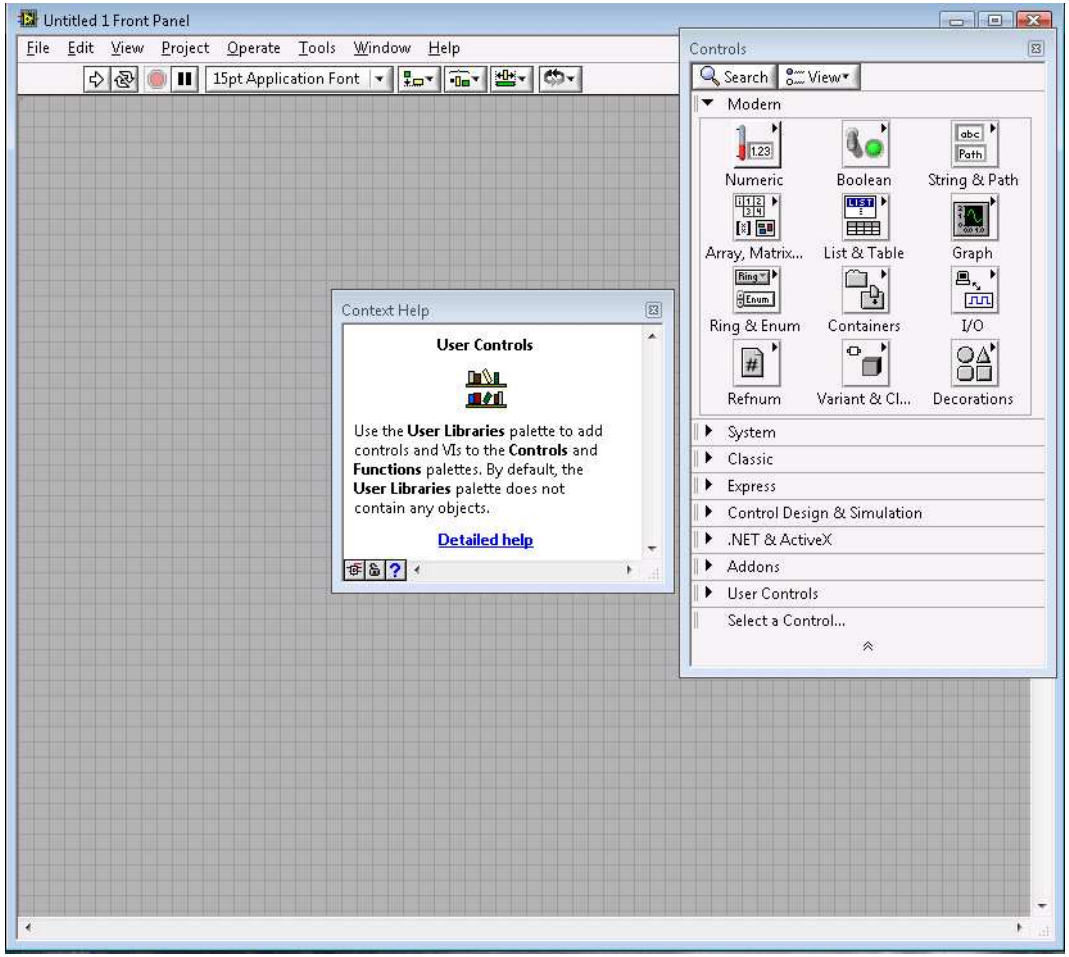


More Complex Topics

- From here on we focus on more complex topics:
  - Arrays
  - File Input Output
  - Data Acquisition

Modern Palette

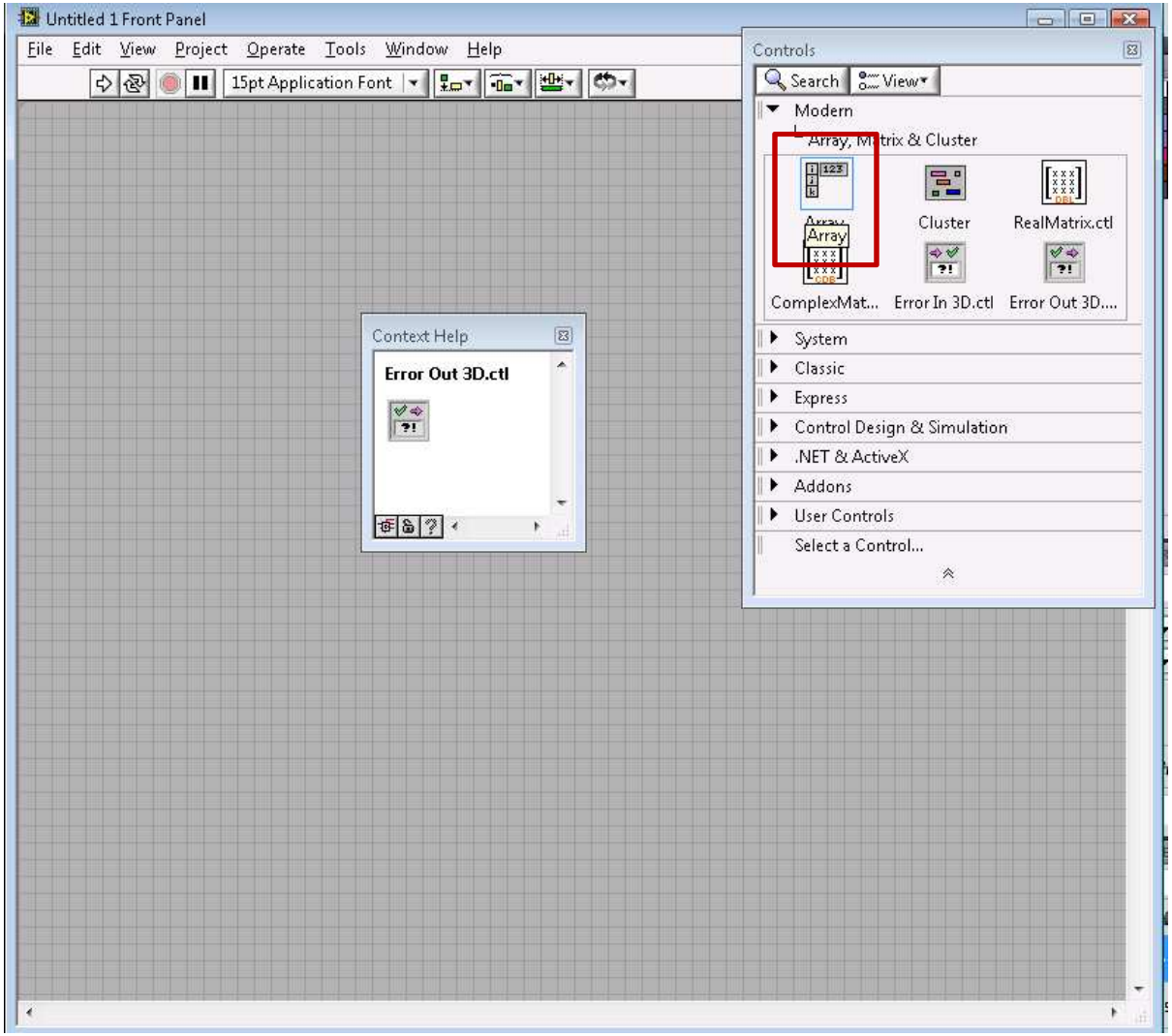
Sometimes we need more than what Express palette offers



Arrays

- Arrays are created as empty shells
- In order to assign a type you should drop in a control.
- Array casts into the dropped control’s type.

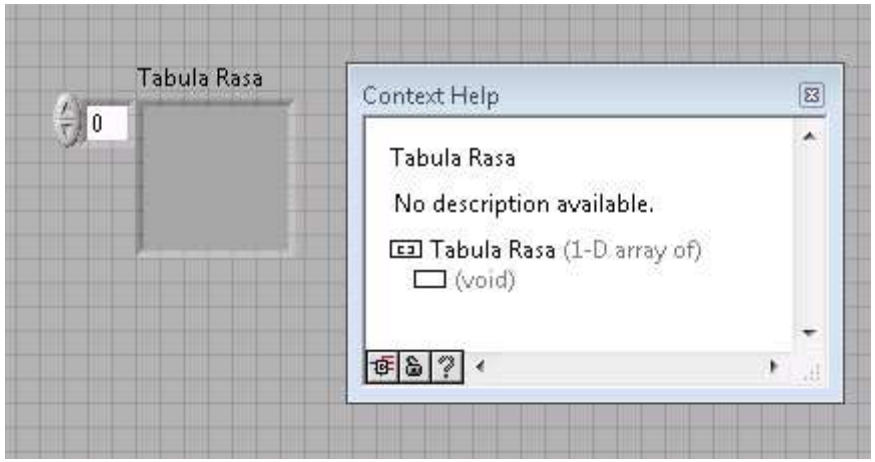
Arrays



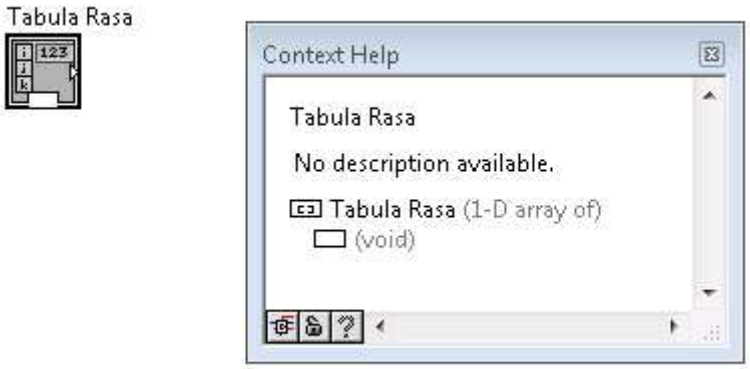
Array, Matrix & Cluster palette has the empty array shell.

