Graphical Programming

LabVIEW Programming Fundamentals

Hans-Petter Halvorsen

LabVIEW

- LabVIEW is a Graphical Programming Language from NI/Emerson
- Professional paid version or free Community version for personal, non-commercial, non-academic and non-industrial purposes
- https://www.ni.com

LabVIEW Topics

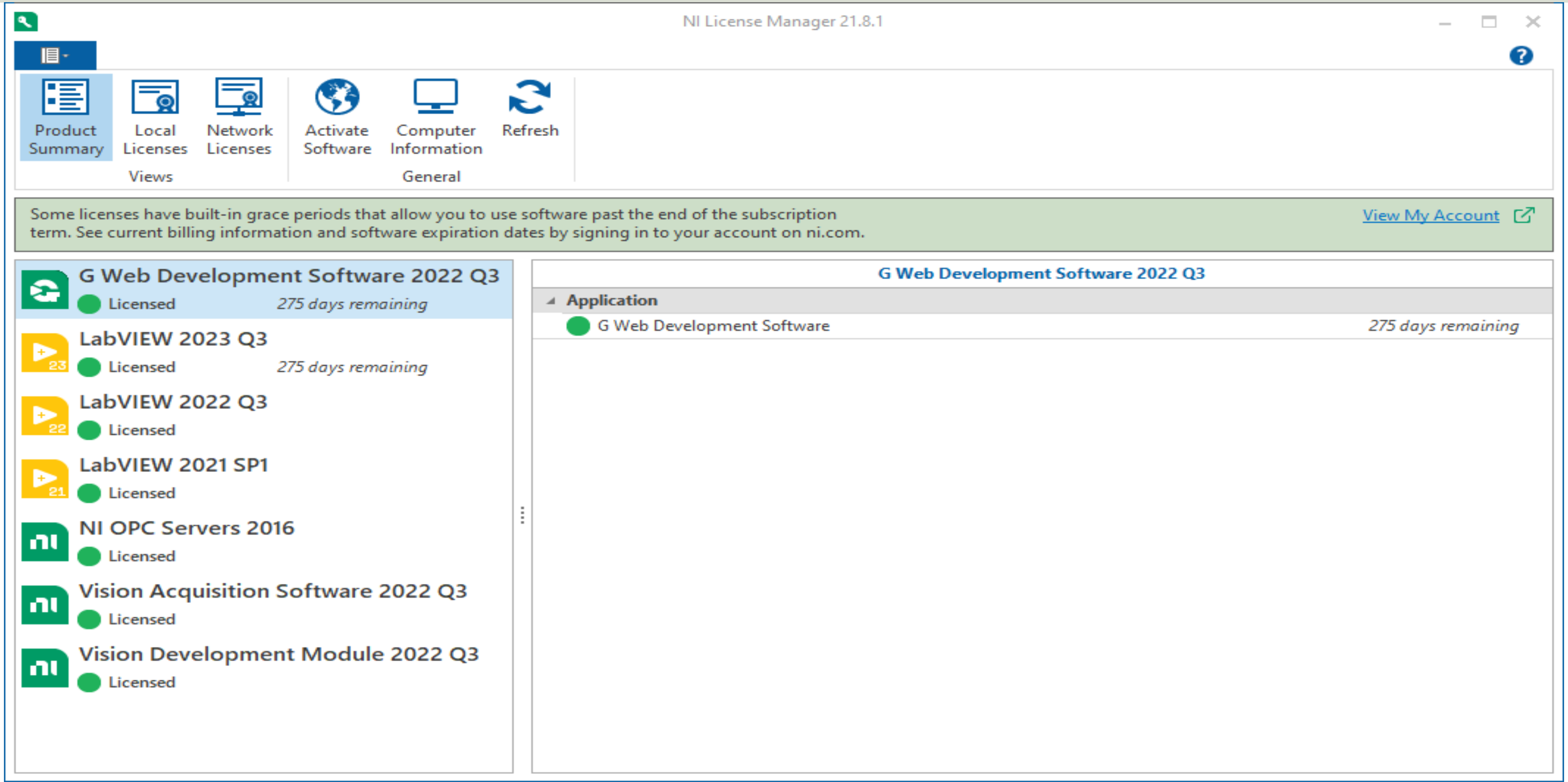
https://www.halvorsen.blog

LabVIEW Fundamentals

LabVIEW Programming Environment

Hans-Petter Halvorsen

NI License Manager



Some licenses have built-in grace periods that allow you to use software past the end of the subscription term. See current billing information and software expiration dates by signing in to your account on ni.com.

View My Account

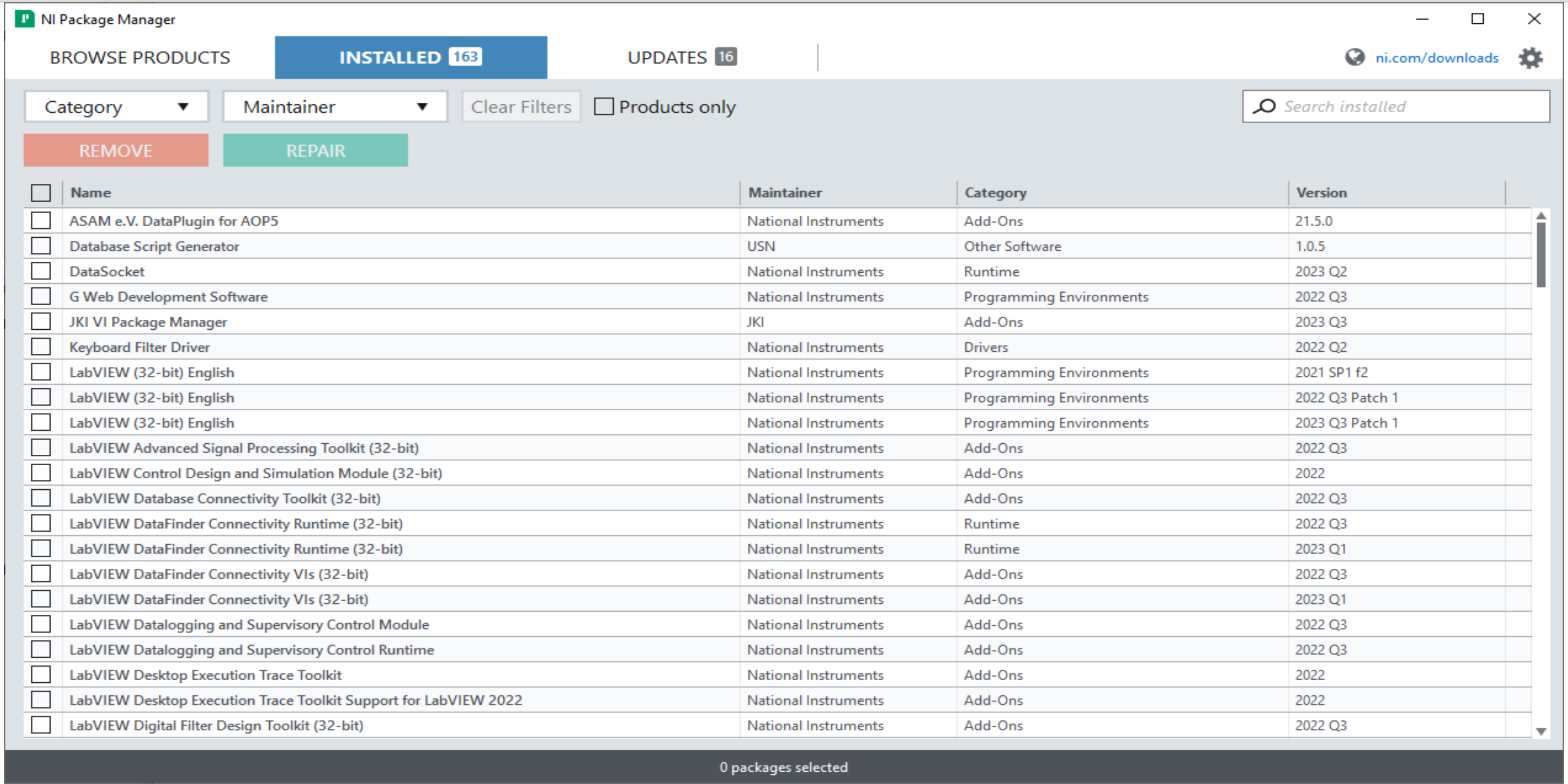
G Web Development Software 2022 Q3	● Licensed	275 days remaining
LabVIEW 2023 Q3	● Licensed	275 days remaining
LabVIEW 2022 Q3	● Licensed	
LabVIEW 2021 SP1	● Licensed	
NI OPC Servers 2016	● Licensed	
Vision Acquisition Software 2022 Q3	● Licensed	
Vision Development Module 2022 Q3	● Licensed	

G Web Development Software 2022 Q3

Application

● G Web Development Software	275 days remaining

NI Package Manager

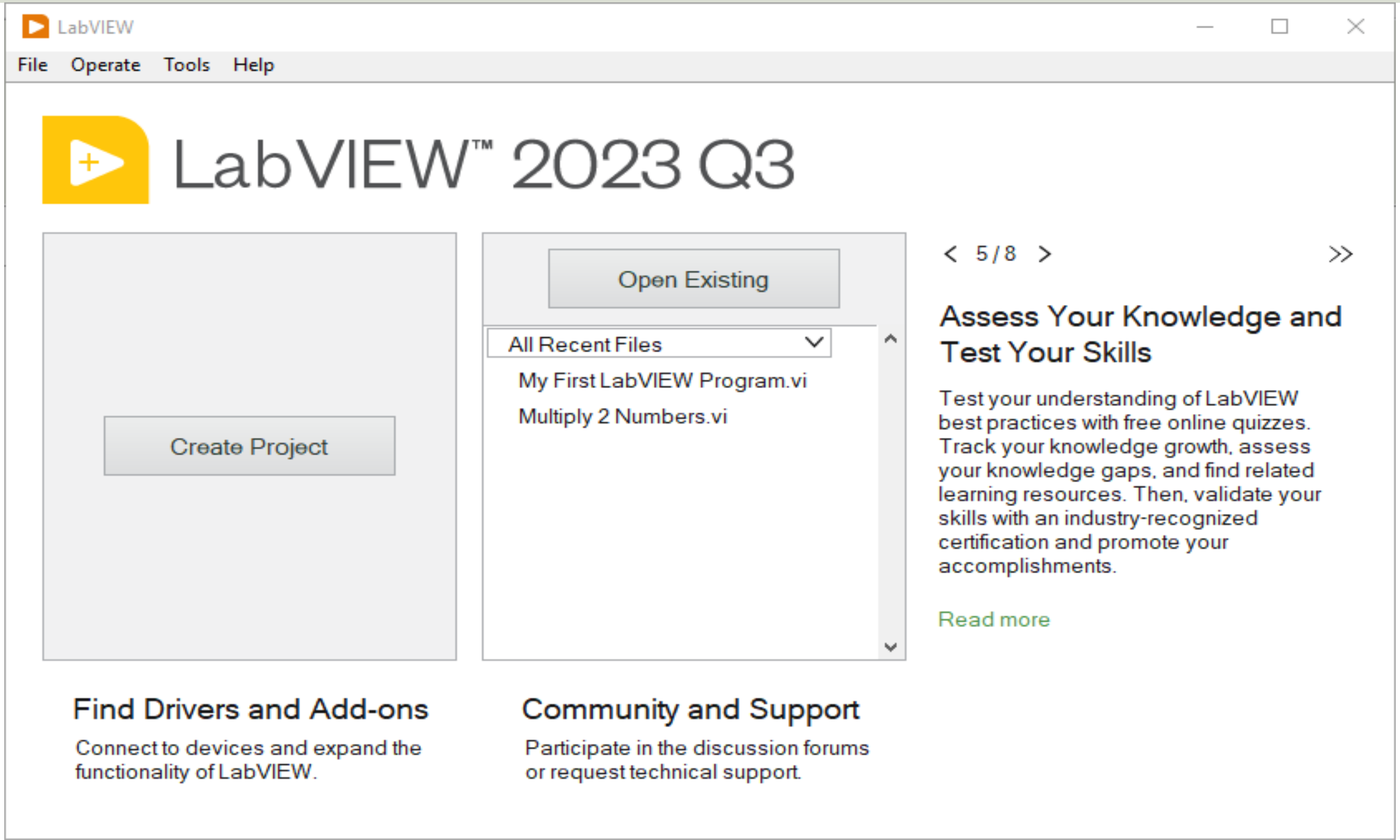


NI Package Manager

Name	Maintainer	Category	Version
ASAM e.V. DataPlugin for AOP5	National Instruments	Add-Ons	21.5.0
Database Script Generator	USN	Other Software	1.0.5
DataSocket	National Instruments	Runtime	2023 Q2
G Web Development Software	National Instruments	Programming Environments	2022 Q3
JKI VI Package Manager	JKI	Add-Ons	2023 Q3
Keyboard Filter Driver	National Instruments	Drivers	2022 Q2
LabVIEW (32-bit) English	National Instruments	Programming Environments	2021 SP1 f2
LabVIEW (32-bit) English	National Instruments	Programming Environments	2022 Q3 Patch 1
LabVIEW (32-bit) English	National Instruments	Programming Environments	2023 Q3 Patch 1
LabVIEW Advanced Signal Processing Toolkit (32-bit)	National Instruments	Add-Ons	2022 Q3
LabVIEW Control Design and Simulation Module (32-bit)	National Instruments	Add-Ons	2022
LabVIEW Database Connectivity Toolkit (32-bit)	National Instruments	Add-Ons	2022 Q3
LabVIEW DataFinder Connectivity Runtime (32-bit)	National Instruments	Runtime	2022 Q3
LabVIEW DataFinder Connectivity Runtime (32-bit)	National Instruments	Runtime	2023 Q1
LabVIEW DataFinder Connectivity VIs (32-bit)	National Instruments	Add-Ons	2022 Q3
LabVIEW DataFinder Connectivity VIs (32-bit)	National Instruments	Add-Ons	2023 Q1
LabVIEW Datalogging and Supervisory Control Module	National Instruments	Add-Ons	2022 Q3
LabVIEW Datalogging and Supervisory Control Runtime	National Instruments	Add-Ons	2022 Q3
LabVIEW Desktop Execution Trace Toolkit	National Instruments	Add-Ons	2022
LabVIEW Desktop Execution Trace Toolkit Support for LabVIEW 2022	National Instruments	Add-Ons	2022
LabVIEW Digital Filter Design Toolkit (32-bit)	National Instruments	Add-Ons	2022 Q3

0 packages selected

LabVIEW



Assess Your Knowledge and Test Your Skills

Test your understanding of LabVIEW best practices with free online quizzes. Track your knowledge growth, assess your knowledge gaps, and find related learning resources. Then, validate your skills with an industry-recognized certification and promote your accomplishments.

Read more

Find Drivers and Add-ons

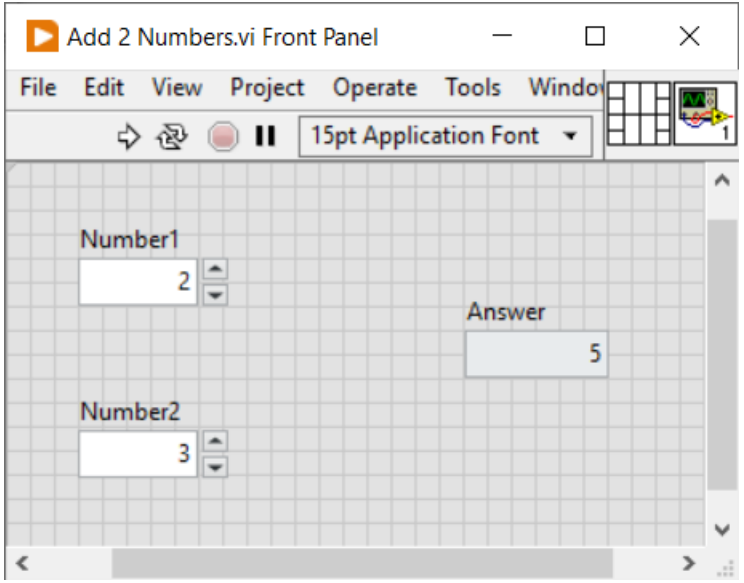
Connect to devices and expand the functionality of LabVIEW.

Community and Support

Participate in the discussion forums or request technical support.