Arrays

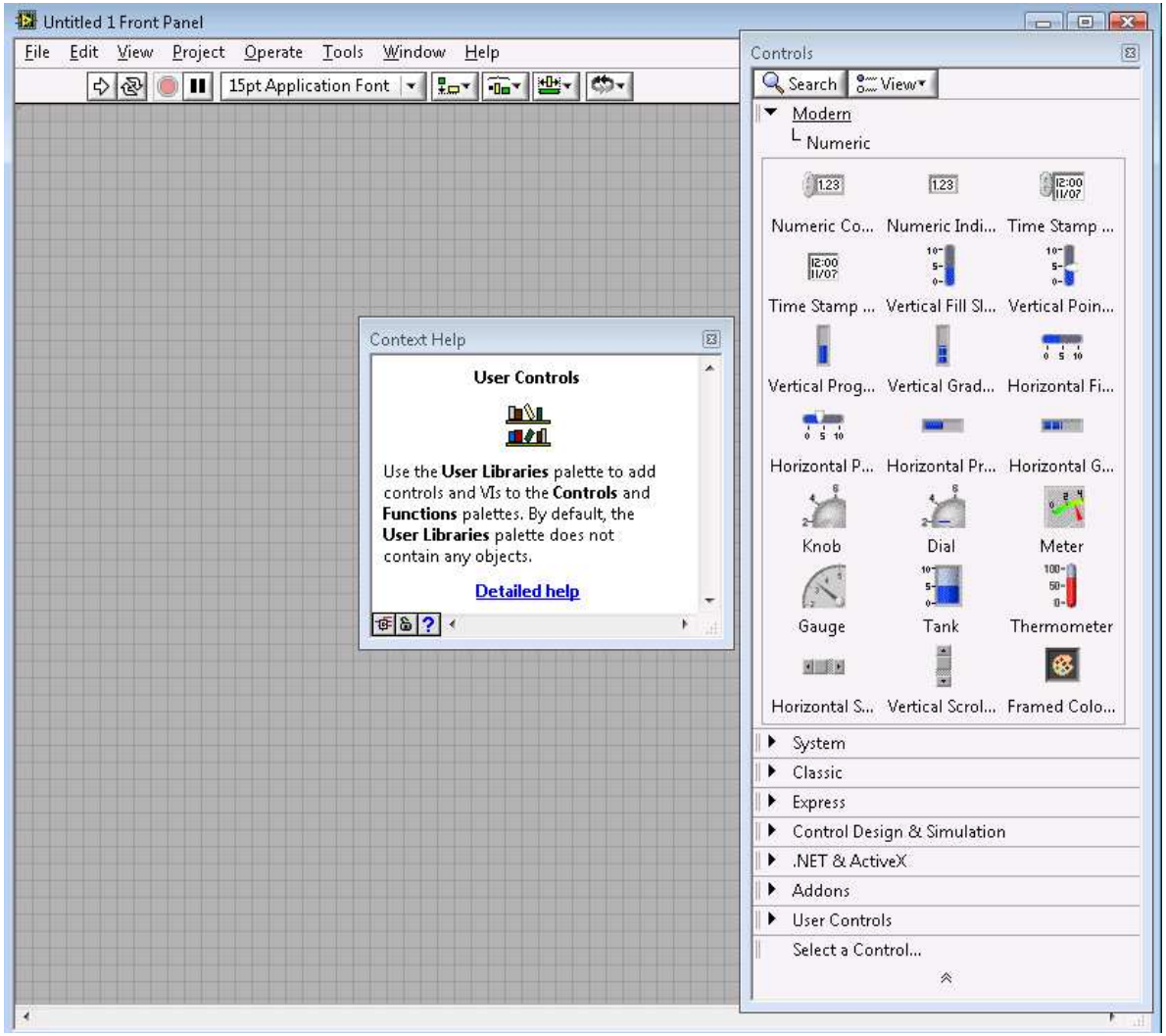
Create an empty array shell



LabVIEW creates a proxy for that array in block diagram



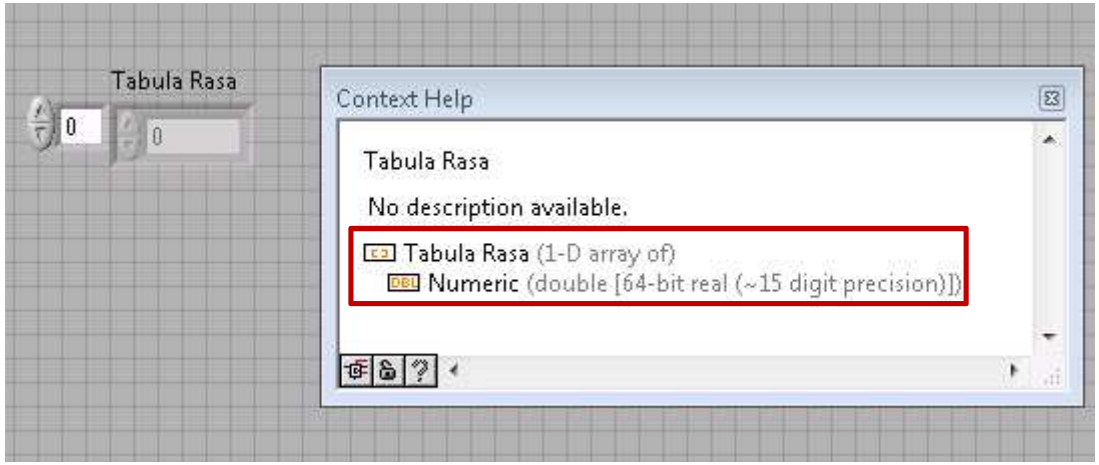
Arrays



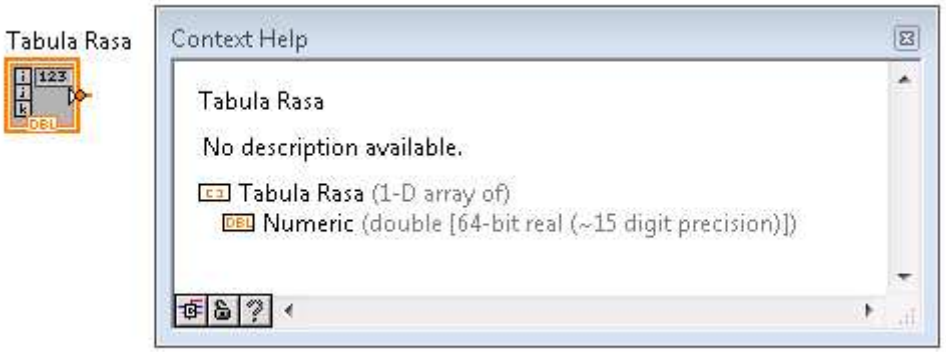
- Invoke Numeric Palette under Modern
- Drop a Numeric Control inside the empty box of the Array shell

Arrays

Now the array has a type

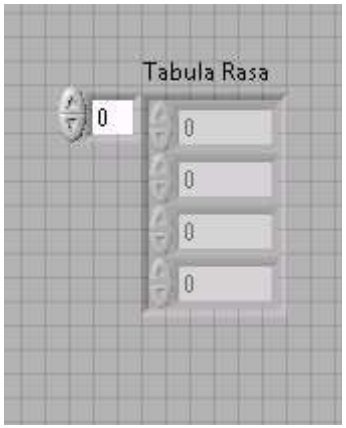


So does its proxy



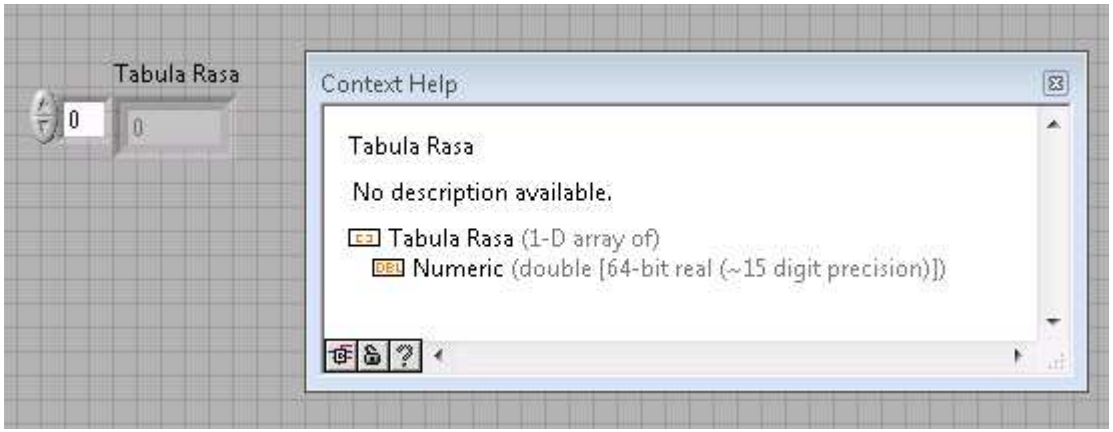
Arrays

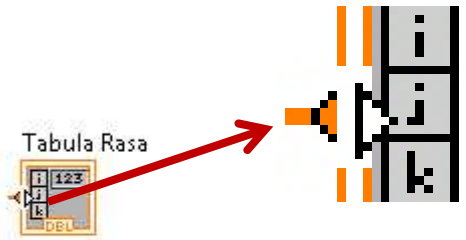
- You can expand the array container to show more elements at once
- Or click to the pager buttons to change the starting index



Arrays

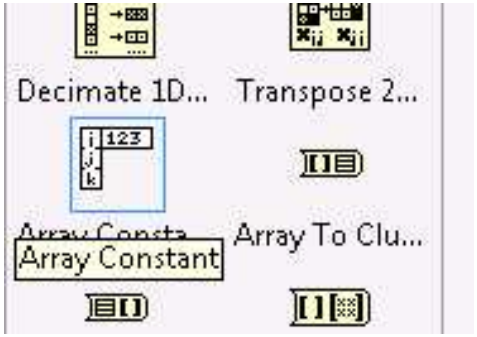
- Note that there is no array INDICATOR
- That is because you indicate that an Array is an indicator by dropping in an indicator inside that array shell