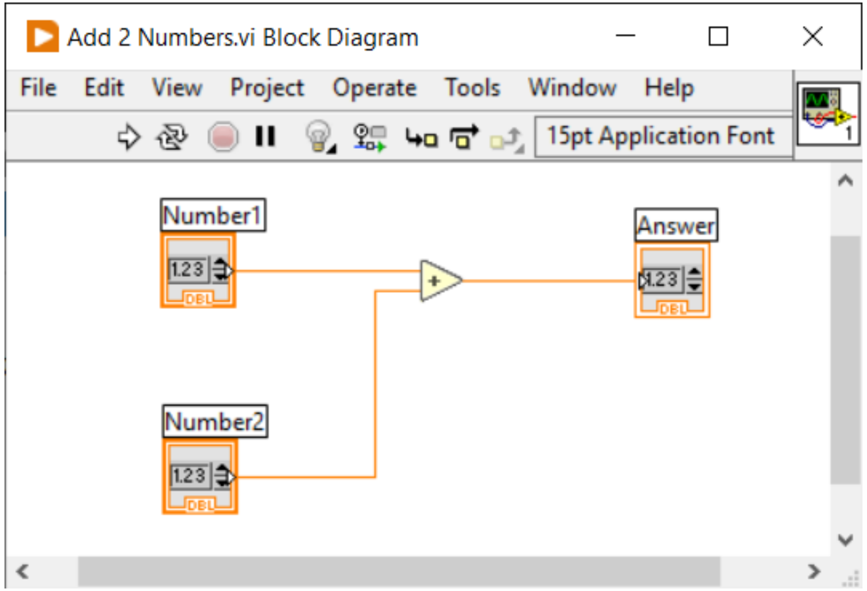
Front Panel and Block Diagram

Block Diagram



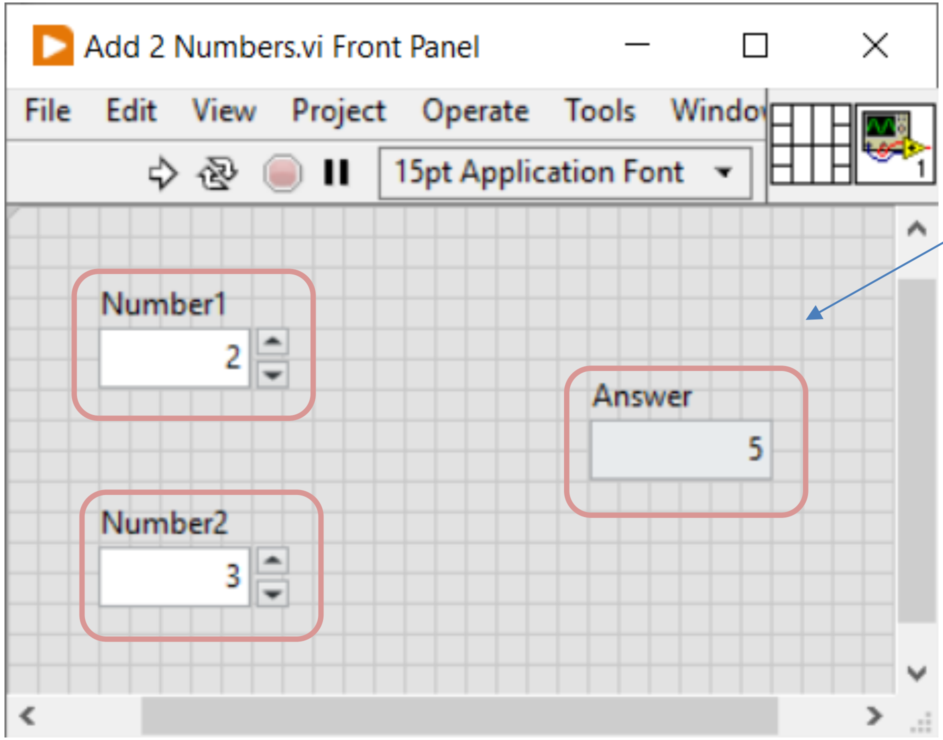
Add 2 Numbers.vi Front Panel
File Edit View Project Operate Tools Windo
15pt Application Font
Number1
2
Number2
3
Answer
5

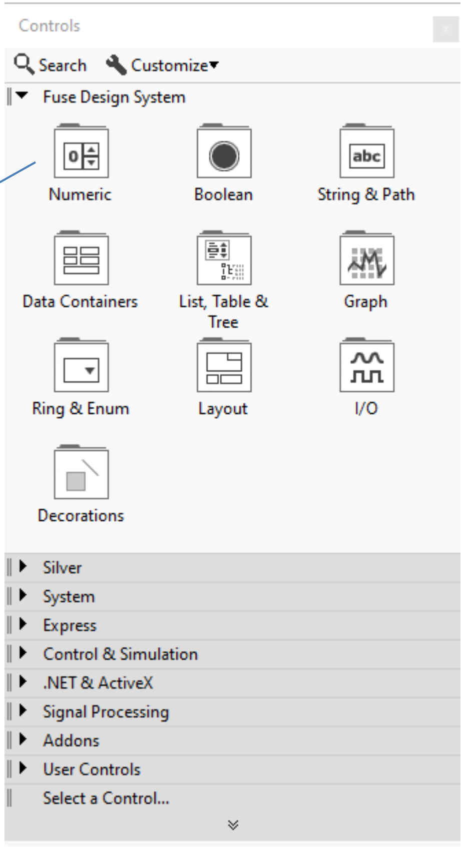
Front Panel



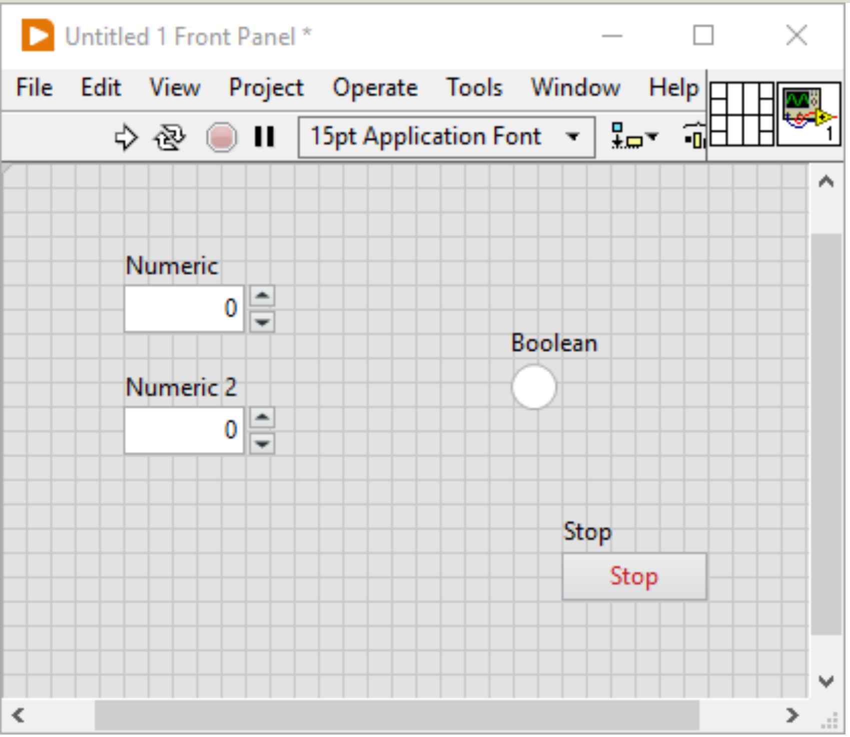
Add 2 Numbers.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Number1
1.23
DBL
Number2
1.23
DBL
Answer
1.23
DBL

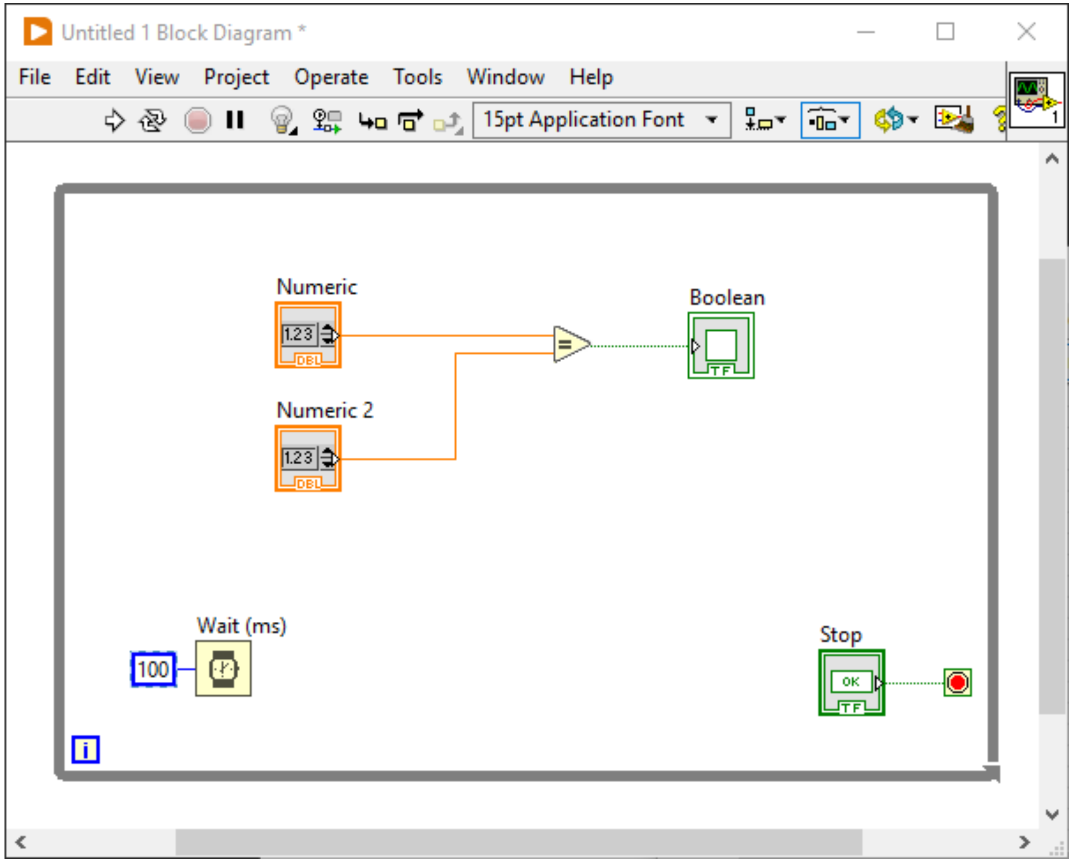
Front Panel and Controls



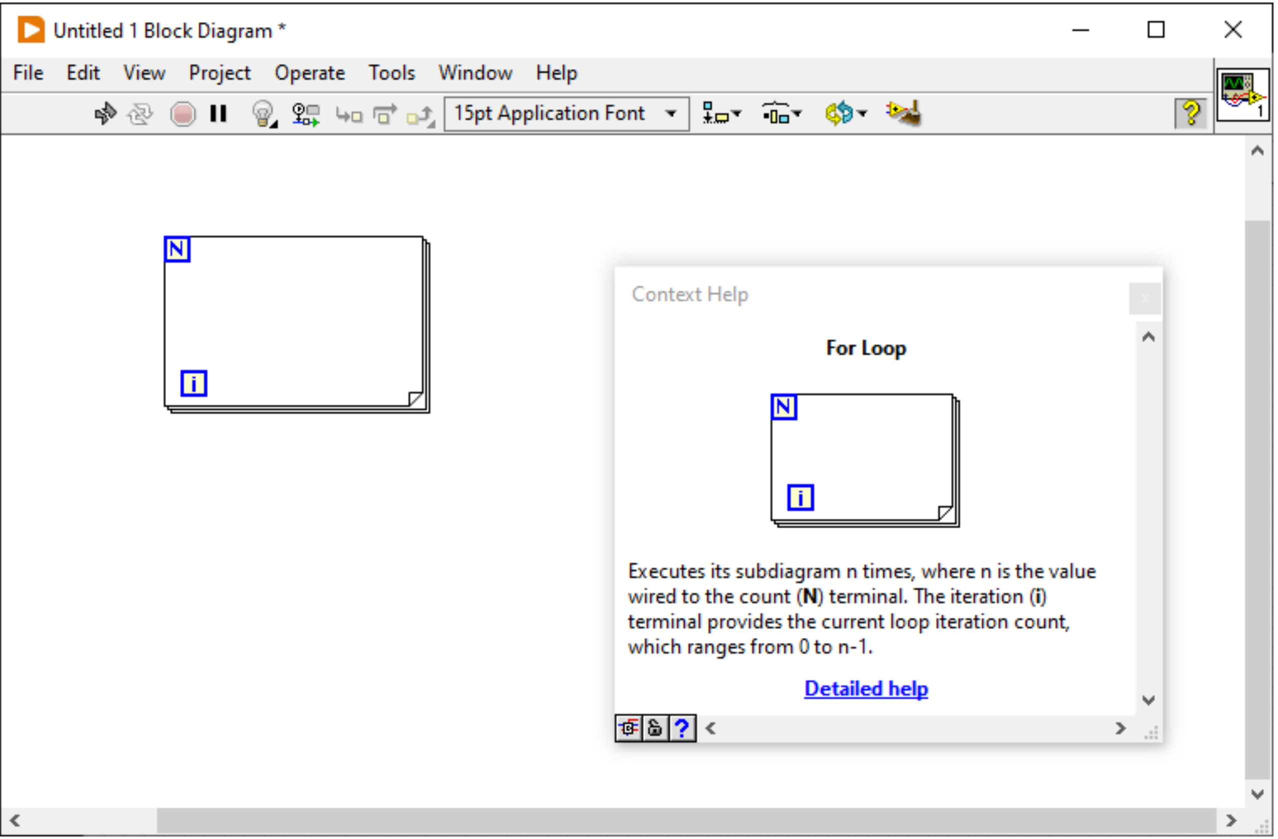


Front Panel and Block Diagram





Getting Help -> Ctrl + H



Untitled 1 Block Diagram *
File Edit View Project Operate Tools Window Help
15pt Application Font
Context Help
N
i

For Loop

Executes its subdiagram n times, where n is the value wired to the count (**N**) terminal. The iteration (**i**) terminal provides the current loop iteration count, which ranges from 0 to n-1.

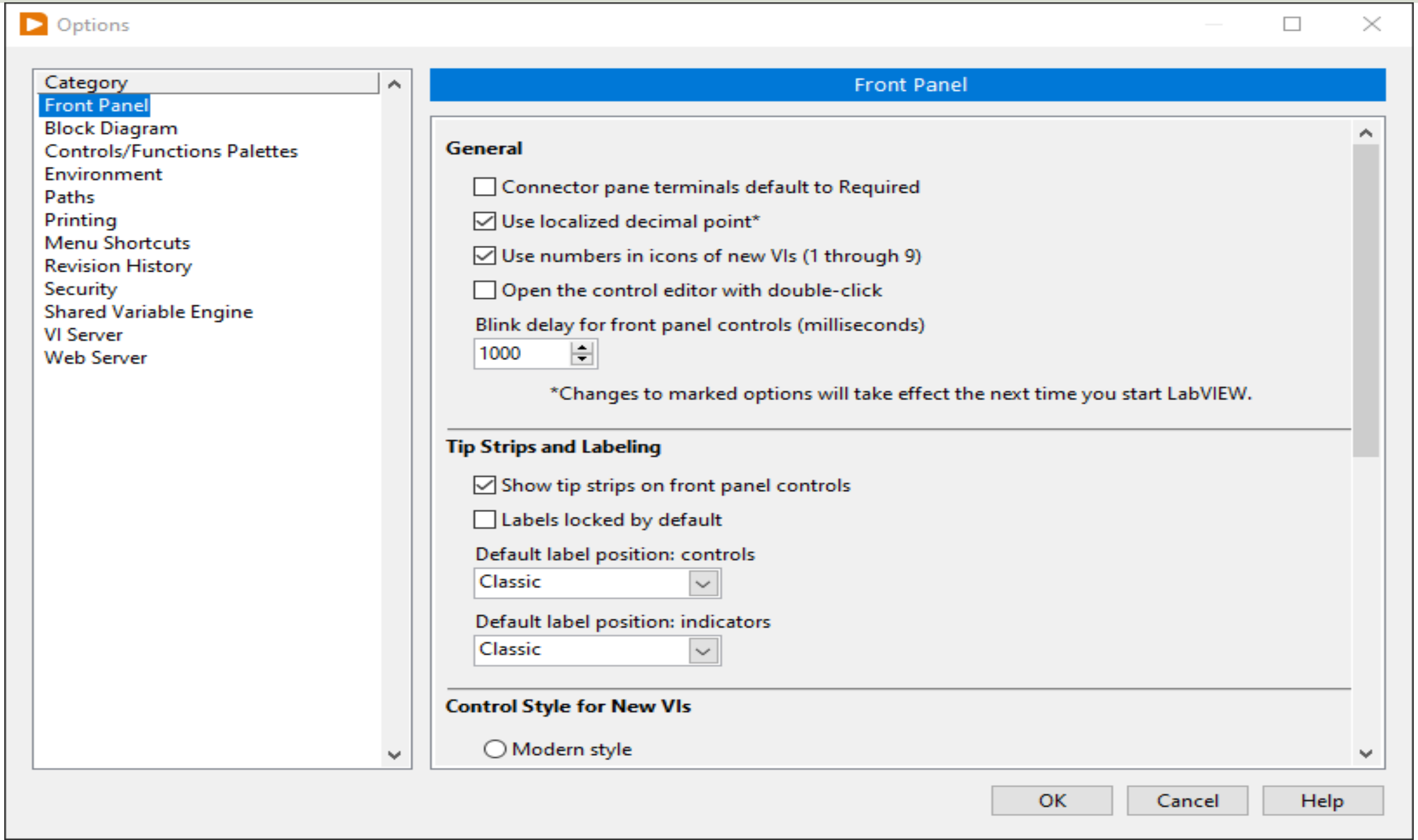
Detailed help

If you click Ctrl + H, then the "Context Help" window will appear.

Then you can click on different items to get specific help

Options/Settings

Tools -> Options…



Options

Category
- Front Panel
- Block Diagram
- Controls/Functions Palettes
- Environment
- Paths
- Printing
- Menu Shortcuts
- Revision History
- Security
- Shared Variable Engine
- VI Server
- Web Server

Front Panel

General

- [ ] Connector pane terminals default to Required
- [x] Use localized decimal point*
- [x] Use numbers in icons of new VIs (1 through 9)
- [ ] Open the control editor with double-click

Blink delay for front panel controls (milliseconds)
1000

*Changes to marked options will take effect the next time you start LabVIEW.

Tip Strips and Labeling

- [x] Show tip strips on front panel controls
- [ ] Labels locked by default

Default label position: controls
Classic

Default label position: indicators
Classic

Control Style for New VIs

○ Modern style

OK Cancel Help

Shortcuts

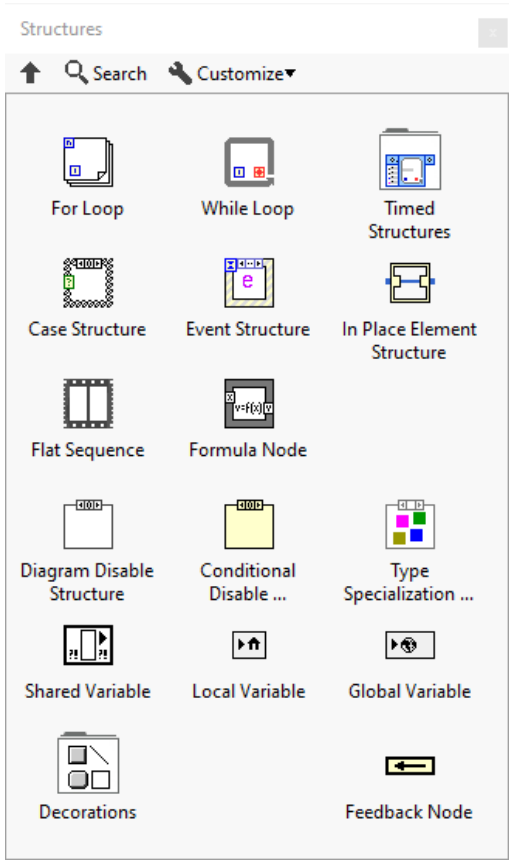
- **Ctrl + H** -> Getting Help
- **Ctrl + B** -> Remove “Bad or Broken” Wires
- **Ctrl + R** -> Run your Program/Current LabVIEW Application