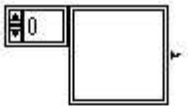
It is an indicator

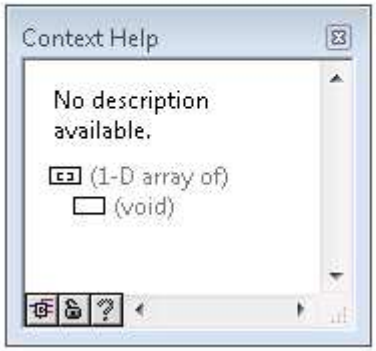
Arrays



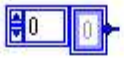
You can also create Array constants

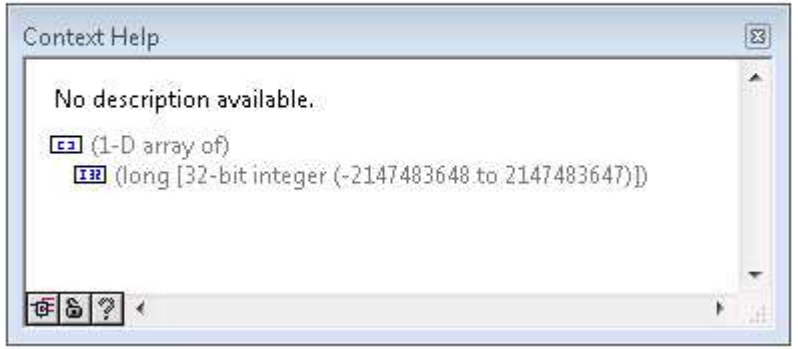
Programming Palette > Array Palette > Array Constant





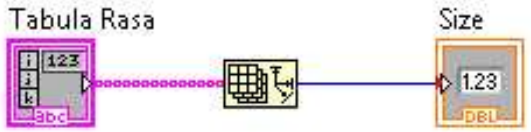
Drop an integer constant inside and get integer array

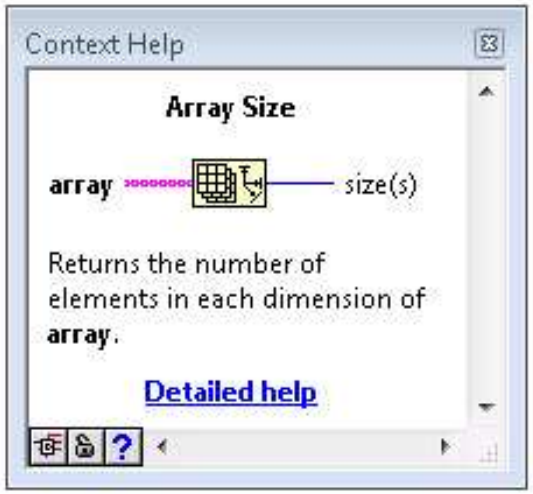


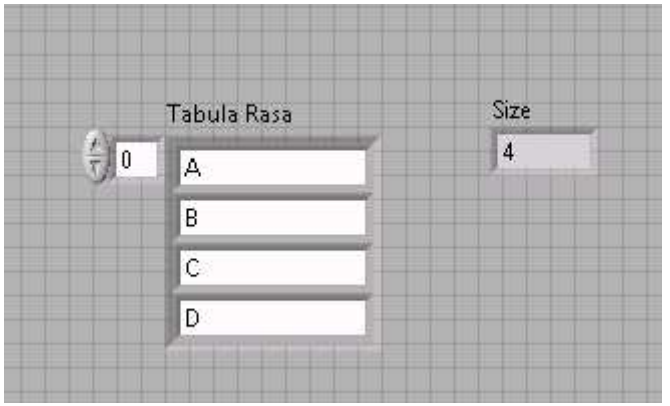


Arrays

- Arrays palette has many useful Array functions.
- Like getting the size of an array

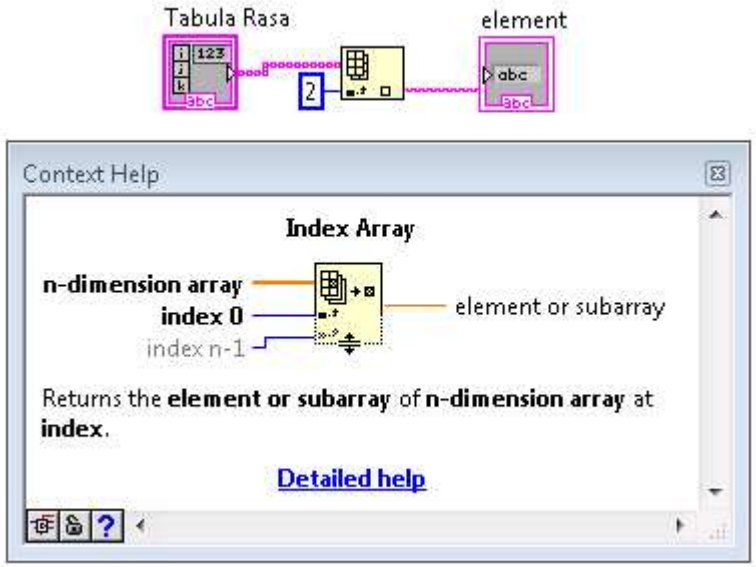


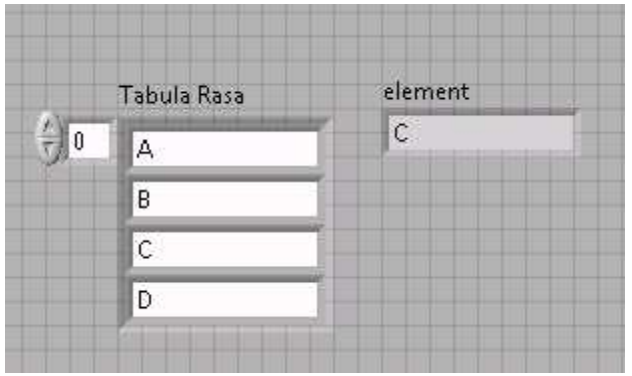




Arrays

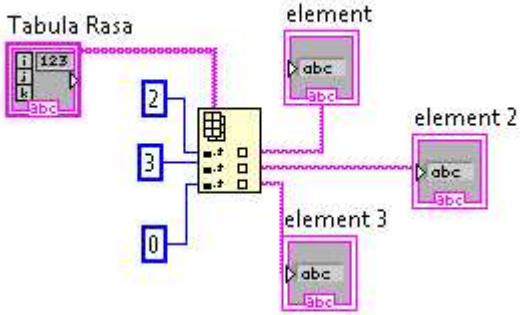
Get an element at a specific index.

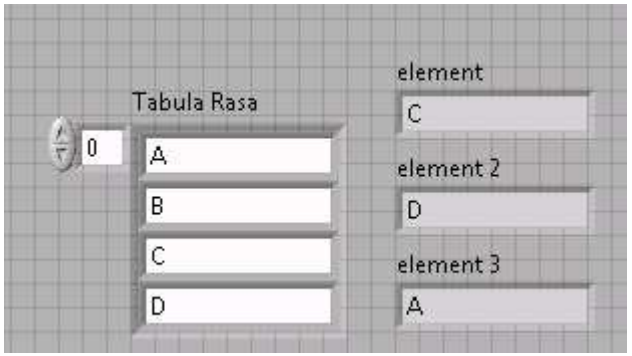




Arrays

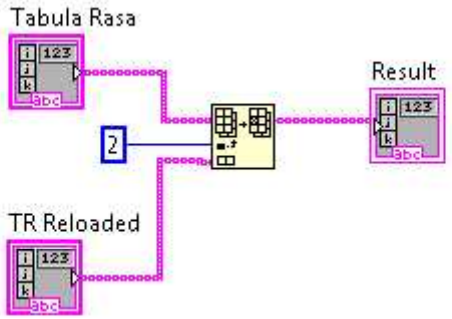
Or several indices (expand the icon)