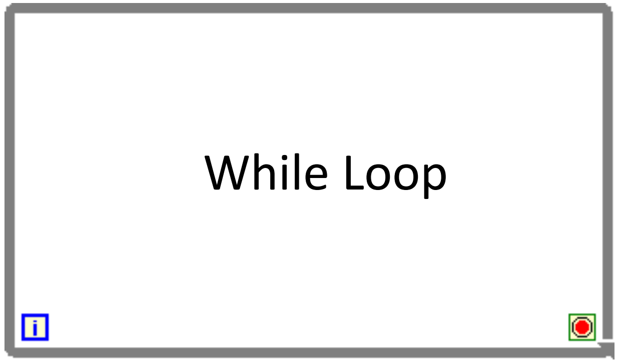
LabVIEW Fundamentals

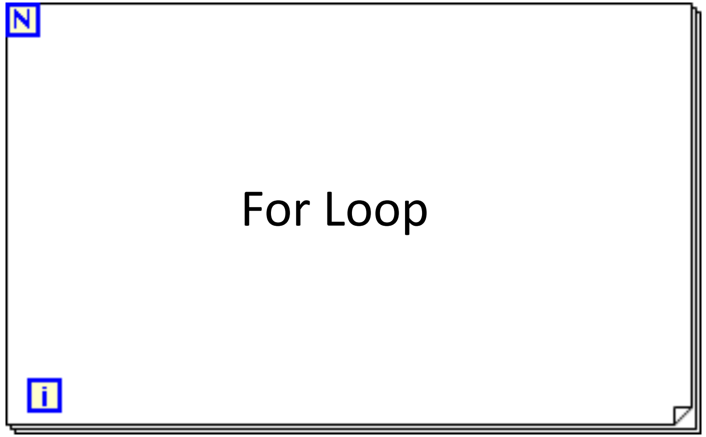
While Loops and For Loops in LabVIEW

Hans-Petter Halvorsen

While Loop and For Loop





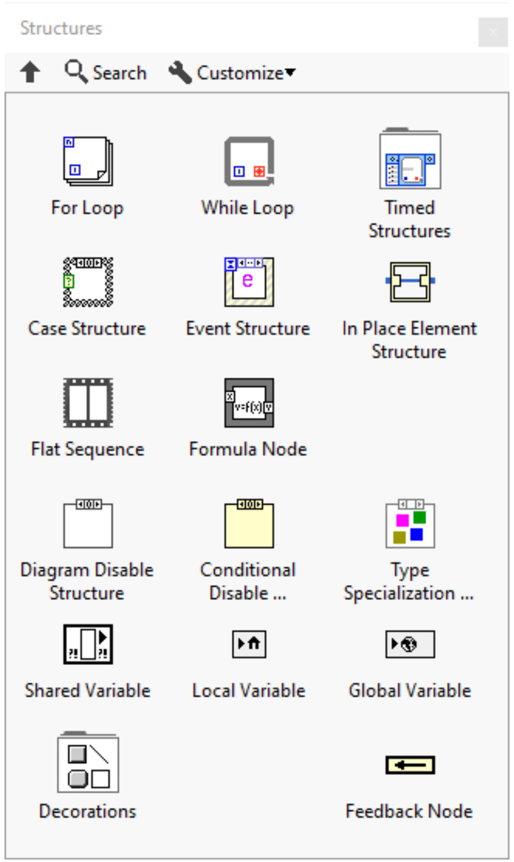


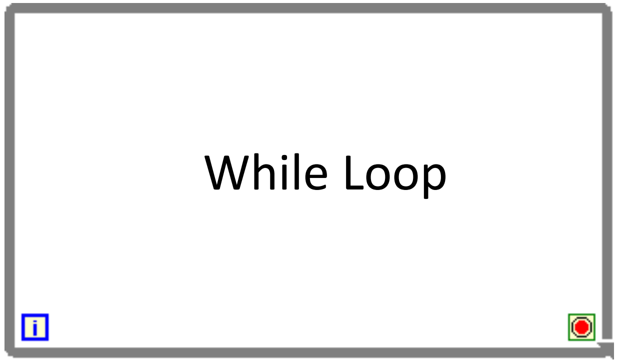
LabVIEW Fundamentals

While Loops in LabVIEW

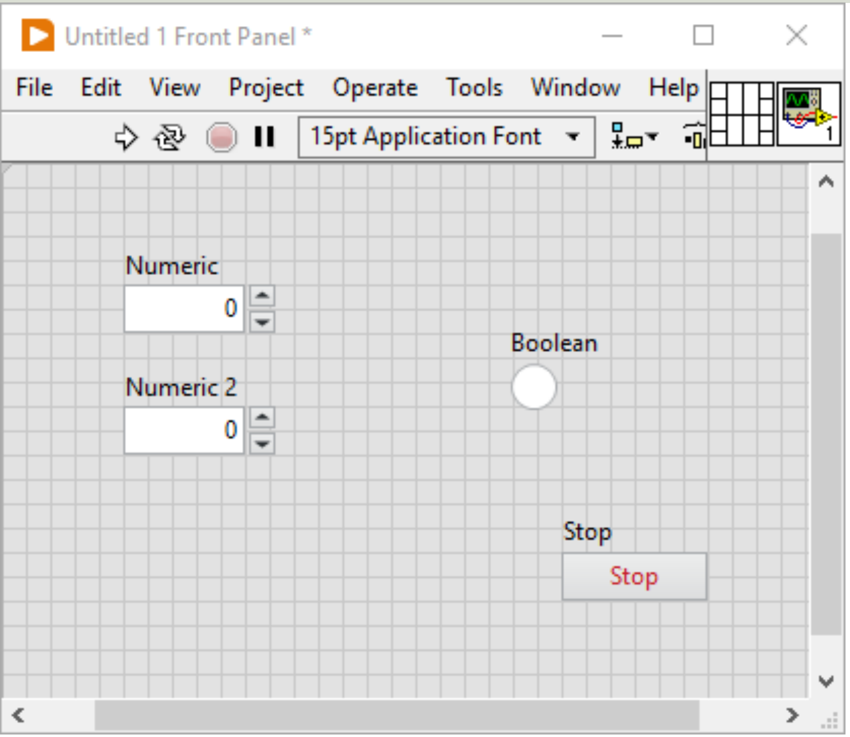
Hans-Petter Halvorsen

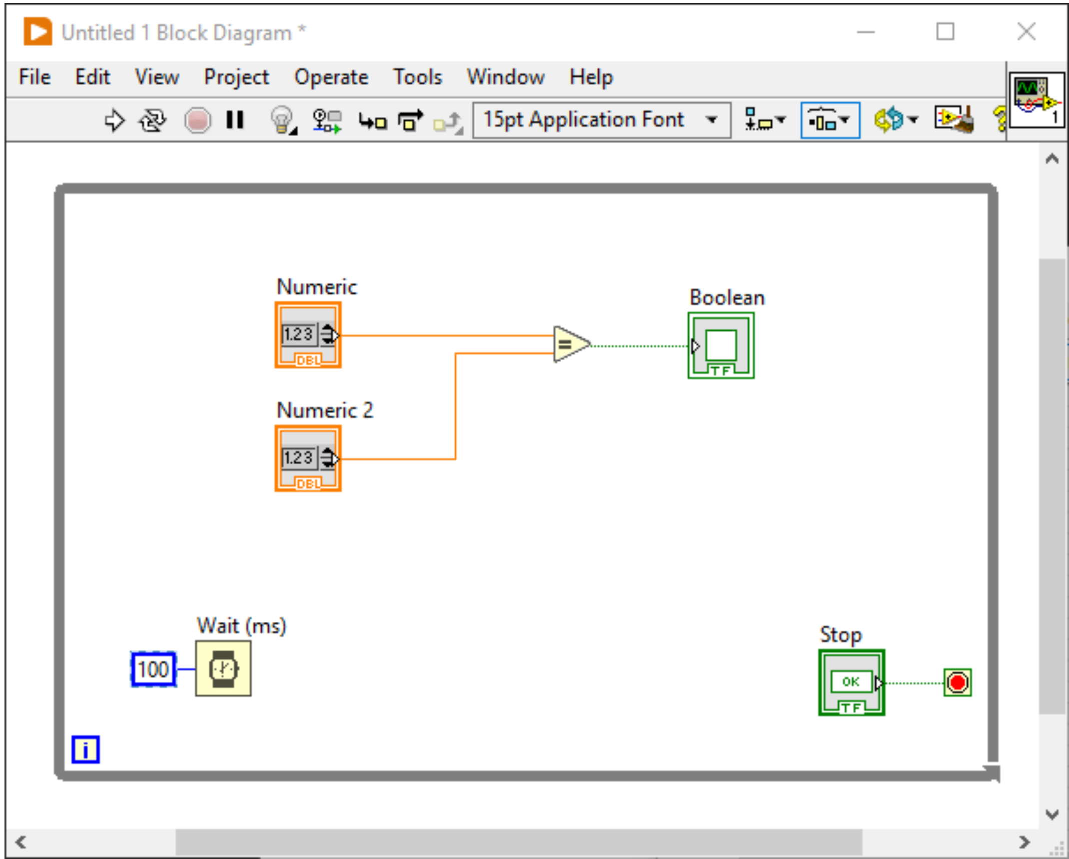
While Loop and For Loop



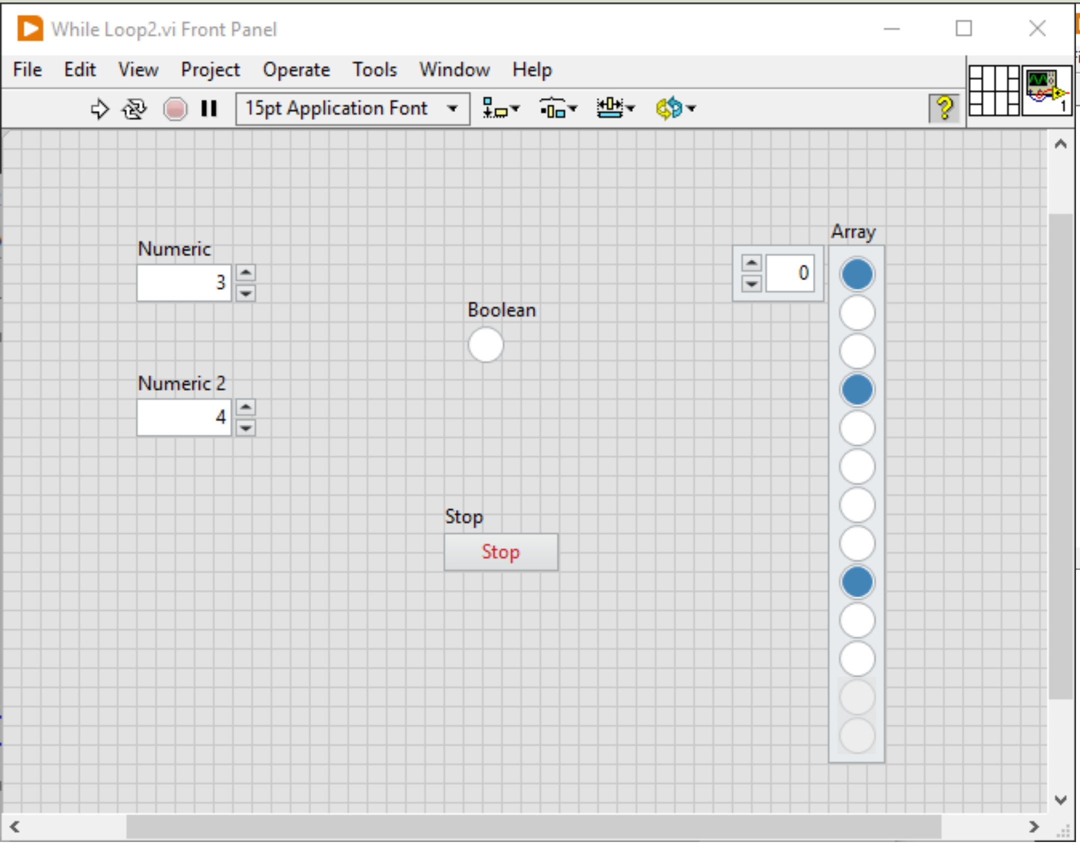


While Loop Example

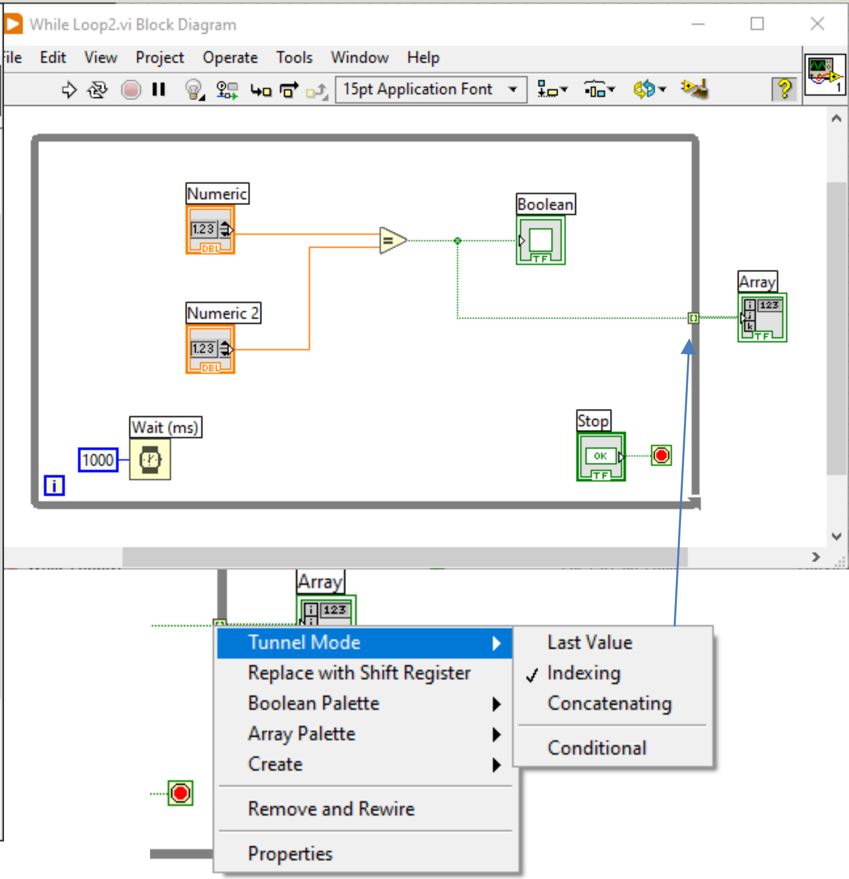




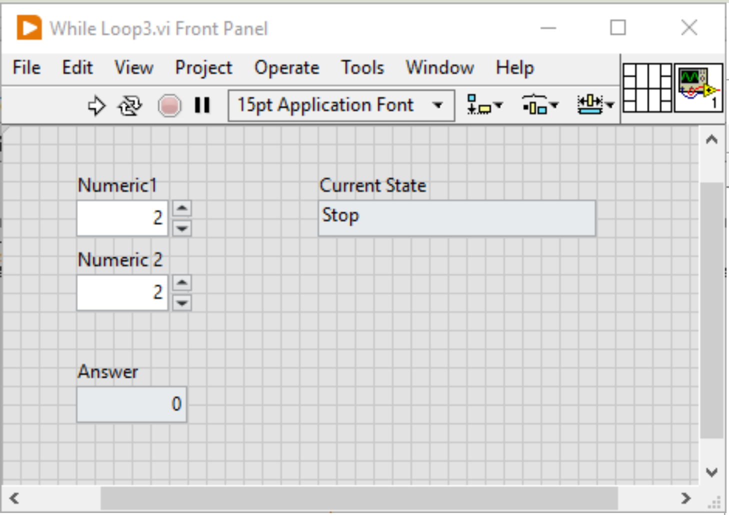
While Loop Example 2

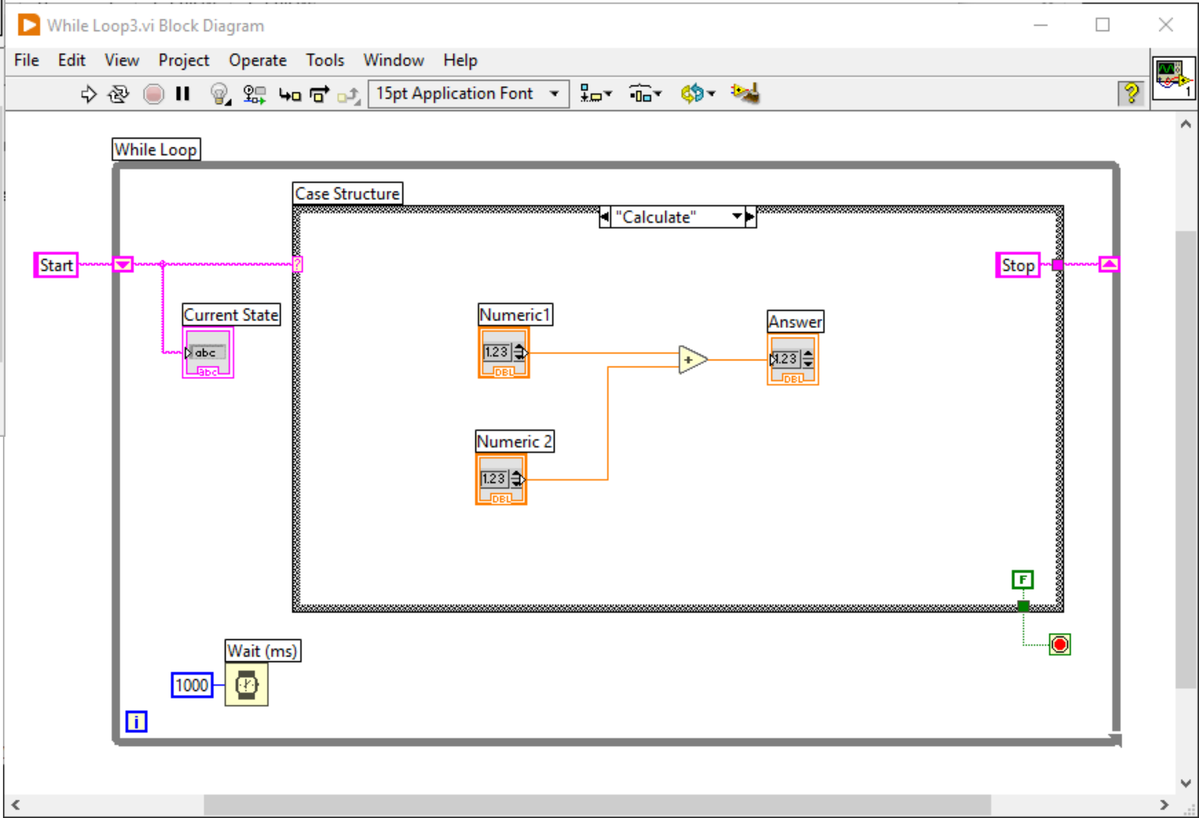


While Loop2.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Numeric
3
Numeric 2
4
Boolean
Array
0
Stop
Stop

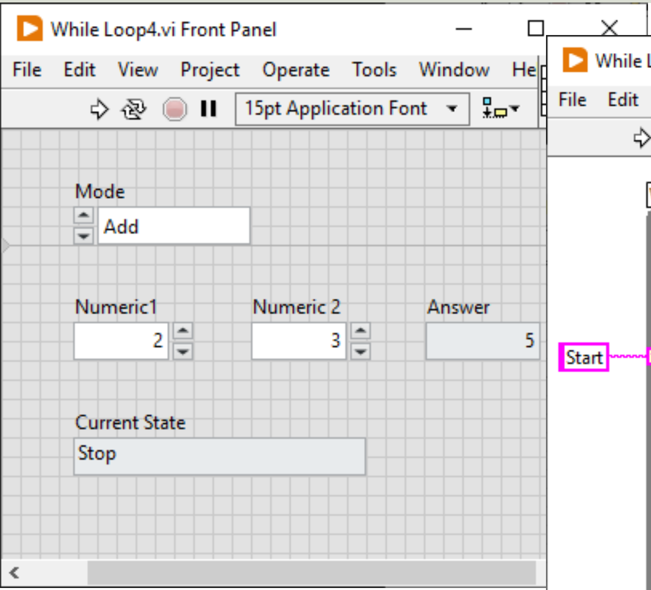


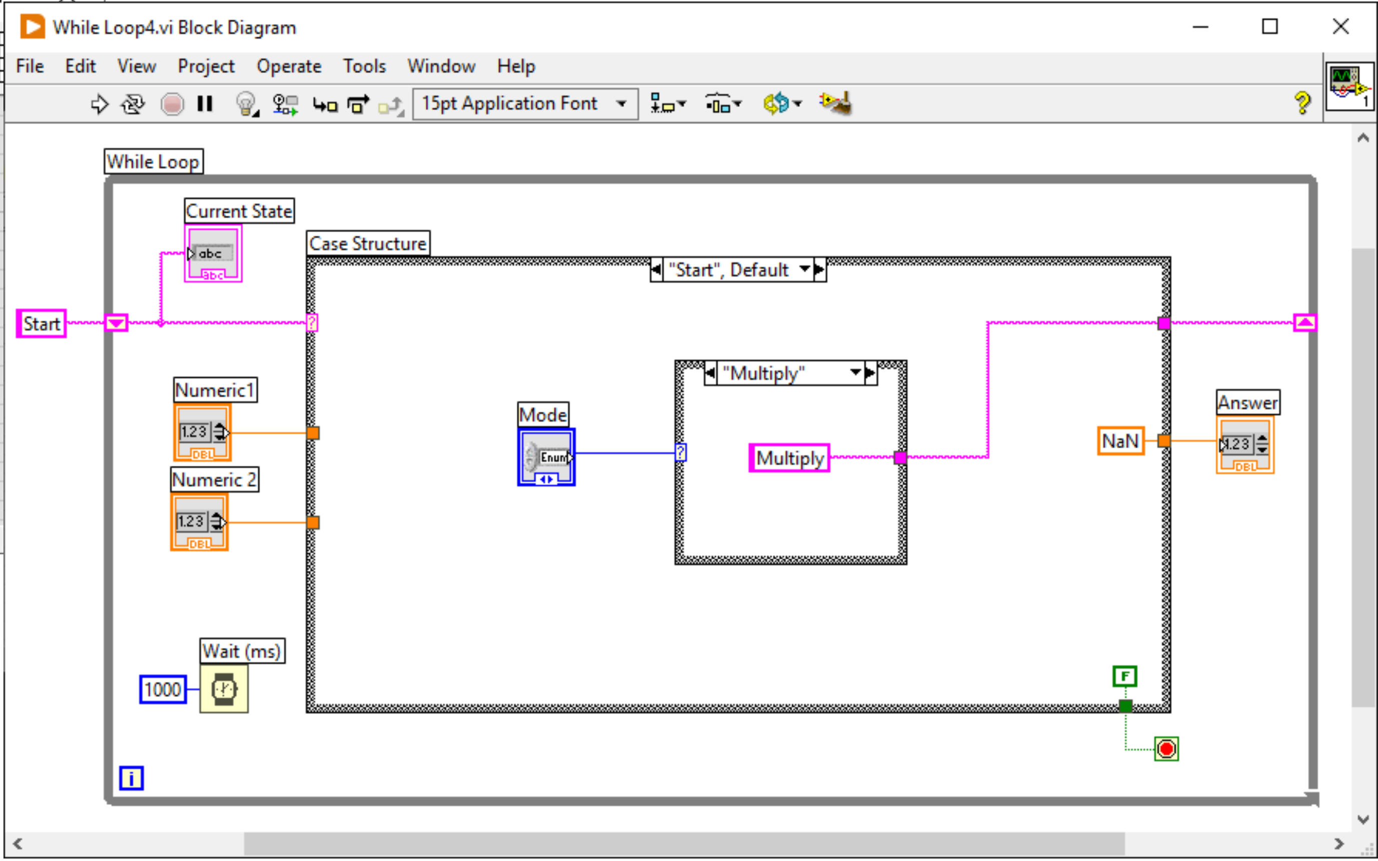
While Loop - Shift Register


While Loop3.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Numeric1
2
Numeric 2
2
Answer
0
Current State
Stop



While Loop – Basic Calculator


While Loop4.vi Front Panel
File Edit View Project Operate Tools Window He
15pt Application Font
Mode
Add
Numeric1
2
Numeric 2
3
Answer
5
Current State
Stop

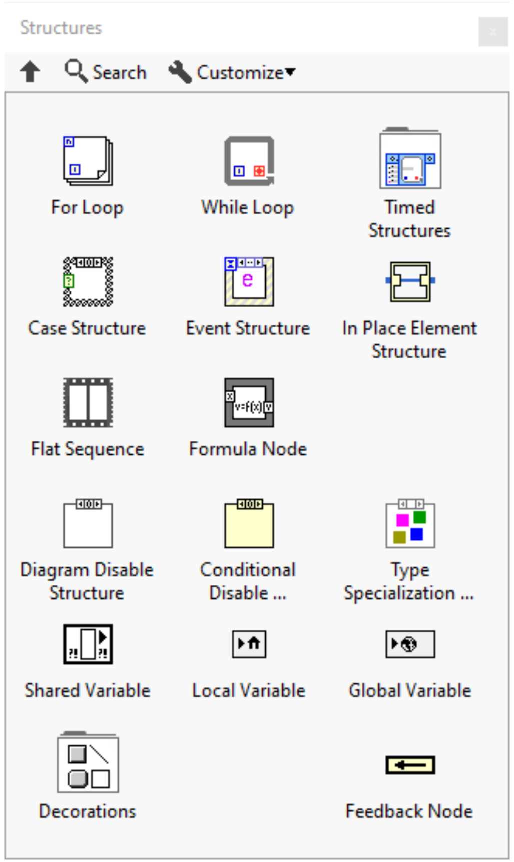


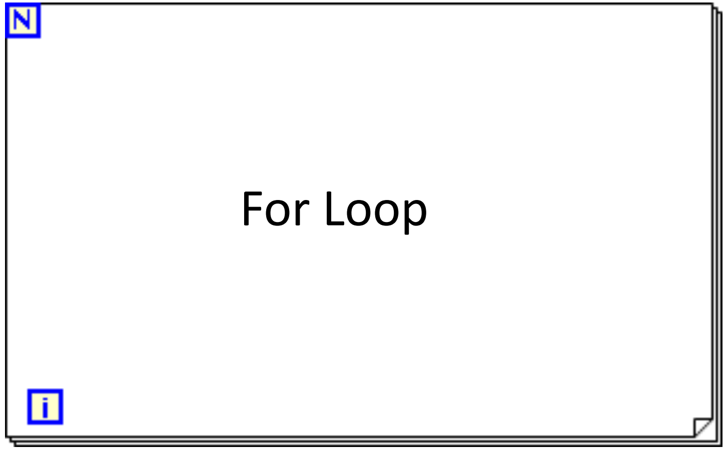
LabVIEW Fundamentals

For Loops in LabVIEW

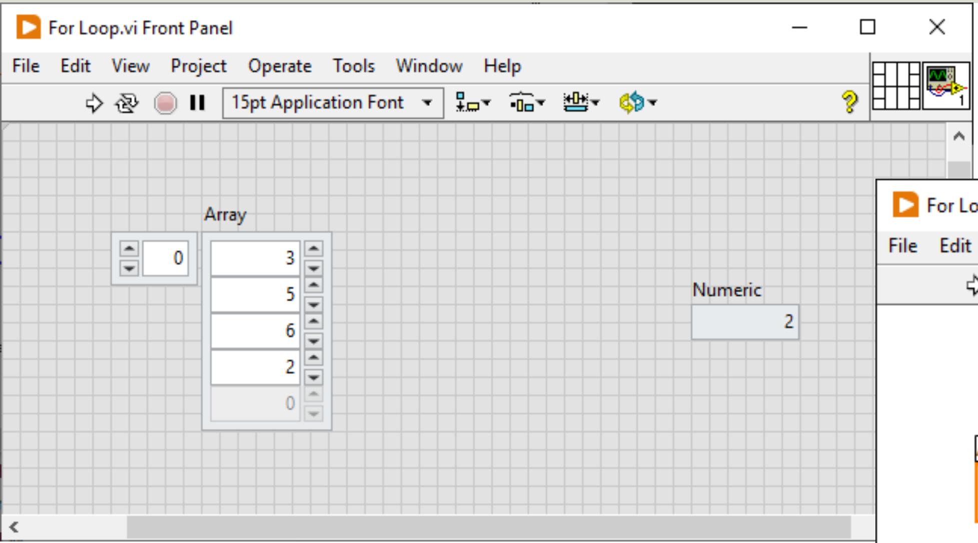
Hans-Petter Halvorsen

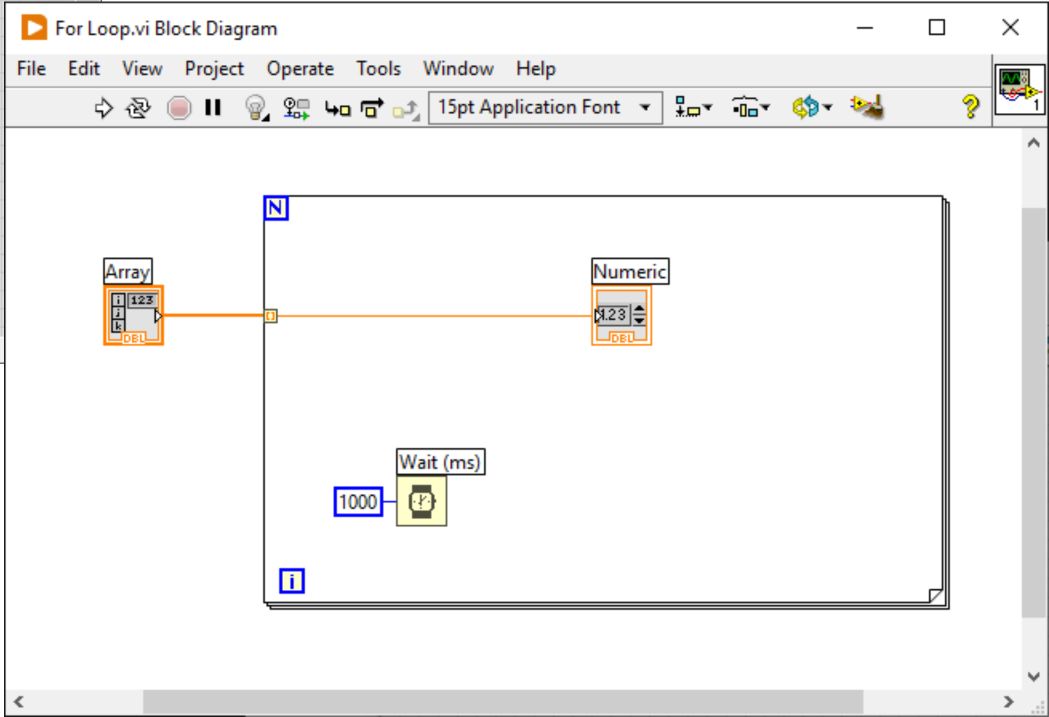
For Loop


Structures
Search
Customize
For Loop
While Loop
Timed Structures
Case Structure
Event Structure
In Place Element Structure
Flat Sequence
Formula Node
Diagram Disable Structure
Conditional Disable ...
Type Specialization ...
Shared Variable
Local Variable
Global Variable
Decorations
Feedback Node