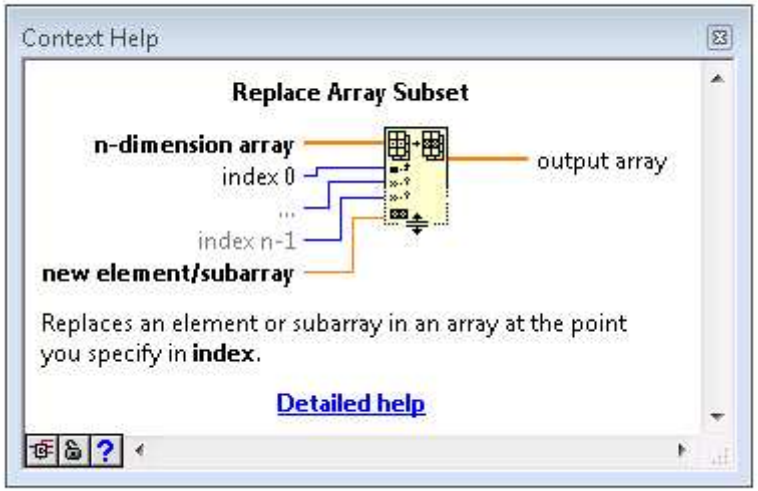


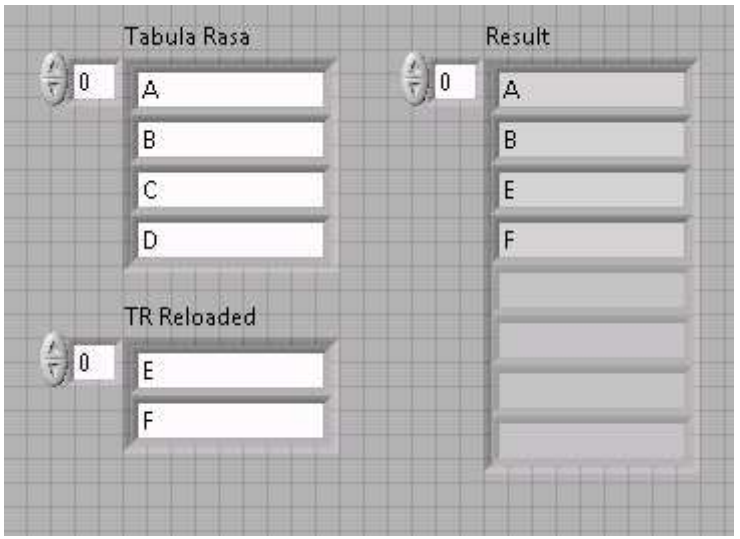


Arrays

Replace some elements

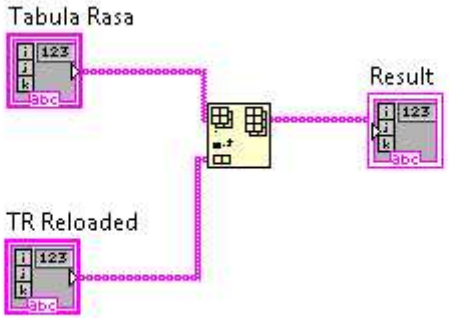


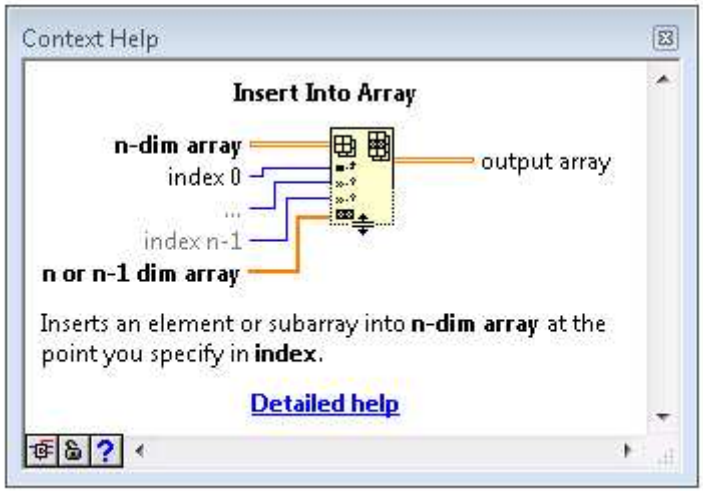


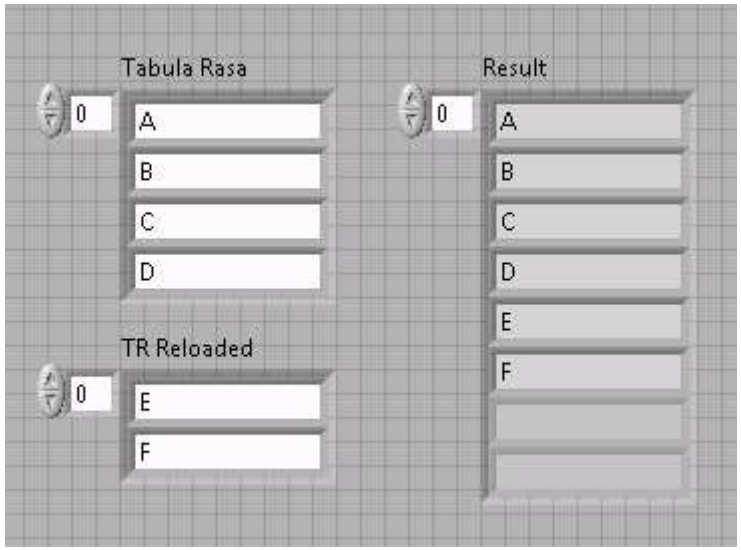


Arrays

Concatenate

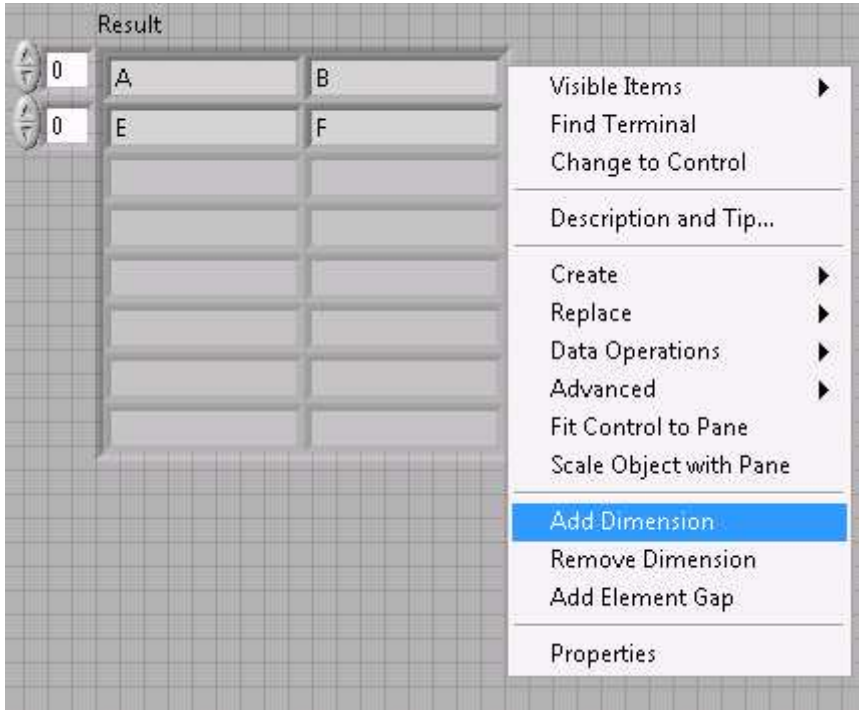






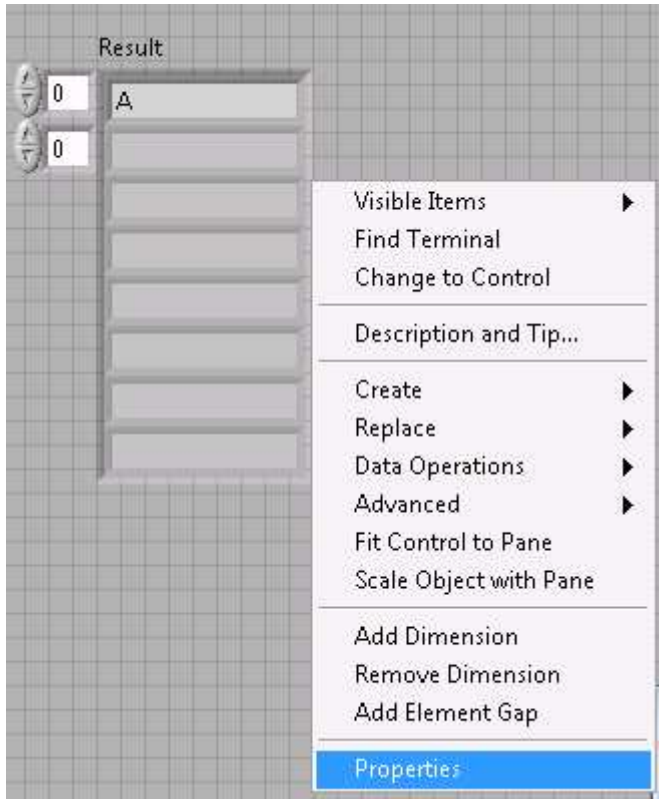
Arrays

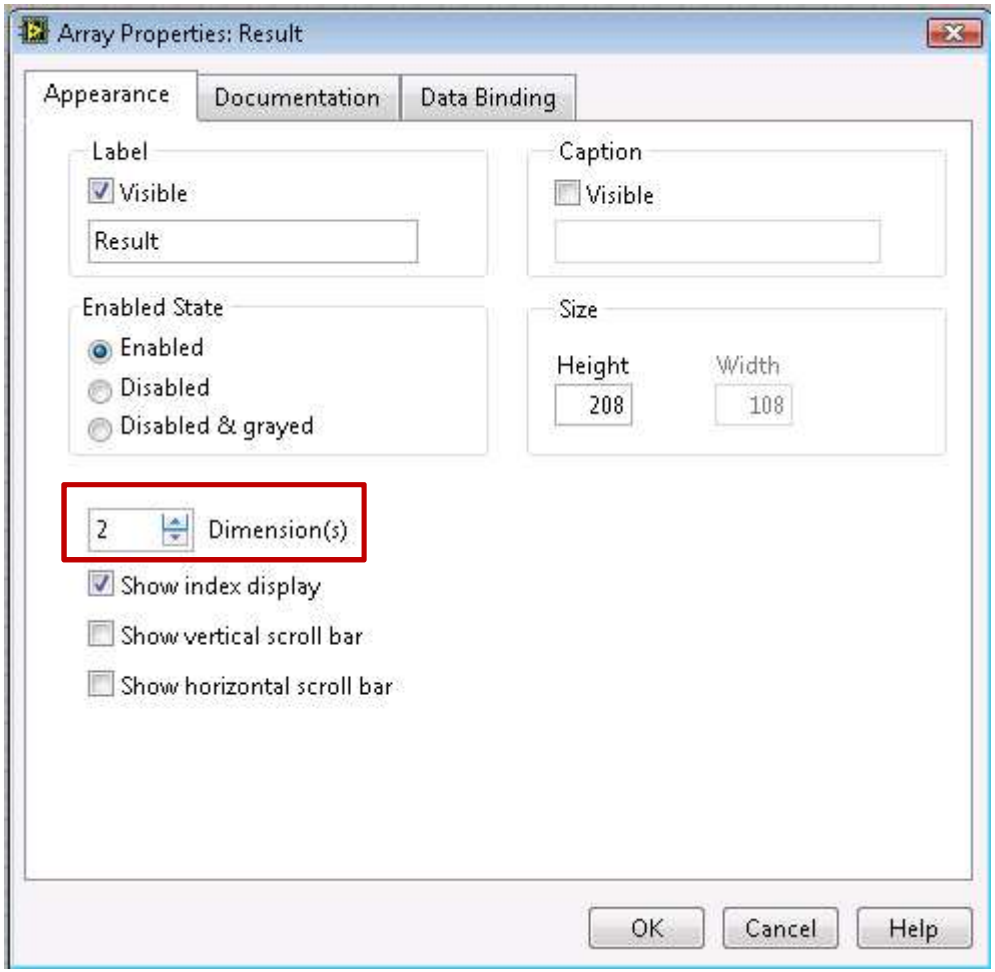
You can add dimensions to an array by right clicking onto the container



Arrays

Or from array properties.

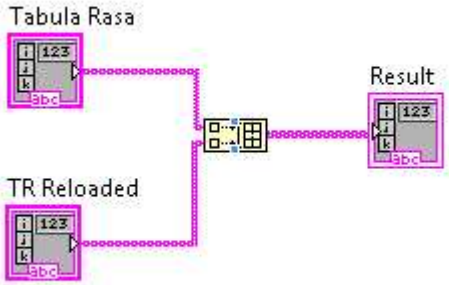


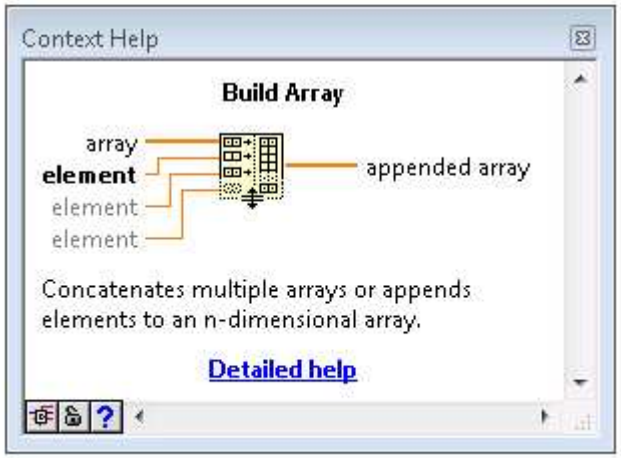


Arrays

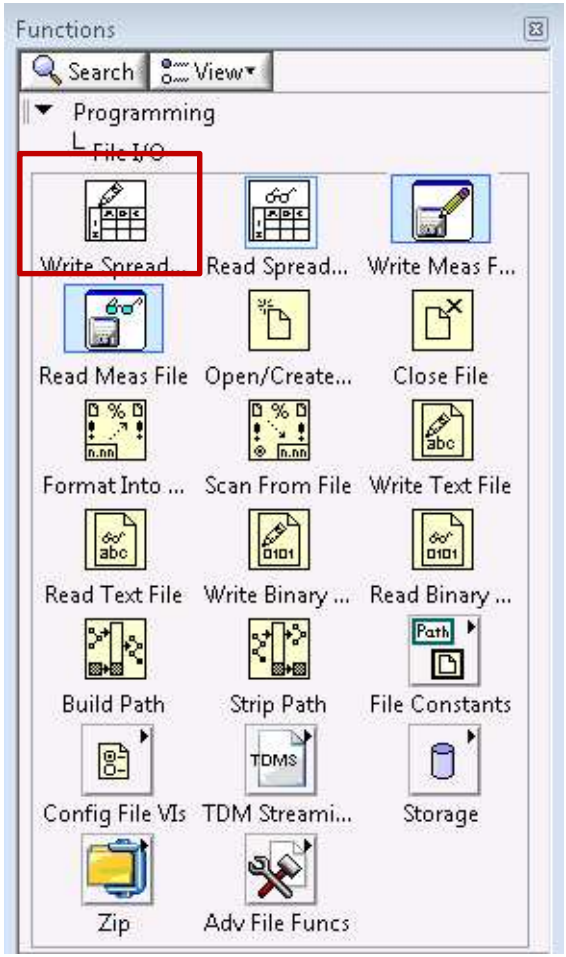
You can build more dimensional arrays by using Build Array function

Note that front panel representation has changed into row-wise indication rather than column-wise.



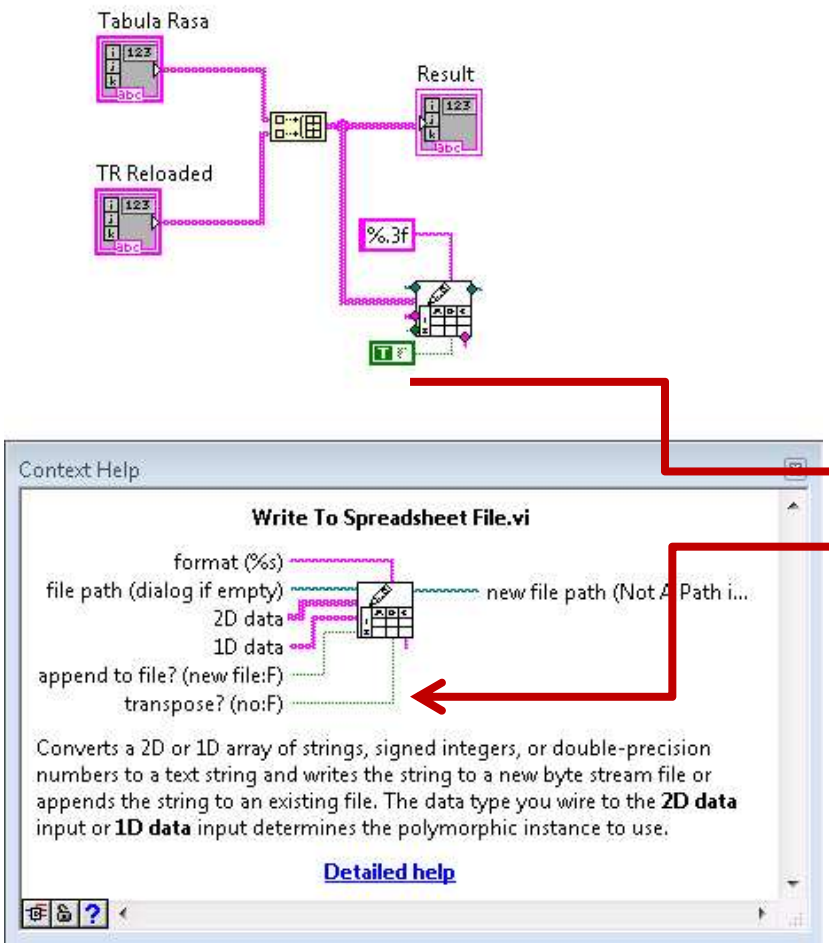


File I/O



- LabVIEW allows extensive use of file stream operations
- We'll only use Write Spreadsheet function

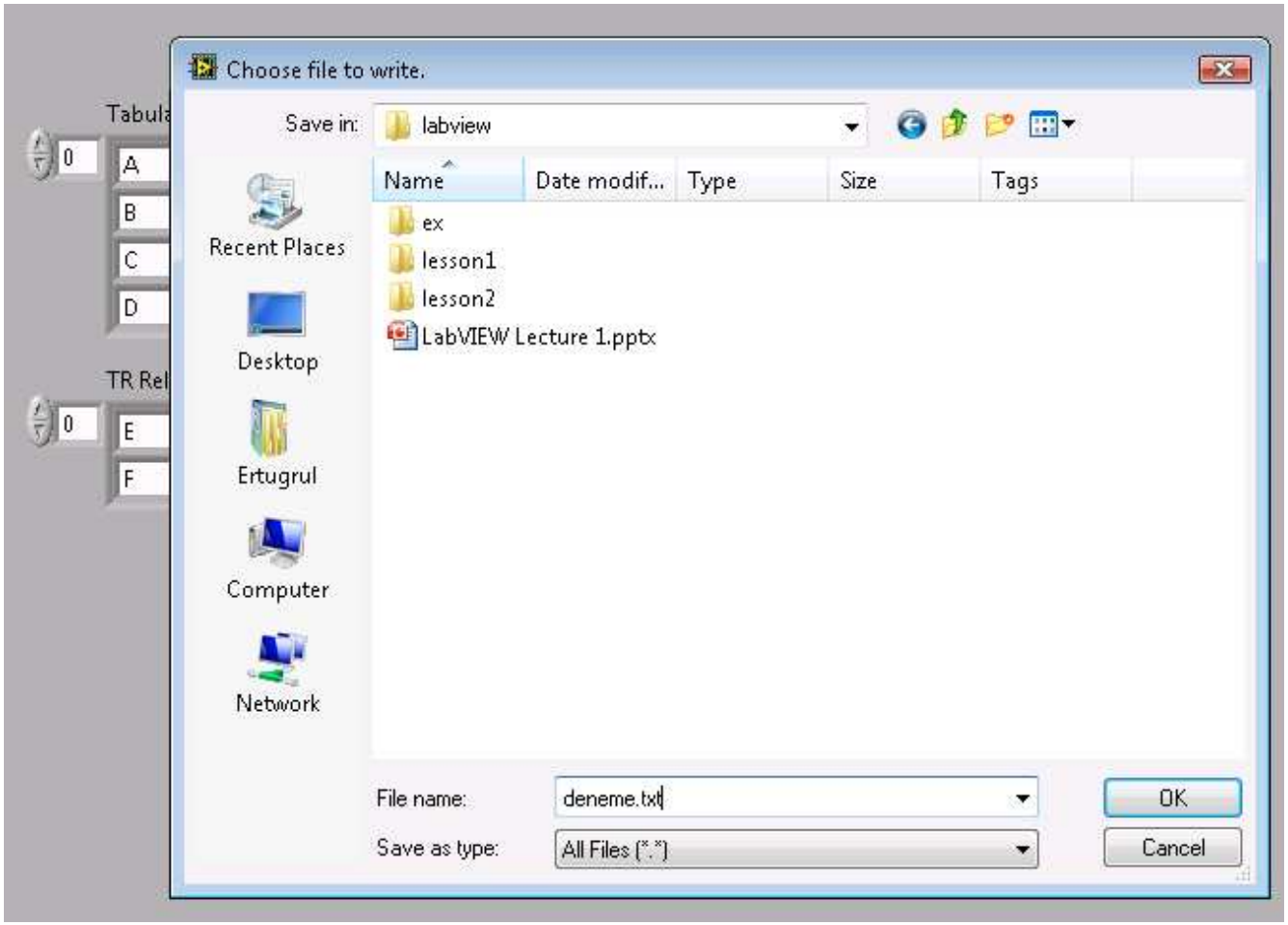
File I/O



All operations are done in Block Diagram

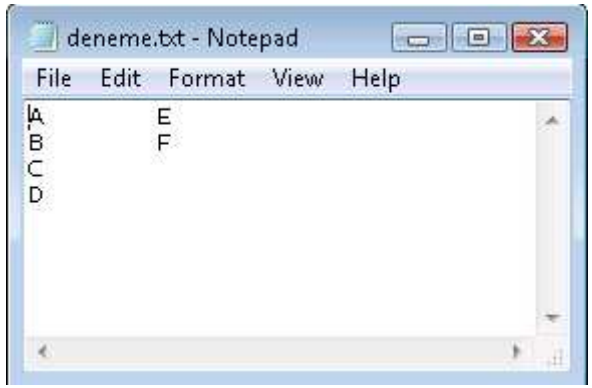
Note that we've transposed input data

File I/O


Choose file to write.
Save in:
labview
Name
Date modif...
Type
Size
Tags
ex
lesson1
lesson2
LabVIEW Lecture 1.pptx
Recent Places
Desktop
Ertugrul
Computer
Network
File name:
deneme.txt
Save as type:
All Files (*.*)
OK
Cancel
Tabul
0
A
B
C
D
TR Rel
0
E
F