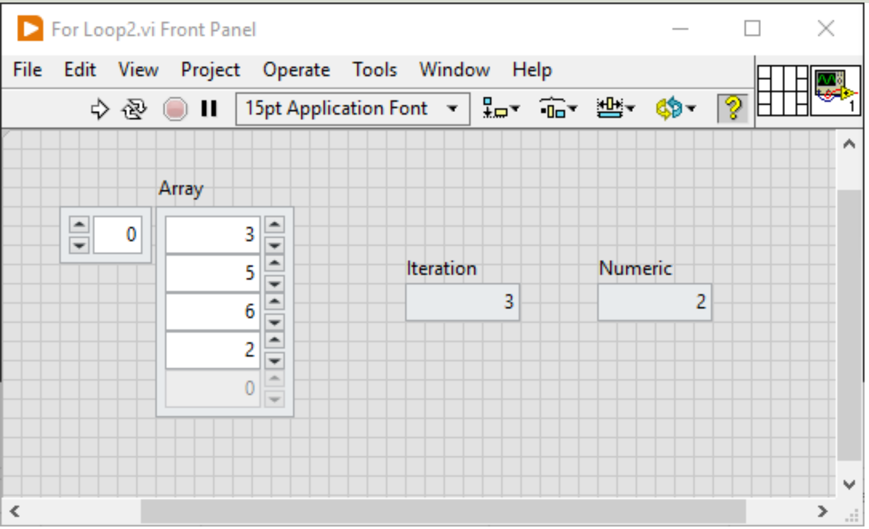

N
For Loop
i

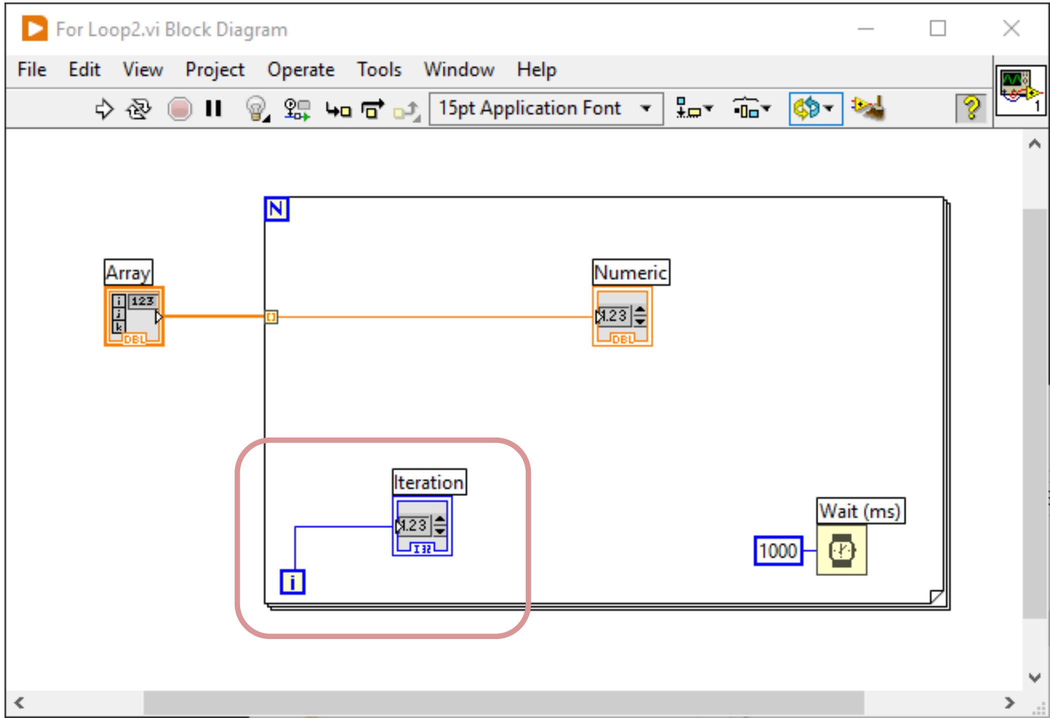
For Loop Example



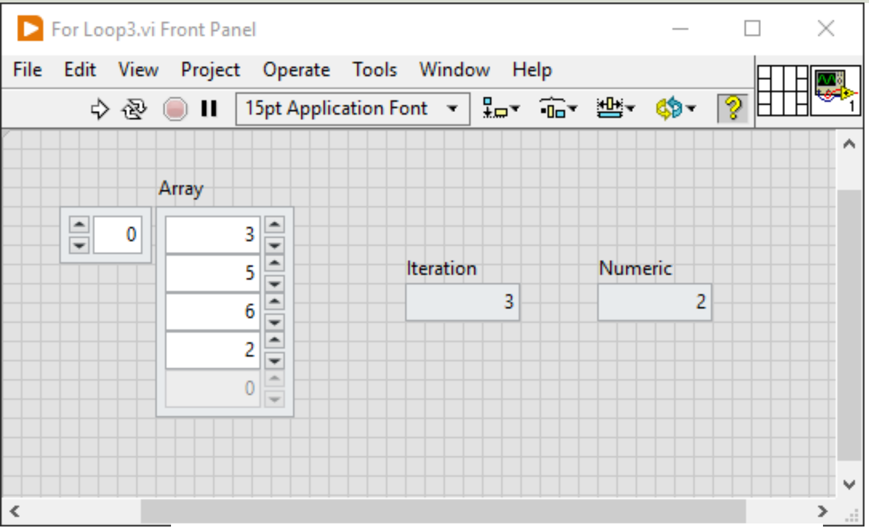

For Loop.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
N
Array
Numeric
Wait (ms)
1000
i

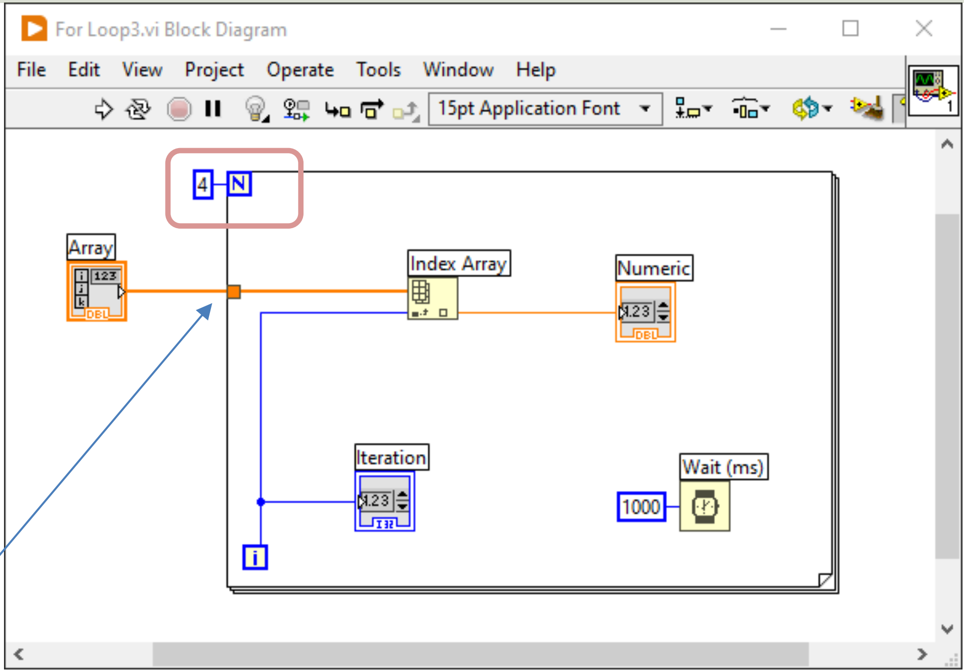
For Loop - Iterations

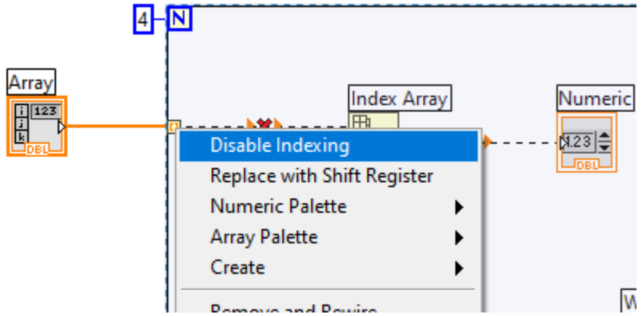



For Loop2.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
N
Array
Numeric
Iteration
i
Wait (ms)
1000

For Loop - N

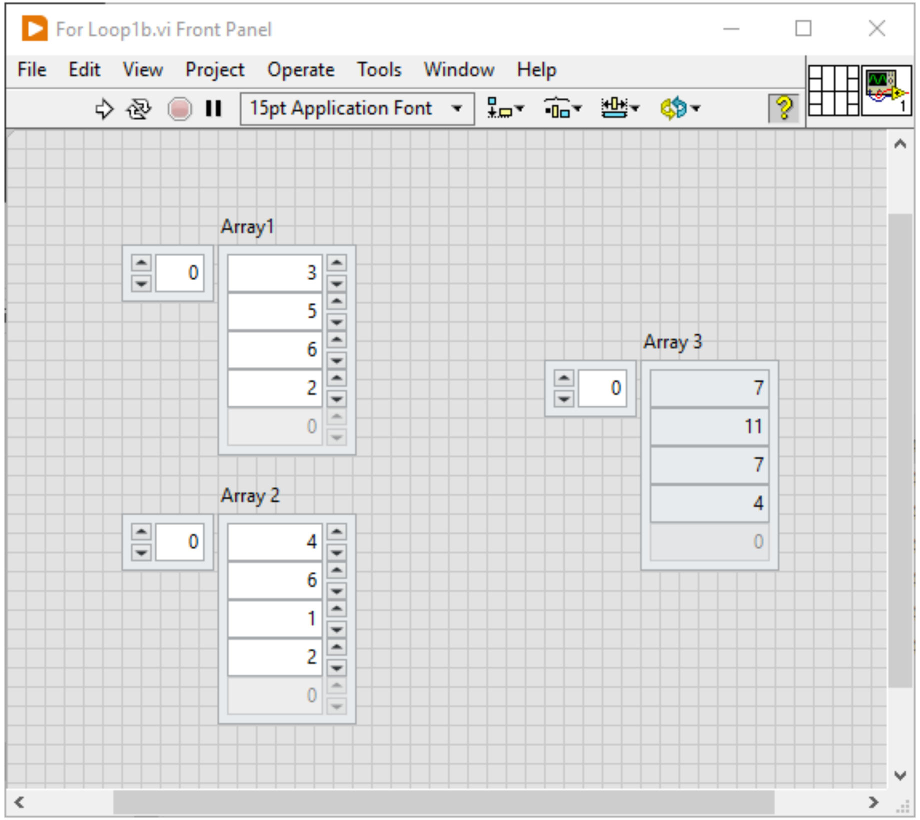



For Loop3.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
4
N
Array
Index Array
Numeric
Iteration
Wait (ms)
1000
i



Right-click and select “Disable Indexing”

For Loop Example


For Loop1b.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font

Array1

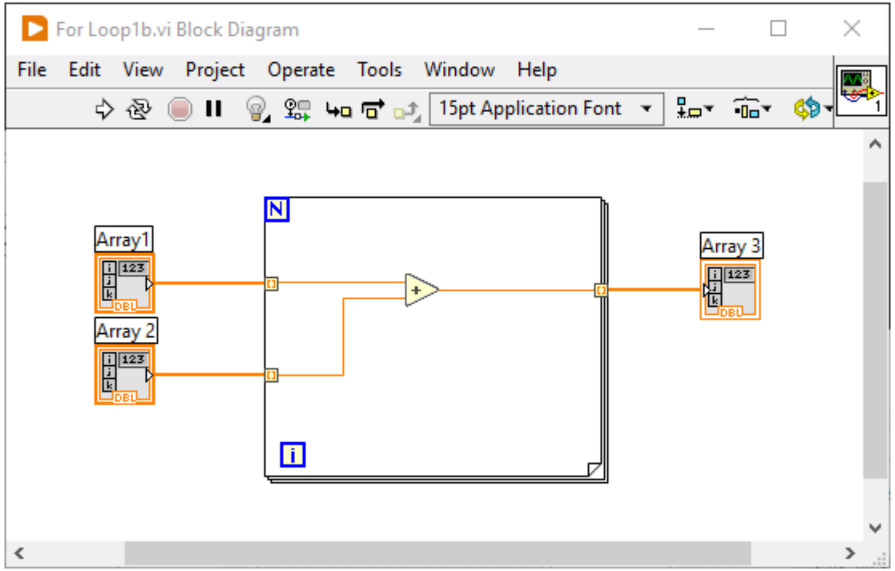
0	3
	5
	6
	2
	0

Array 2

0	4
	6
	1
	2
	0

Array 3

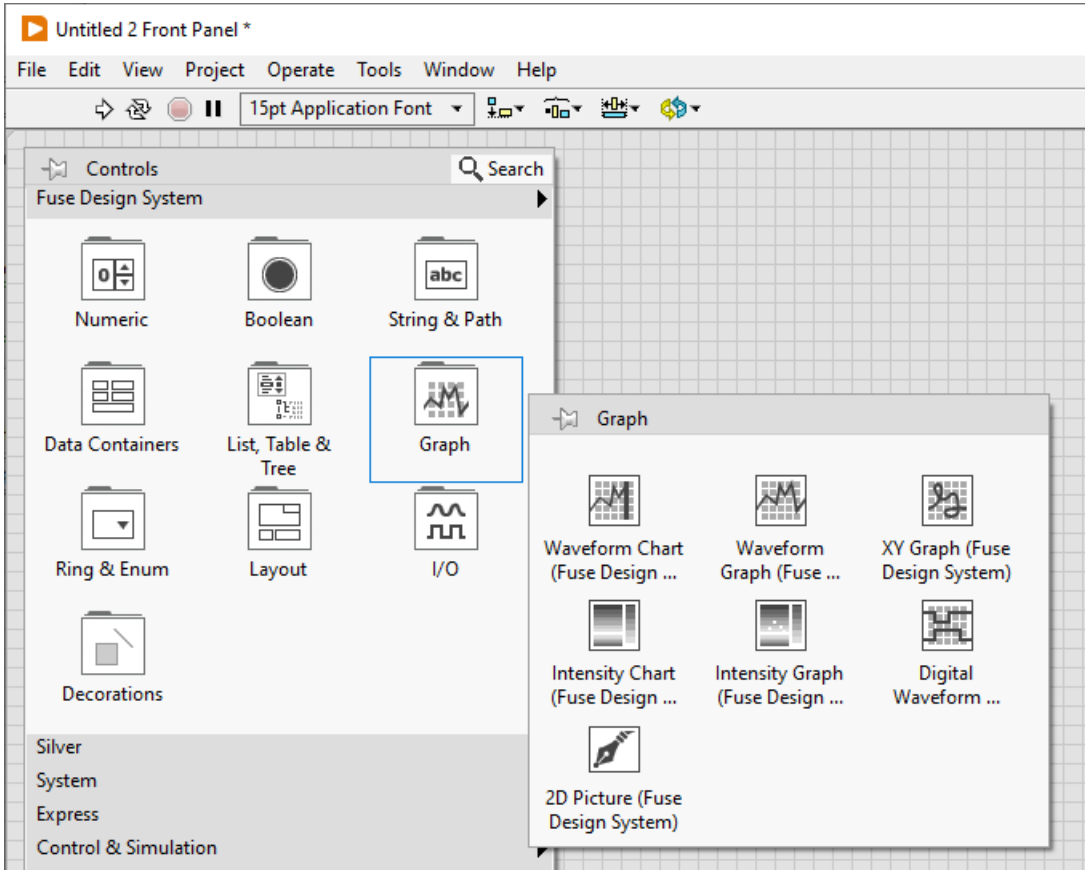
0	7
	11
	7
	4
	0


For Loop1b.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Array1
Array 2
Array 3
N
i
DBL