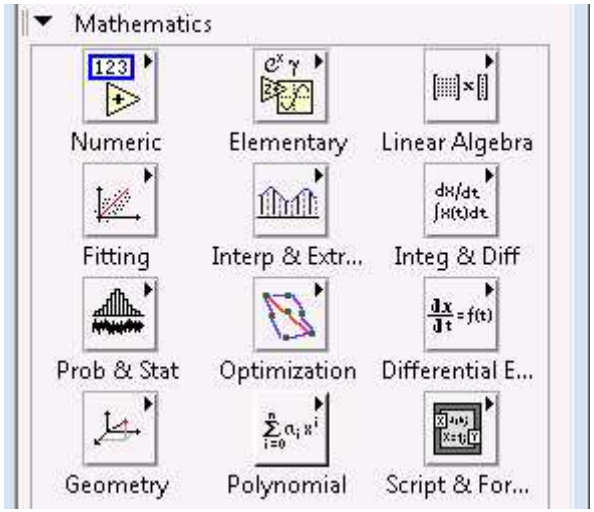
File I/O

Once saved, we can use this data in other applications like MATLAB or Origin



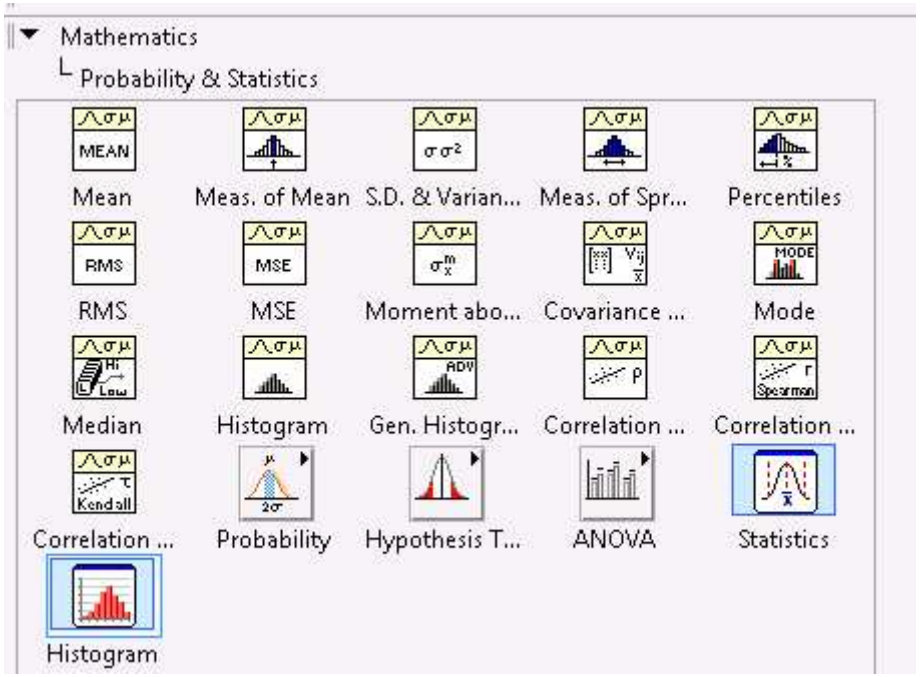
More

Besides Programming palette, LabVIEW has an extensive selection of Mathematical fuctions under Mathematics palette.



More

You may need some functions under Probability & Statistics palette.



Data Acquisition

- What LabVIEW does best
- There are many ways to communicate with instruments:
  - GPIB 
  - RS232 / Serial Port 
  - LPT / Parallel Port 
  - USB 
  - TCP/IP 

GPIB

- Standardized as IEEE-488
- a.k.a. HP-IP; Hewlett Packard/General Purpose Interface Bus
- Has deep similarities to parallel port
- 15 instruments can be connected to one bus

Pics: Wikipedia

GPIB

- Every instrument has it, even the very old ones

Connection

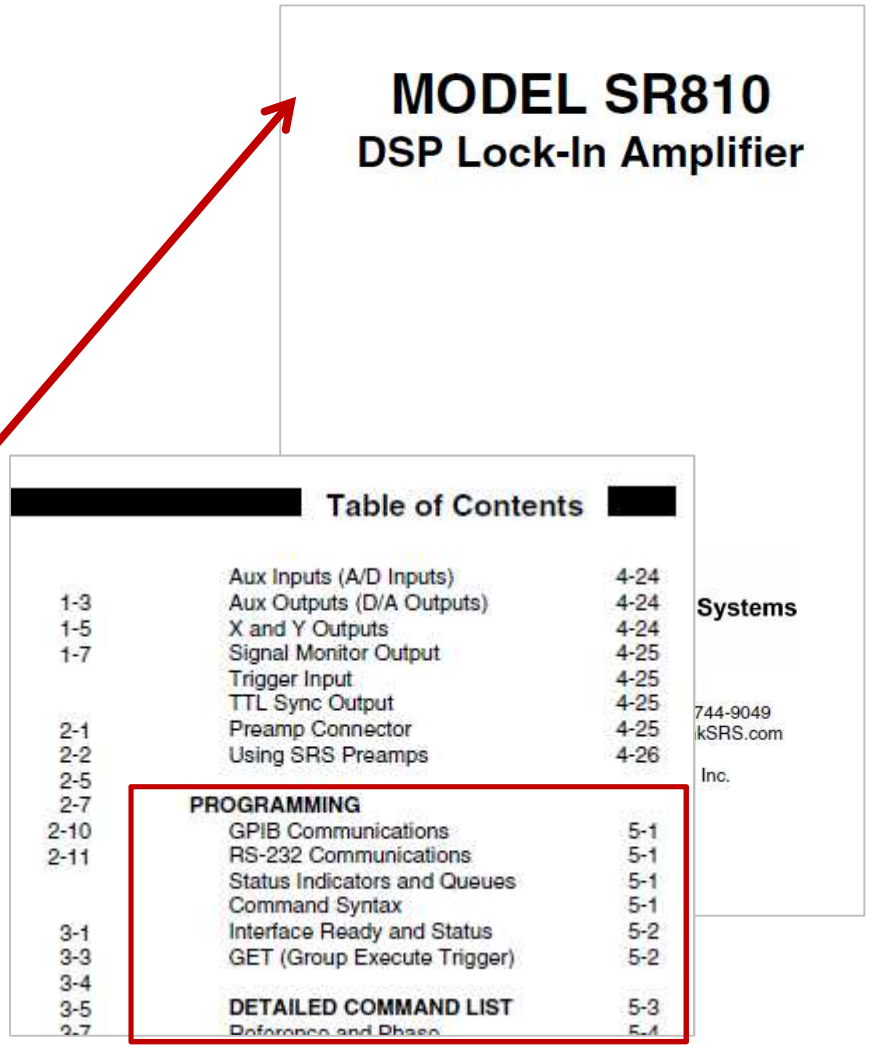
- Every instrument has a different
  - Command set
  - Data I/O pipeline
  - Response time
  - Buffer size
  - …
- What to do?
  - RTFM

Connection

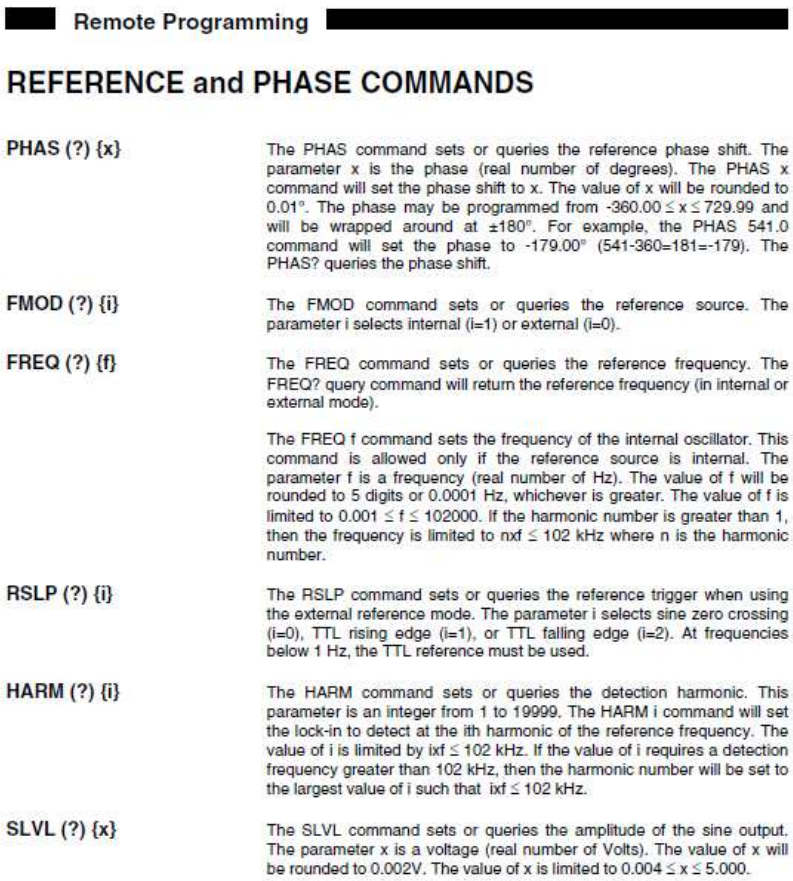
- Internet is a good source for instrument manuals, even the old ones
- Generally instrument vendor site is enough
  - unless it stopped manufacturing that device or upgraded with a new model
  - even then, decent vendors have an archive for old documents
- Find a manual or you waste your time by guessing the command set!

Connection

- Seek for the computer interface chapter



Connection



Remote Programming

REFERENCE and PHASE COMMANDS

PHAS (?) {x}

The PHAS command sets or queries the reference phase shift. The parameter x is the phase (real number of degrees). The PHAS x command will set the phase shift to x. The value of x will be rounded to 0.01°. The phase may be programmed from -360.00 ≤ x ≤ 729.99 and will be wrapped around at ±180°. For example, the PHAS 541.0 command will set the phase to -179.00° (541-360=181=-179). The PHAS? queries the phase shift.

FMOD (?) {i}

The FMOD command sets or queries the reference source. The parameter i selects internal (i=1) or external (i=0).

FREQ (?) {f}

The FREQ command sets or queries the reference frequency. The FREQ? query command will return the reference frequency (in internal or external mode).

The FREQ f command sets the frequency of the internal oscillator. This command is allowed only if the reference source is internal. The parameter f is a frequency (real number of Hz). The value of f will be rounded to 5 digits or 0.0001 Hz, whichever is greater. The value of f is limited to 0.001 ≤ f ≤ 102000. If the harmonic number is greater than 1, then the frequency is limited to nxf ≤ 102 kHz where n is the harmonic number.

RSLP (?) {i}

The RSLP command sets or queries the reference trigger when using the external reference mode. The parameter i selects sine zero crossing (i=0), TTL rising edge (i=1), or TTL falling edge (i=2). At frequencies below 1 Hz, the TTL reference must be used.

HARM (?) {i}

The HARM command sets or queries the detection harmonic. This parameter is an integer from 1 to 19999. The HARM i command will set the lock-in to detect at the ith harmonic of the reference frequency. The value of i is limited by ixf ≤ 102 kHz. If the value of i requires a detection frequency greater than 102 kHz, then the harmonic number will be set to the largest value of i such that ixf ≤ 102 kHz.

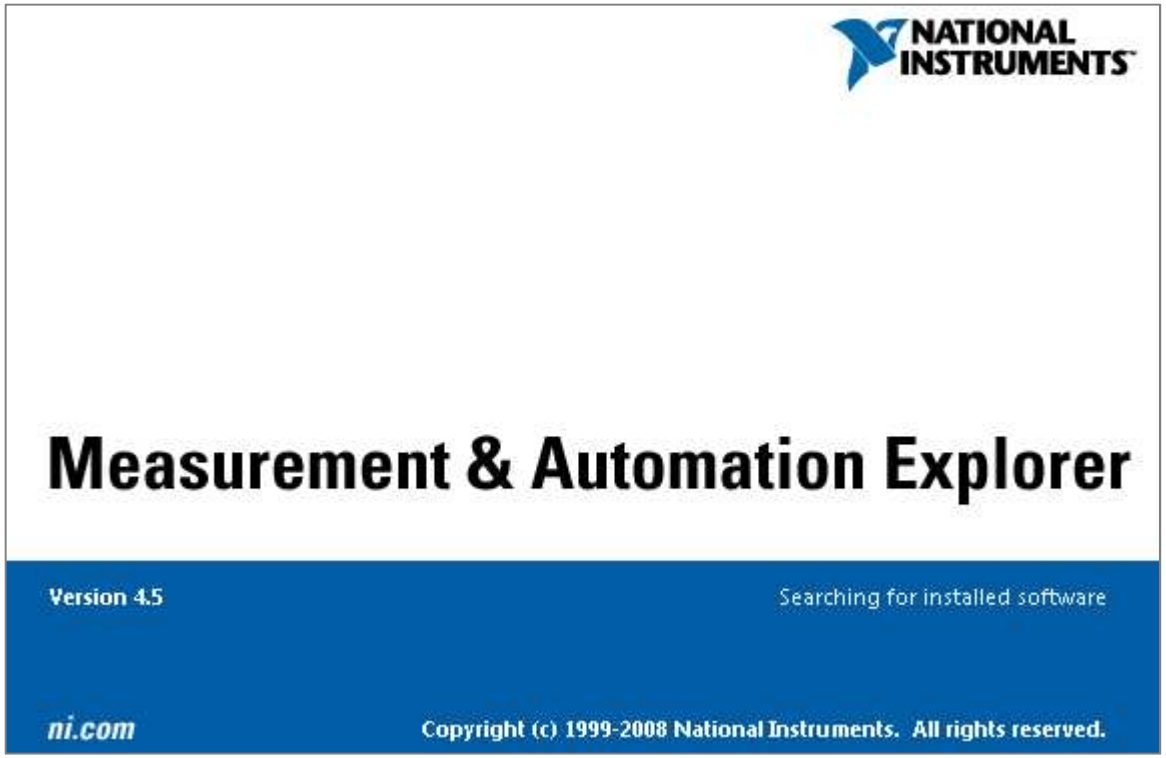
SLVL (?) {x}

The SLVL command sets or queries the amplitude of the sine output. The parameter x is a voltage (real number of Volts). The value of x will be rounded to 0.002V. The value of x is limited to 0.004 ≤ x ≤ 5.000.

- Of course understanding the command set has a prerequisite of understanding the instrument itself.
- Learn the instrument
- Learn what to do, without the computer
- Apply the scheme to LabVIEW

Connection

- You can check the status of the instrument by employing Measurement and Automation Explorer



Instrument I/O

- There is a general scheme of instrument interfacing:

0) Handshake

1) Busy?

2) No

3) Write “FREQ?” to buffer

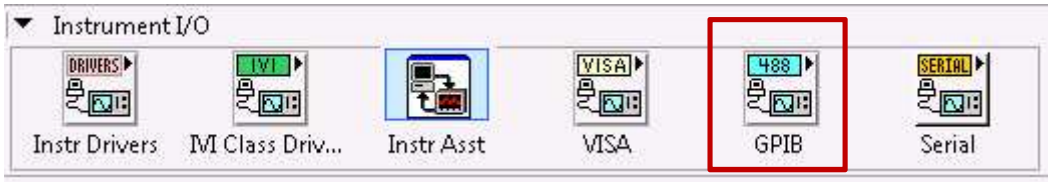
4) Poll

5) Ready

6) Read Buffer

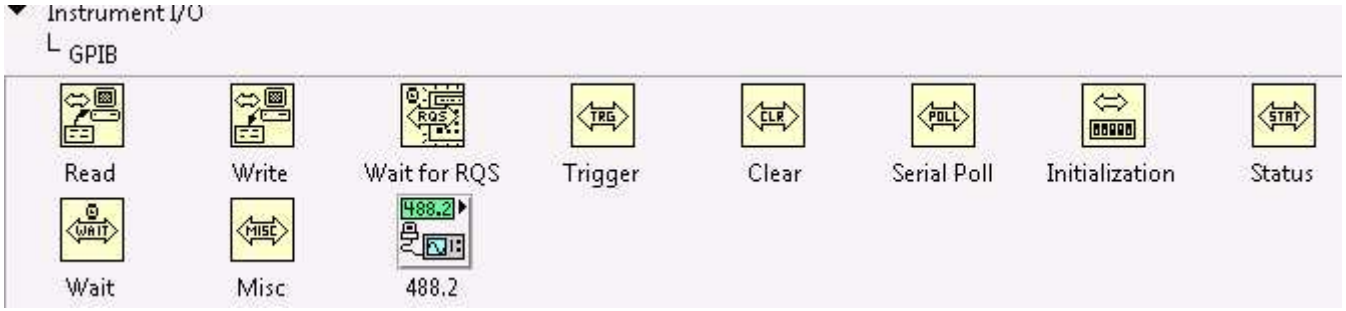
Instrument I/O

- Seek Instrument I/O palette
- Choose an appropriate way from many



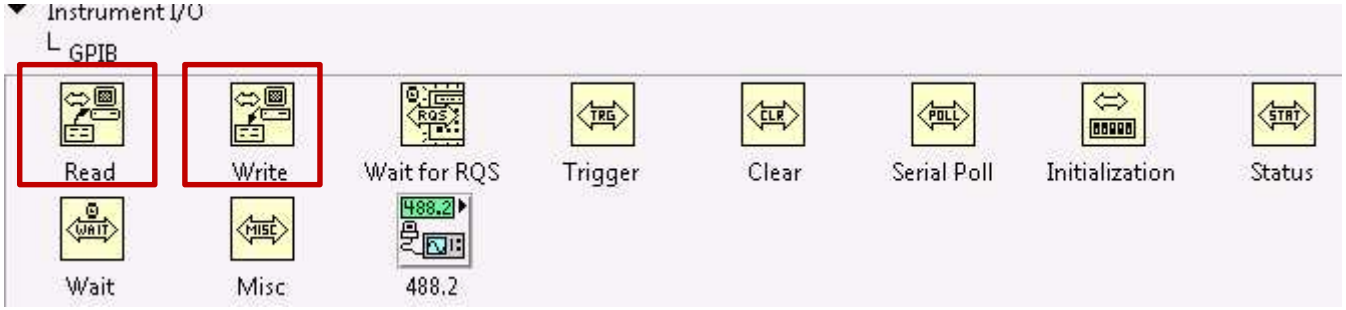
Instrument I/O

- All actions that are needed can be found under that palette.
- You can do more advance stuff by invoking the advanced palette (488.2 in this case)