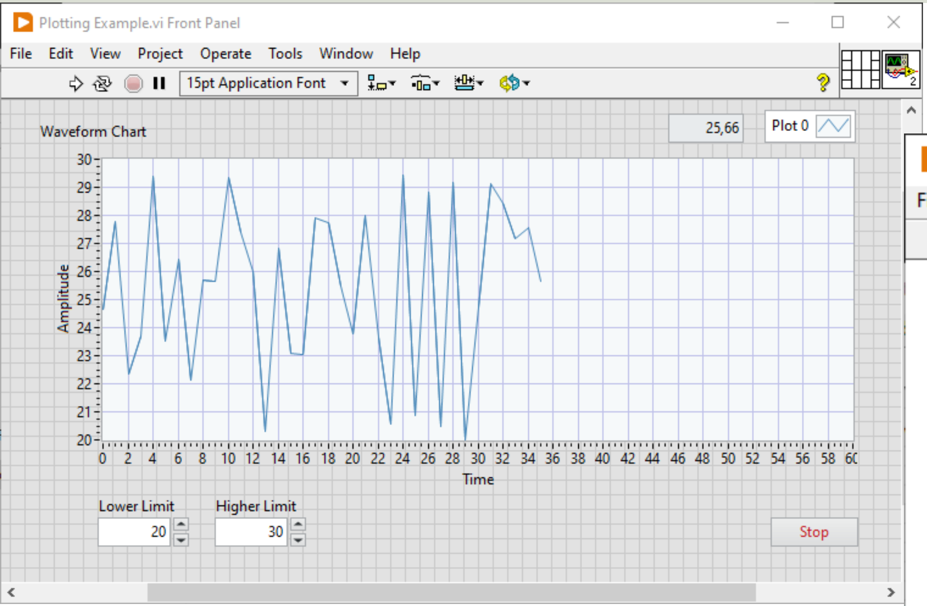
Plotting in LabVIEW

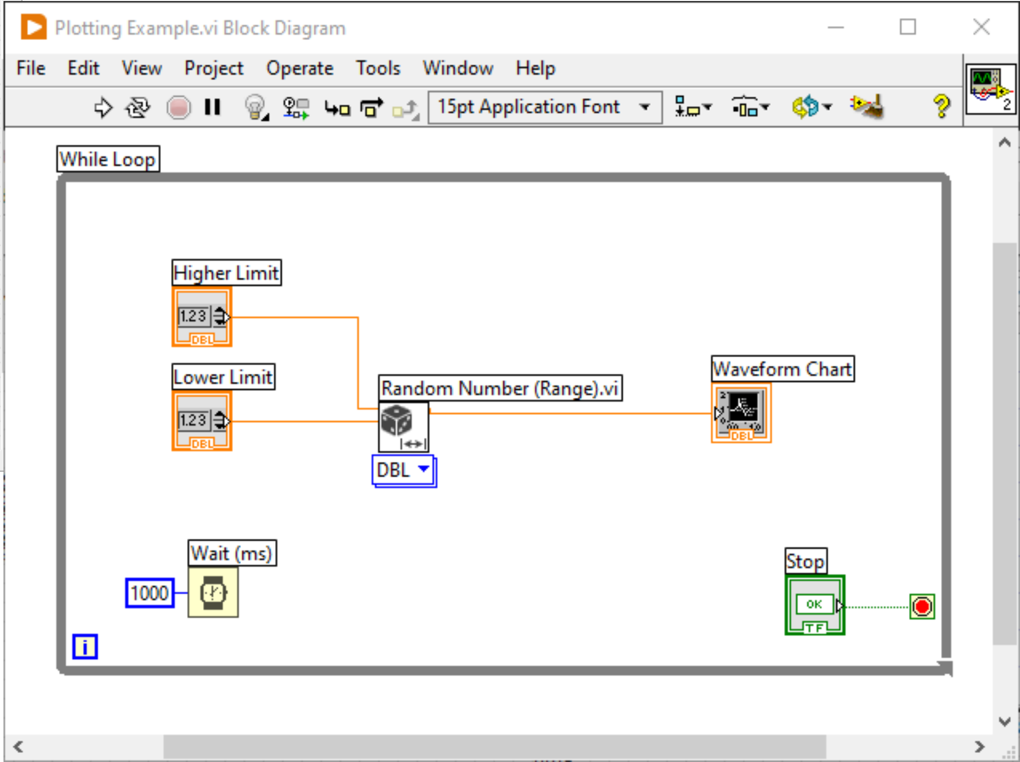
Hans-Petter Halvorsen

Plotting

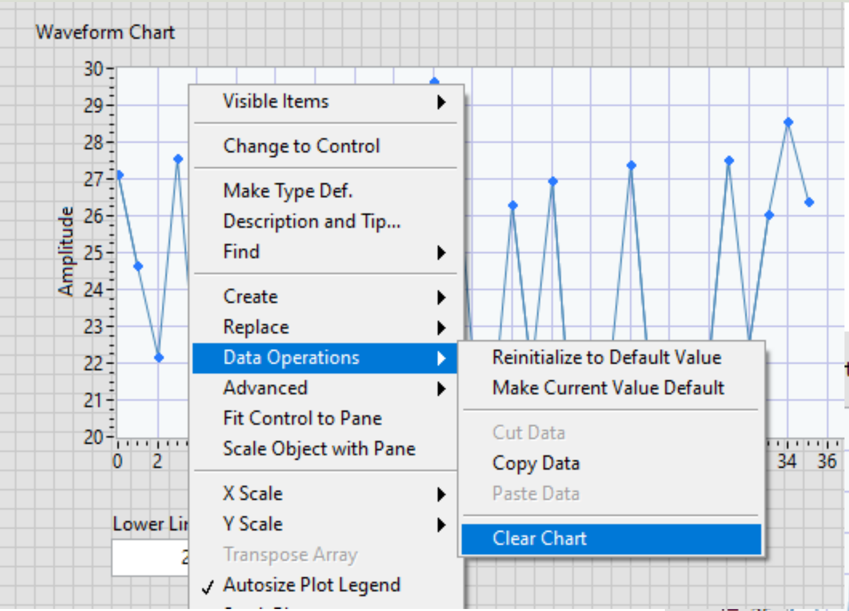

Untitled 2 Front Panel *
File
Edit
View
Project
Operate
Tools
Window
Help
15pt Application Font
Controls
Search
Fuse Design System
Numeric
Boolean
String & Path
Data Containers
List, Table & Tree
Graph
Ring & Enum
Layout
I/O
Decorations
Silver
System
Express
Control & Simulation
Graph
Waveform Chart (Fuse Design ...
Waveform Graph (Fuse ...
XY Graph (Fuse Design System)
Intensity Chart (Fuse Design ...
Intensity Graph (Fuse Design ...
Digital Waveform ...
2D Picture (Fuse Design System)

Plotting Example

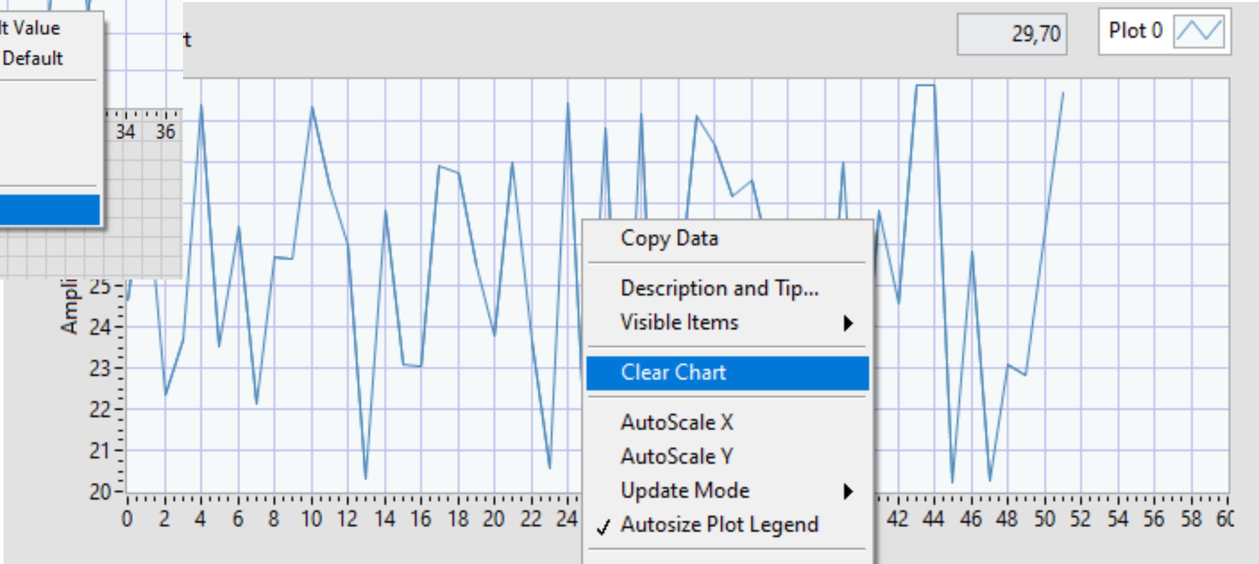




Clear Chart



You can clear the Chart Data in different ways: Right-click and select “Data Operations -> Clear Chart” or when your program is running: Right-click and select “Clear Chart”



Scaling Factors

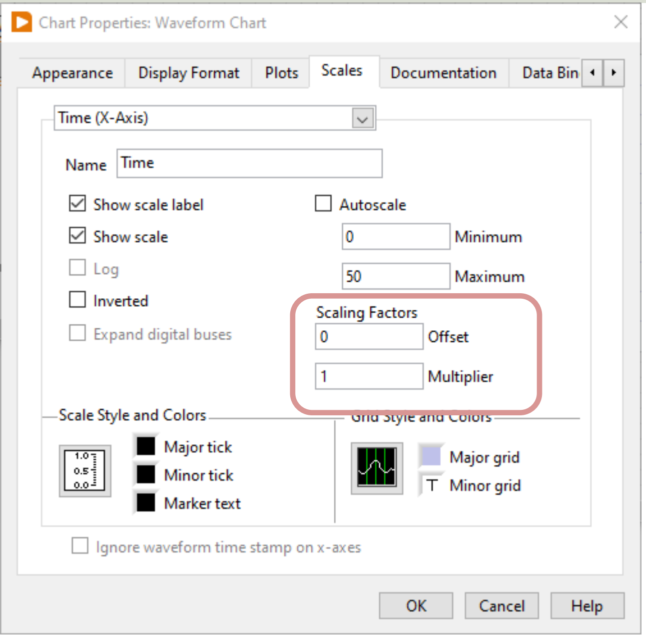


Chart Properties: Waveform Chart

Appearance | Display Format | Plots | Scales | Documentation | Data Bin

Time (X-Axis)

Name: Time

- [✓] Show scale label
- [✓] Show scale
- [ ] Log
- [ ] Inverted
- [ ] Expand digital buses
- [ ] Autoscale

Minimum: 0

Maximum: 50

Scaling Factors

Offset: 0

Multiplier: 1

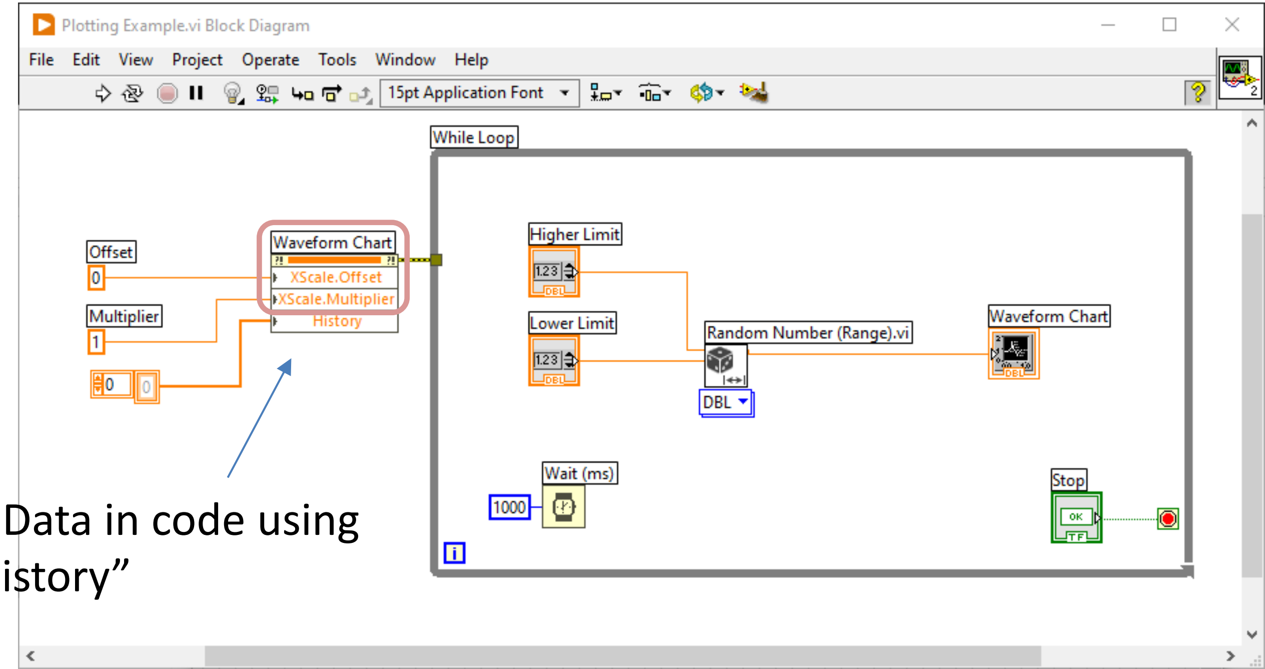
Scale Style and Colors: Major tick, Minor tick, Marker text

Grid Style and Colors: Major grid, Minor grid

- [ ] Ignore waveform time stamp on x-axes

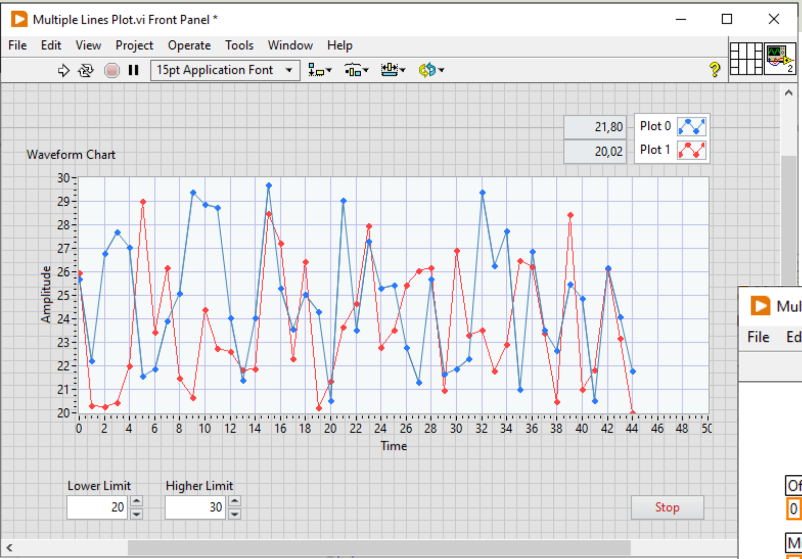
OK | Cancel | Help

You can set these settings either in the GUI or in code using Property Nodes. Offset is typically set equal to zero (starting value on the x-axis), while Multiplier is the interval between to values, e.g., if you plot a new data point every second, you set Multiplier=1, etc.



You can also clear the Chart Data in code using the Property Node called “History”

Multi-Line Plotting



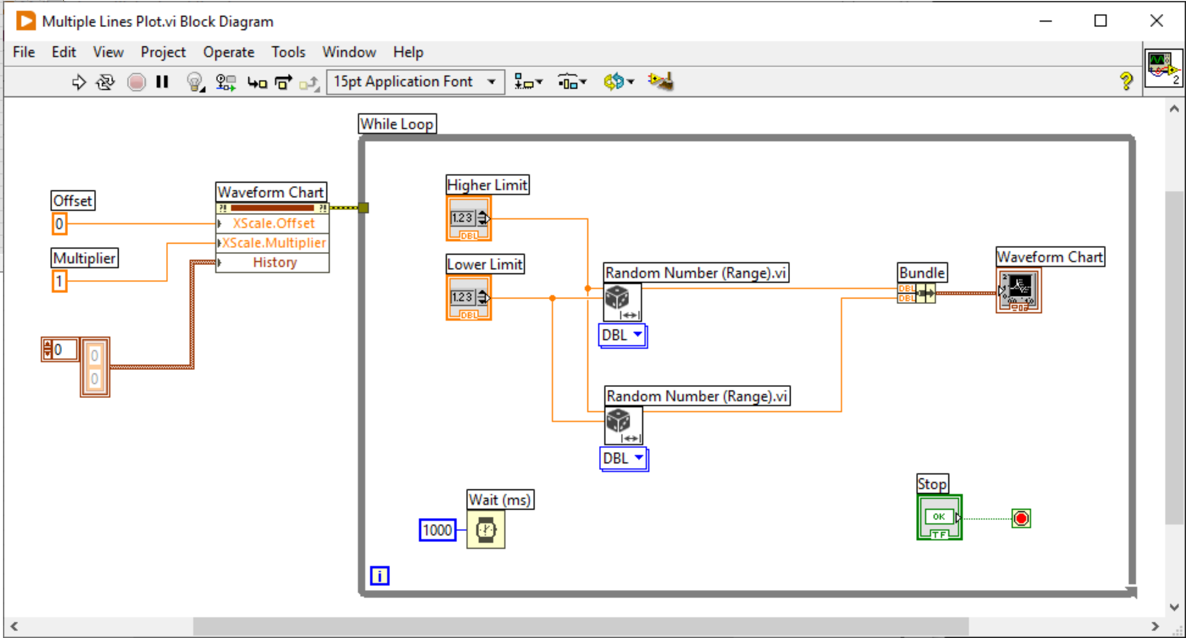
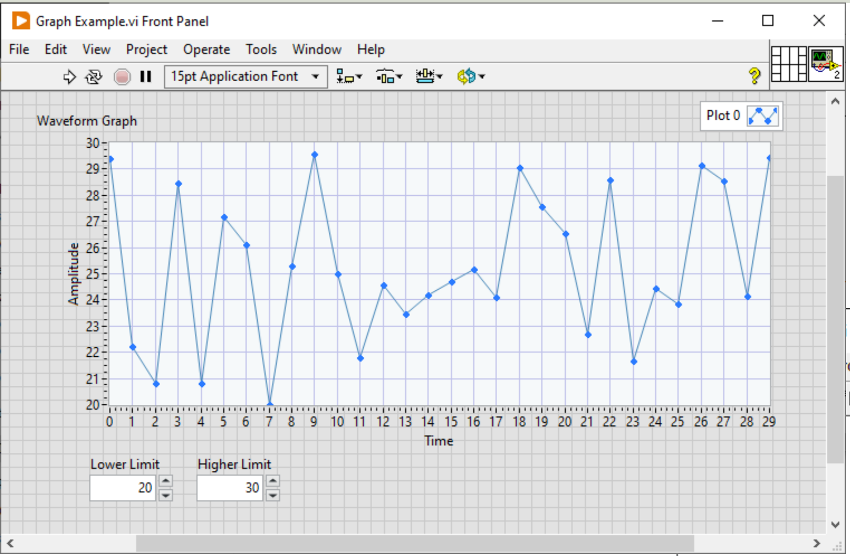
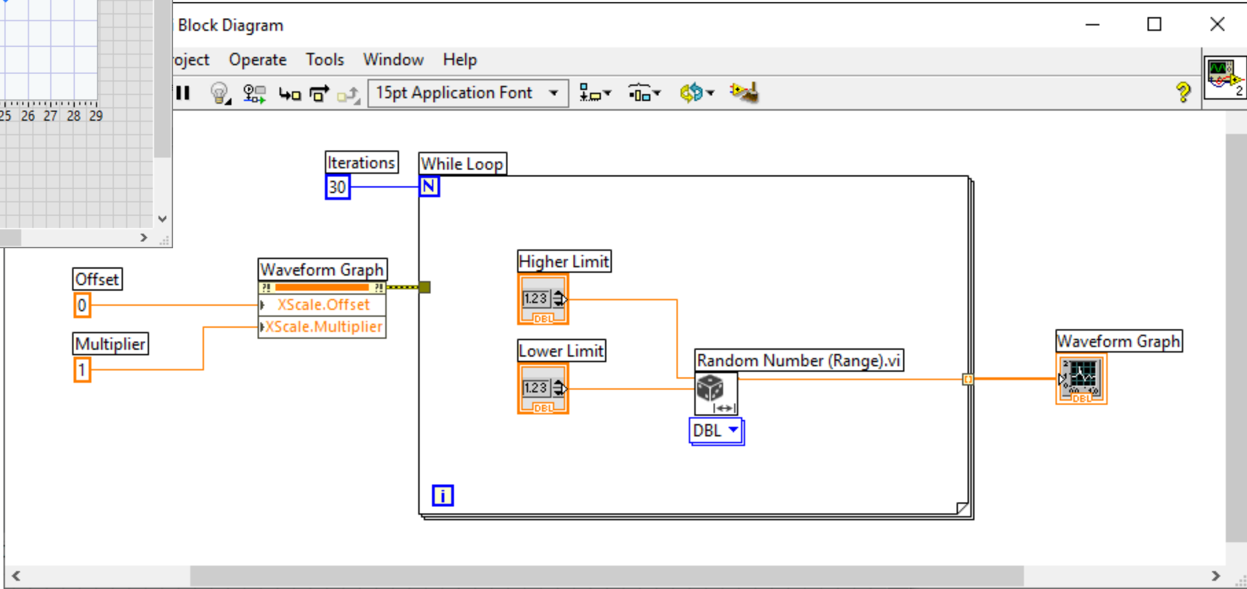


Chart vs Graph



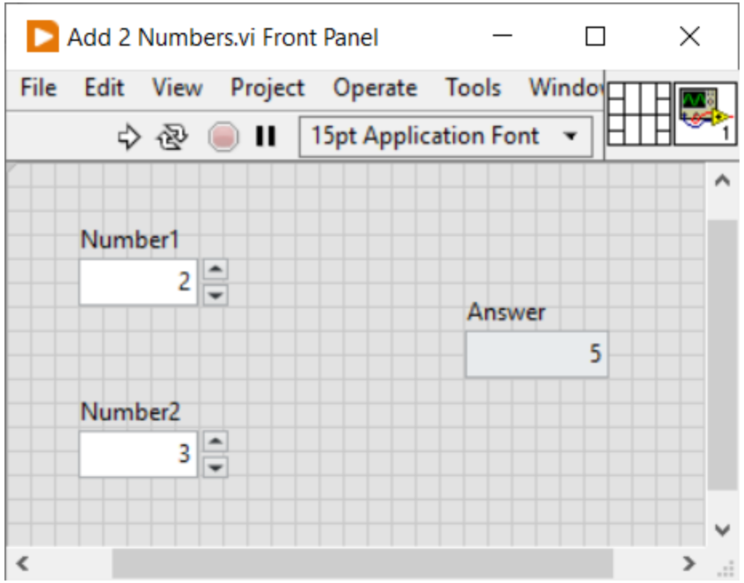


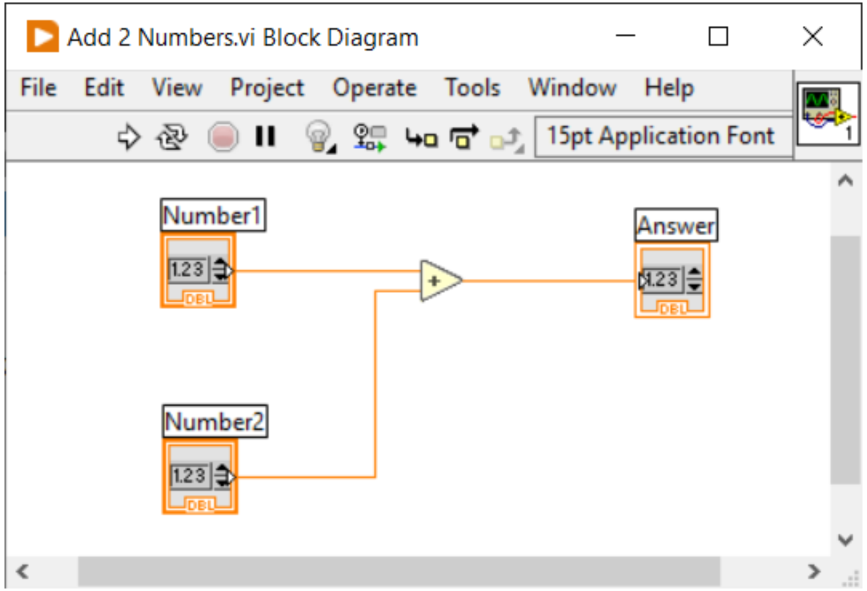
LabVIEW Fundamentals

SubVIs in LabVIEW

Hans-Petter Halvorsen

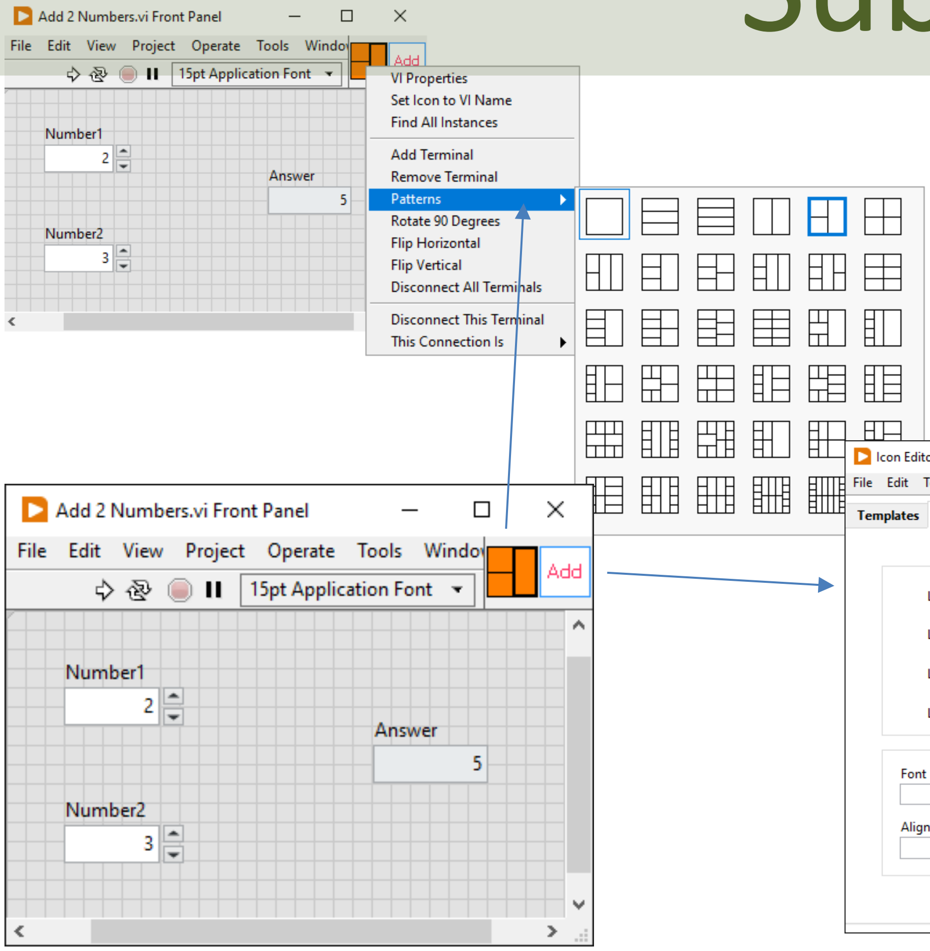
SubVIs


Add 2 Numbers.vi Front Panel
File Edit View Project Operate Tools Windo
15pt Application Font
Number1
2
Number2
3
Answer
5

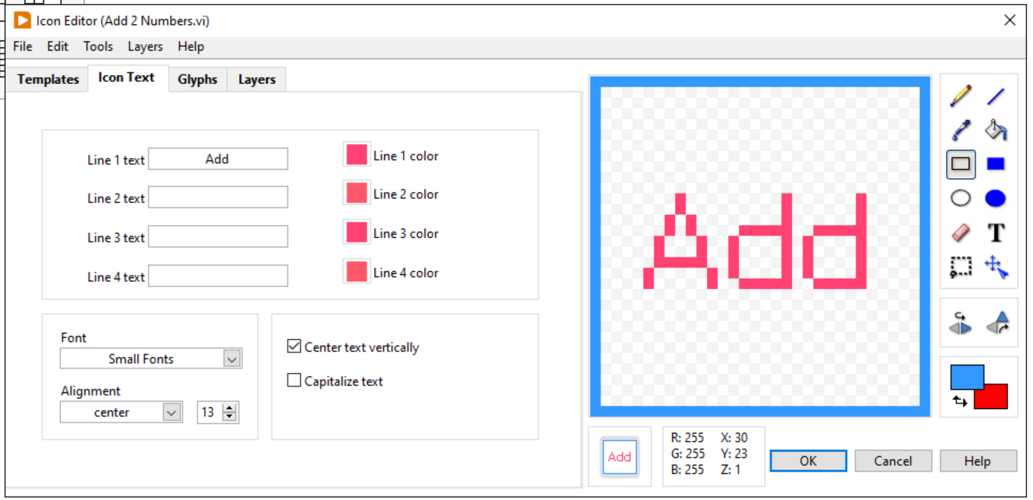

Add 2 Numbers.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Number1
Number2
Answer
1.23
DBL

SubVIs

Create Input and Outputs and create an Icon using the Icon Editor

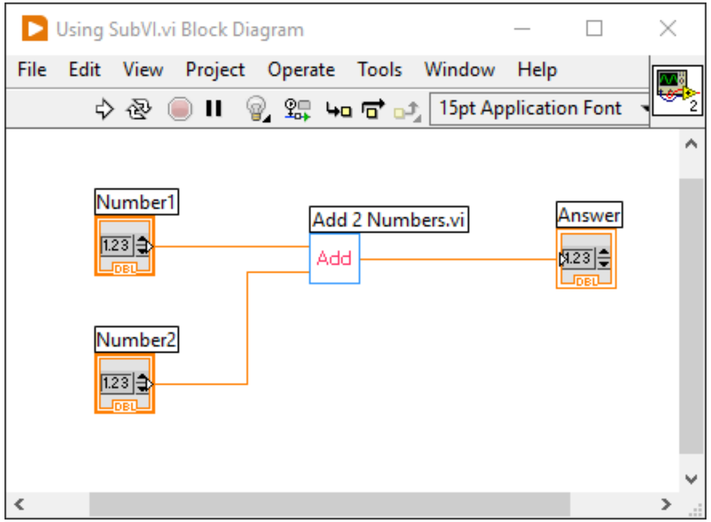


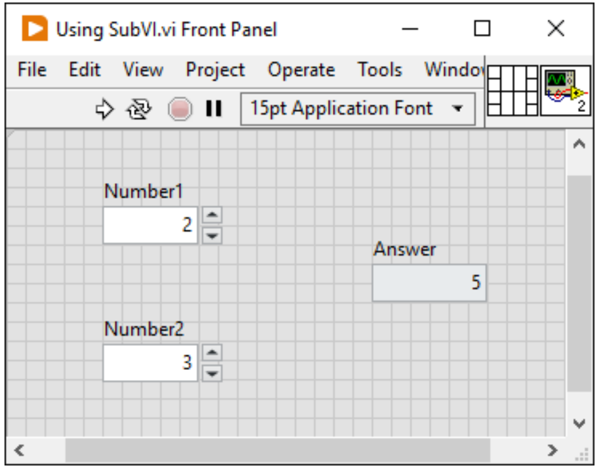
Icon Editor:


Icon Editor (Add 2 Numbers.vi)
File Edit Tools Layers Help
Templates Icon Text Glyphs Layers
Add
R: 255 G: 255 B: 255
X: 30 Y: 23 Z: 1
OK Cancel Help

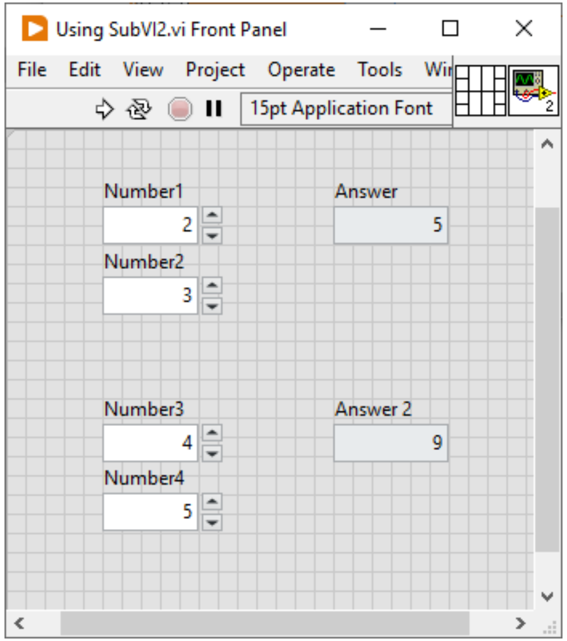
Line 1 text: Add
Line 2 text:
Line 3 text:
Line 4 text:
Line 1 color 🟥
Line 2 color 🟥
Line 3 color 🟥
Line 4 color 🟥
Font: Small Fonts
Alignment: center 13
☑ Center text vertically
☐ Capitalize text

Using SubVIs

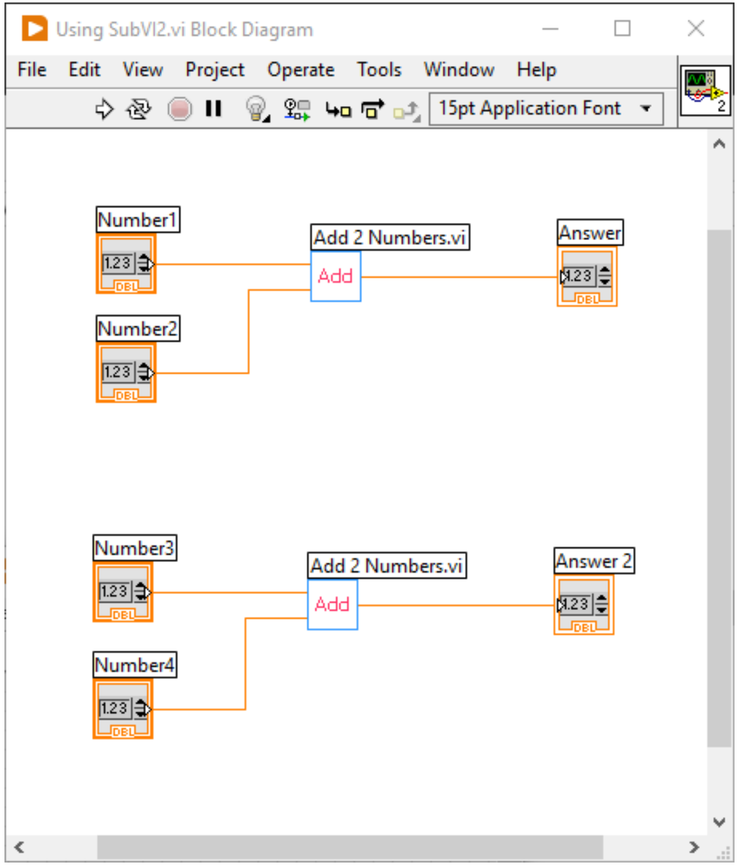

Using SubVI.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Number1
Number2
Add 2 Numbers.vi
Add
Answer
1.23
DBL


Using SubVI.vi Front Panel
File Edit View Project Operate Tools Windo
15pt Application Font
Number1
2
Number2
3
Answer
5

Using SubVIs


Using SubVI2.vi Front Panel
File Edit View Project Operate Tools Wi
15pt Application Font

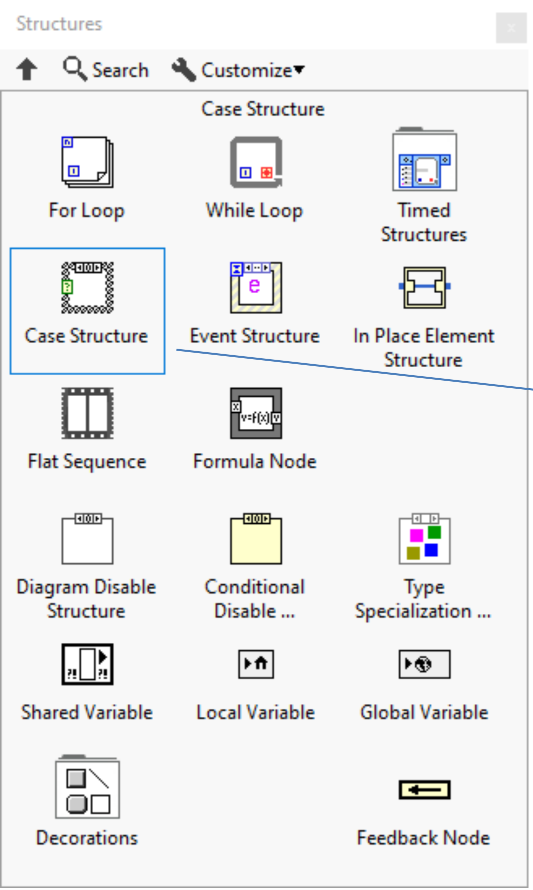
Number1 2
Number2 3
Answer 5
Number3 4
Number4 5
Answer 2 9