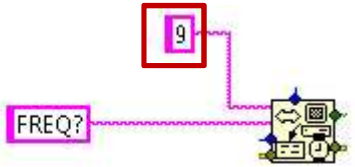
Instrument I/O

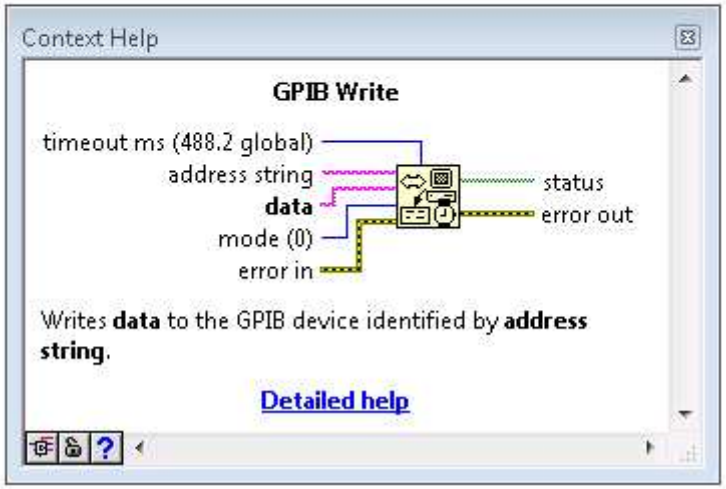
- For modern instruments read and write actions are enough



Instrument I/O

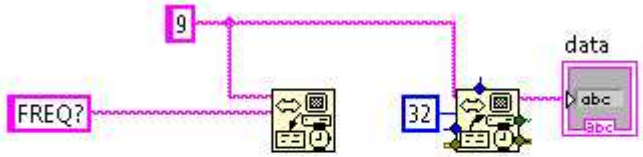
- To send a command to the instrument use Write action
- You need the GPIB address of the instrument

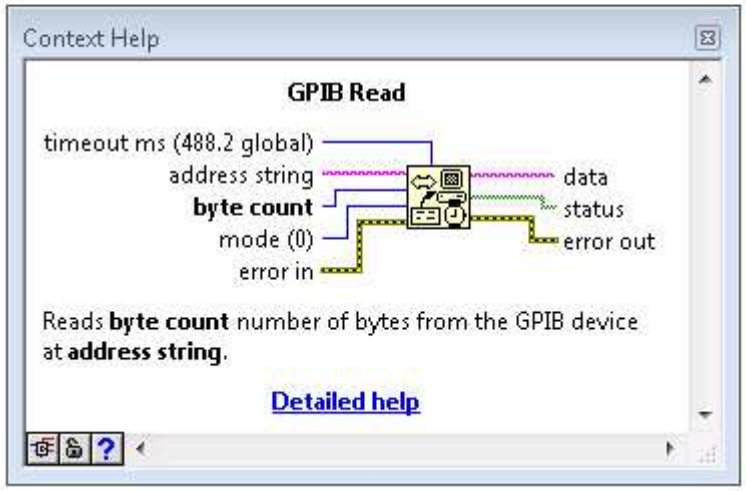




Instrument I/O

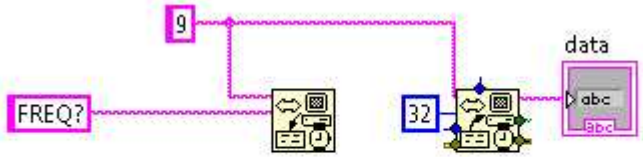
- To get the result of your command use Read action

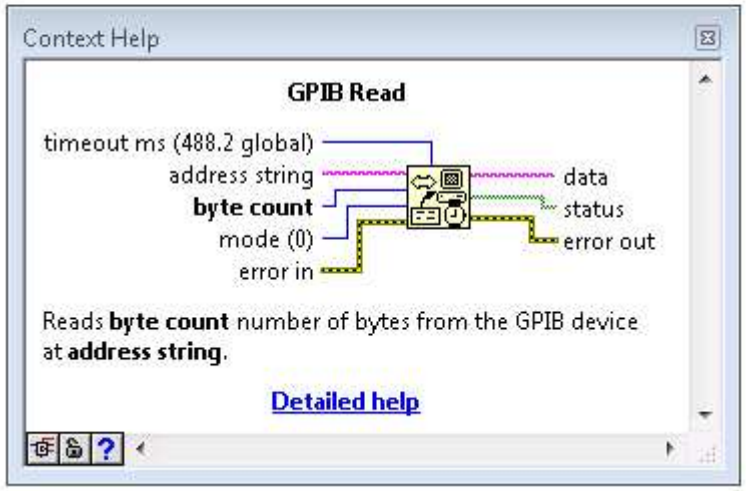




Instrument I/O

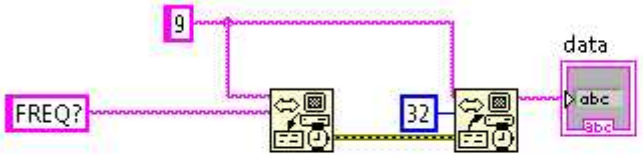
- There is an error in this code. Can you guess?

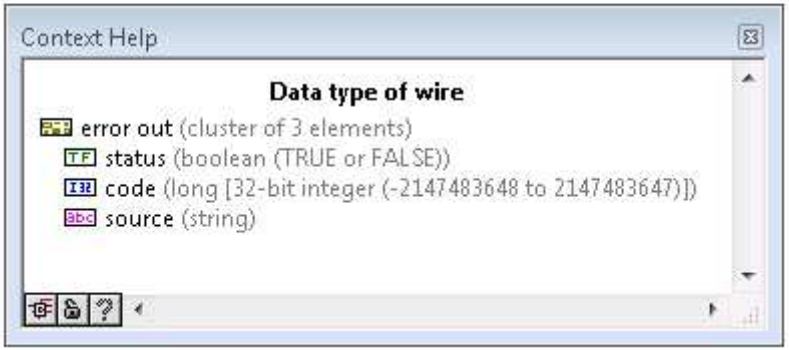




Instrument I/O

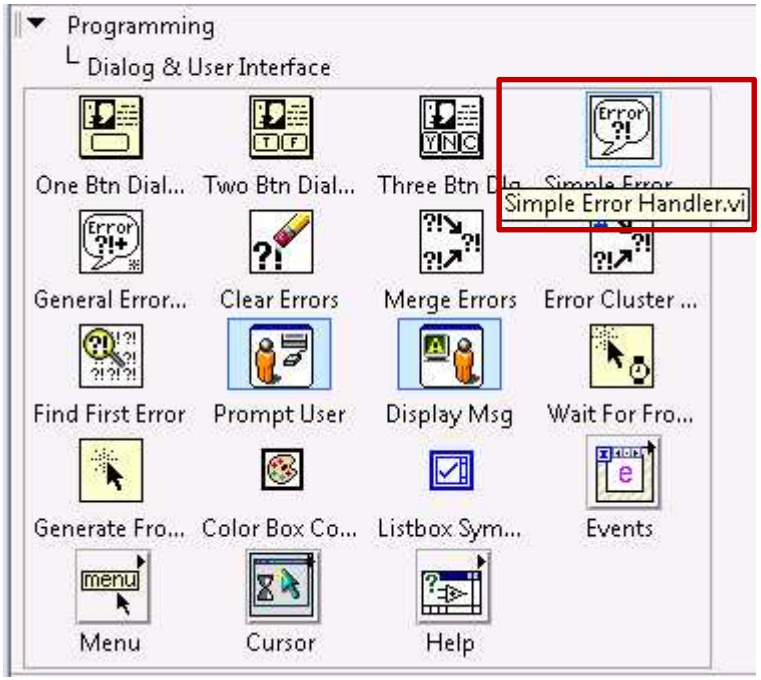
- Much better



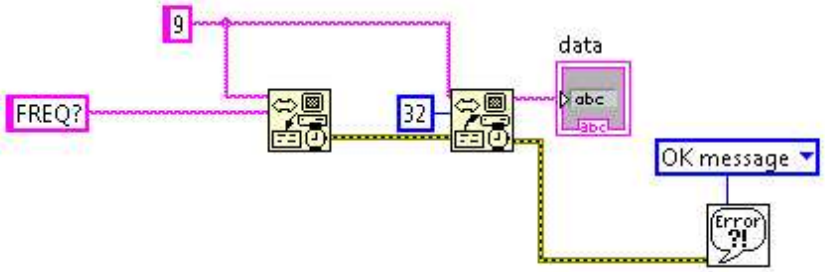


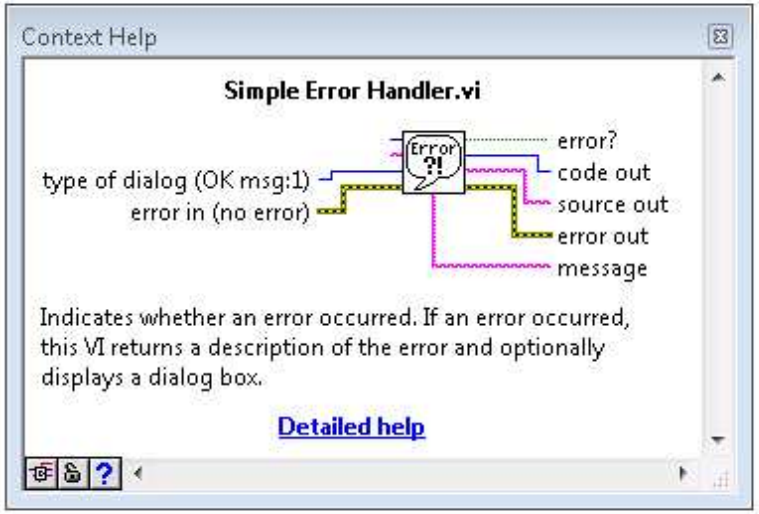
Error Handling

- Simple error handling can be easily done by employing Simple Error Handler



Error Handling





Error Handling

- You may not want to be bothered by some timeout error