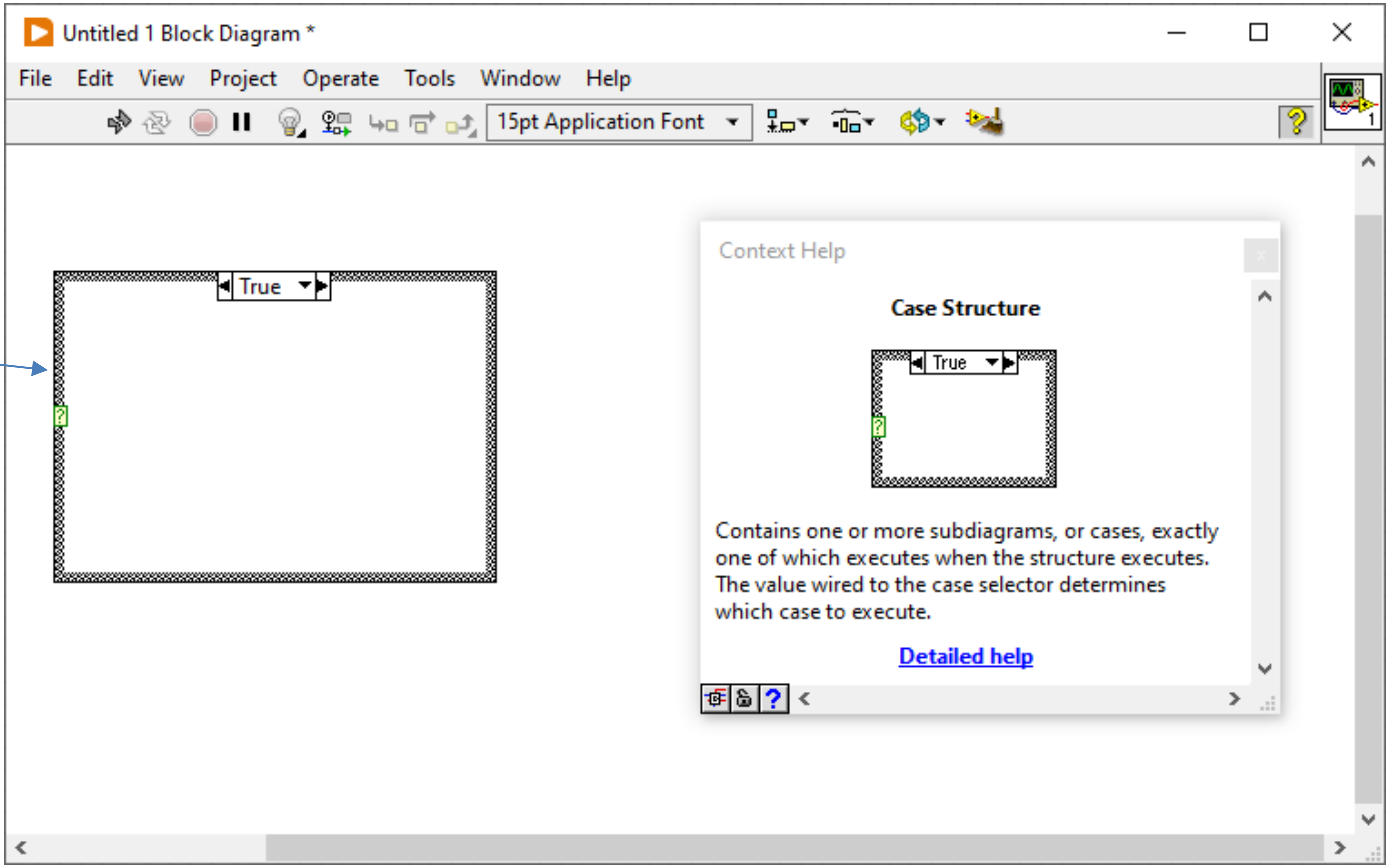


LabVIEW Fundamentals

Case Structures in LabVIEW

Case Structure

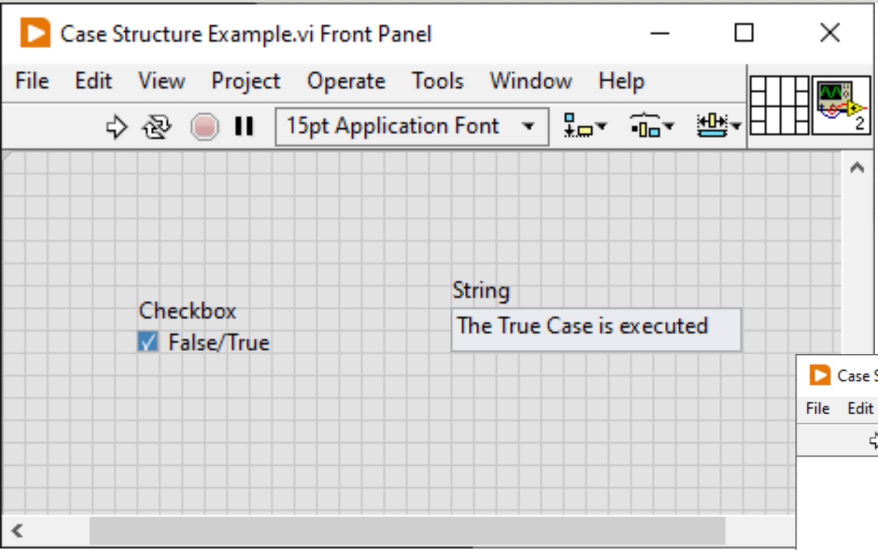




Contains one or more subdiagrams, or cases, exactly one of which executes when the structure executes. The value wired to the case selector determines which case to execute.

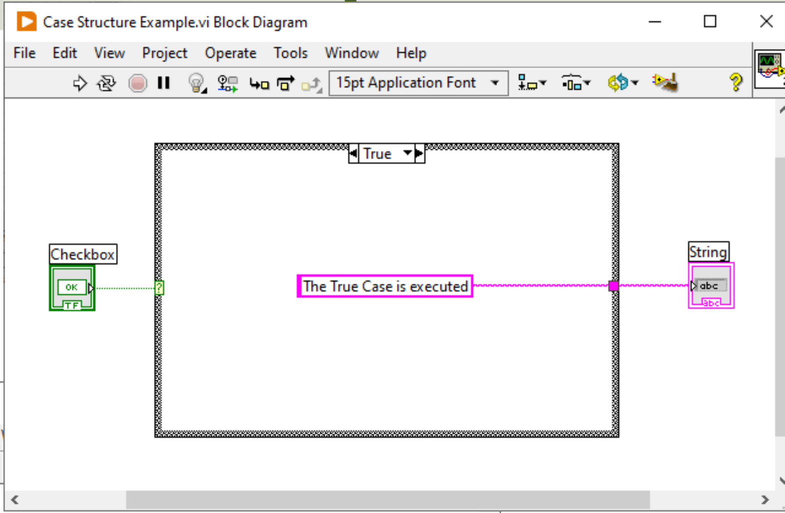
Detailed help

Case Structure Example

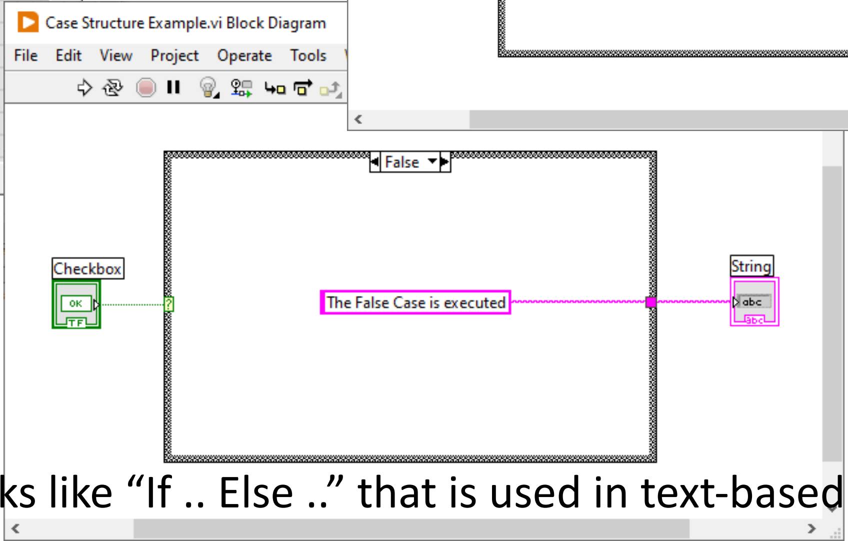

Case Structure Example.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font

Checkbox
✓ False/True

String
The True Case is executed


Case Structure Example.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
True
Checkbox
String

The True Case is executed

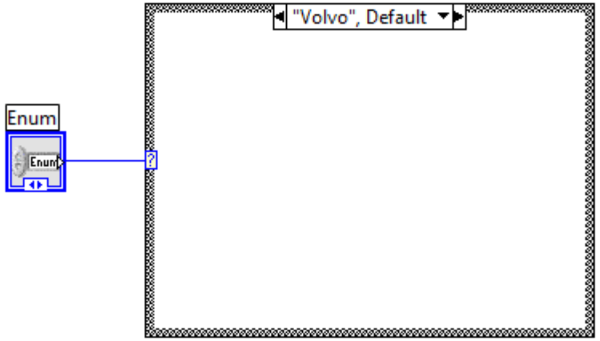

Case Structure Example.vi Block Diagram
File Edit View Project Operate Tools
False
Checkbox
String

The False Case is executed

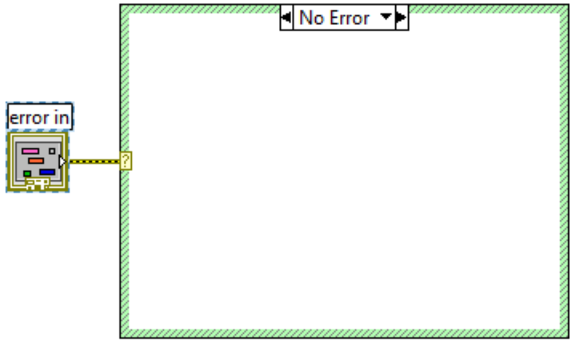
Here you see a Case structure that works like “If .. Else ..” that is used in text-based languages

Data Types

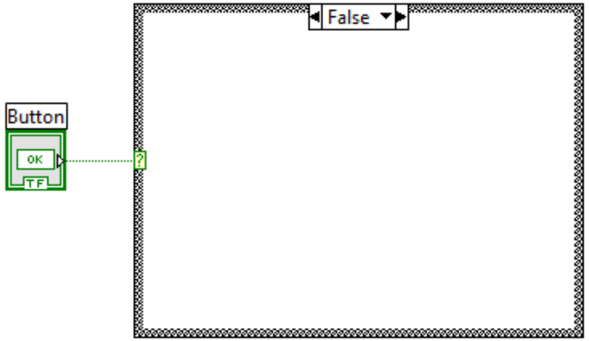
Enum



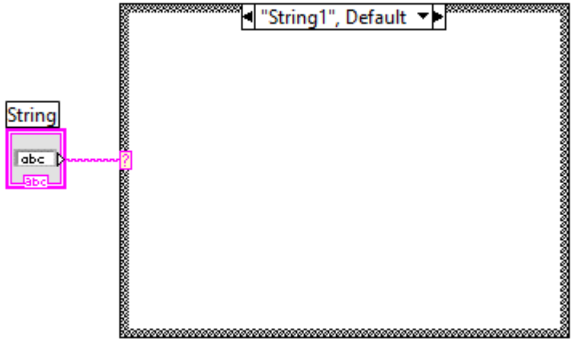
Error Cluster



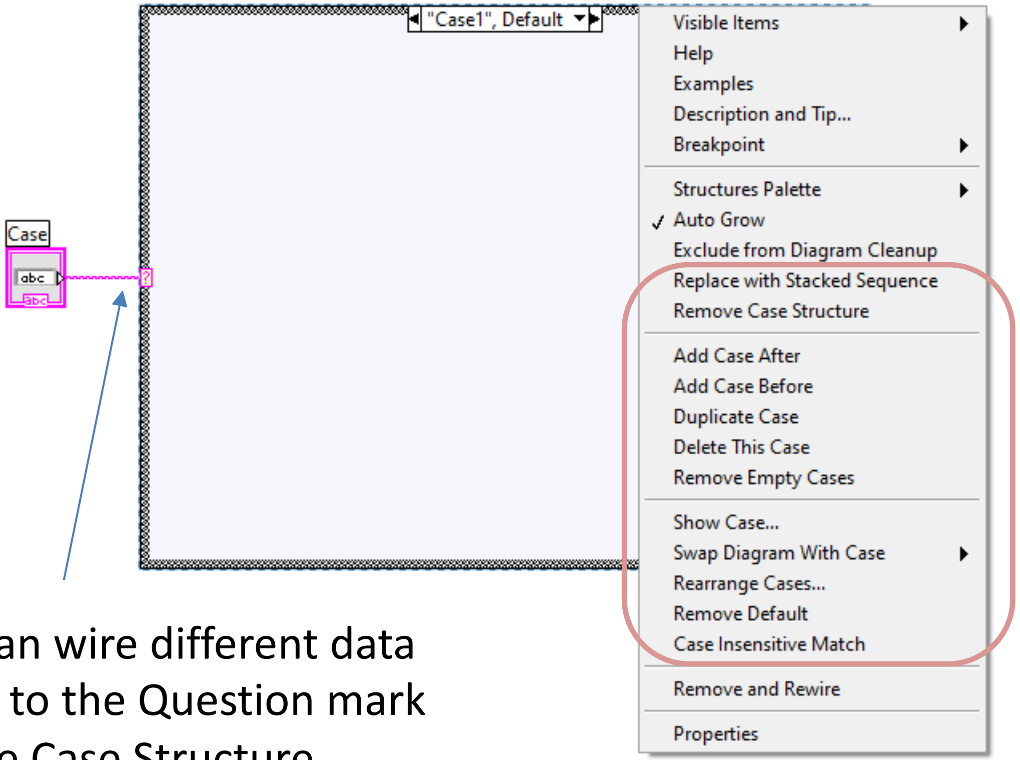
Boolean



String



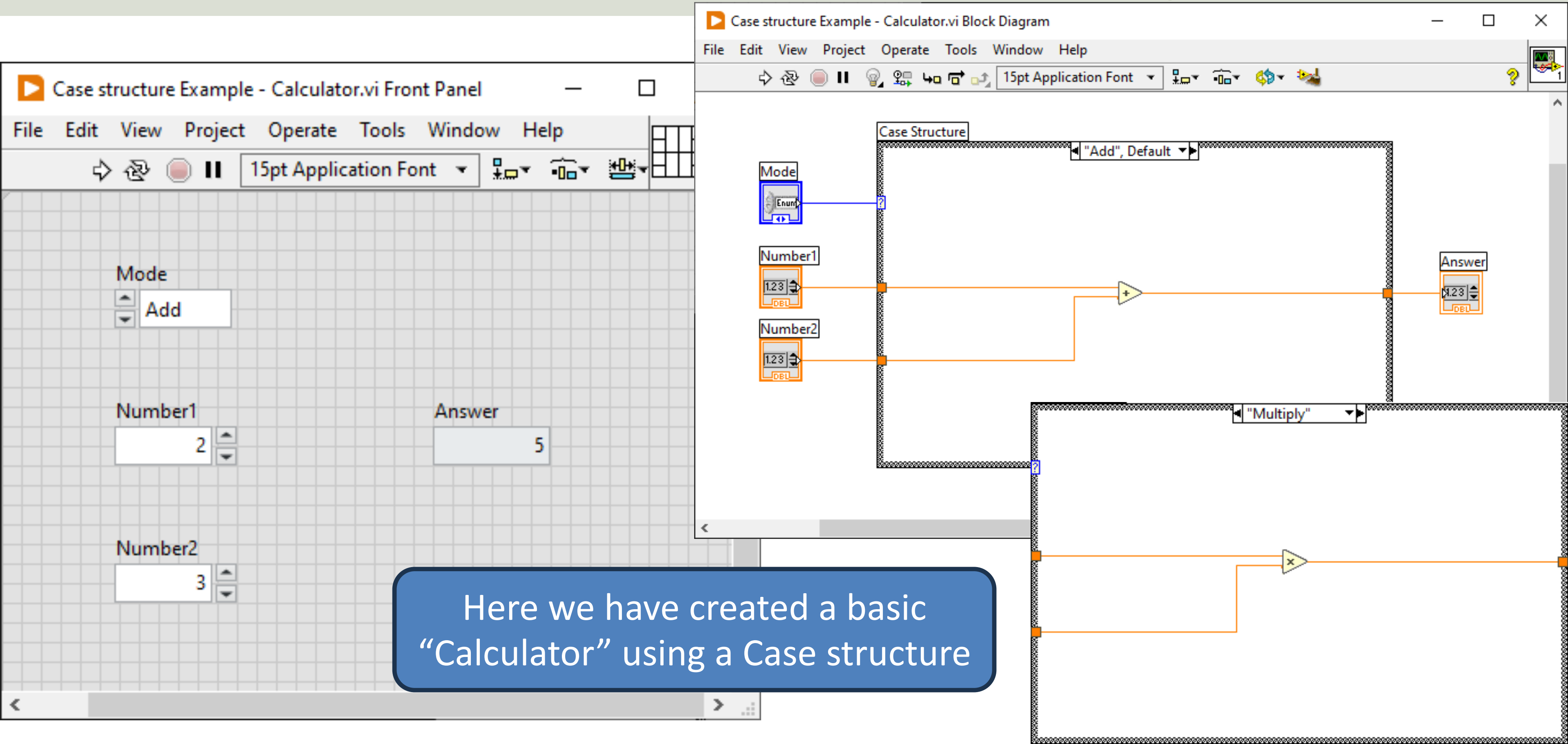
Case Structure



Here you have lots of options for dealing with the Case Structure

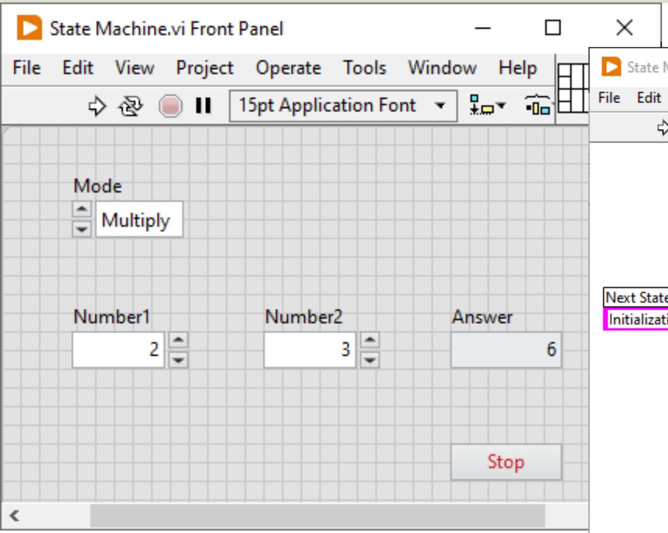
You can wire different data types to the Question mark on the Case Structure

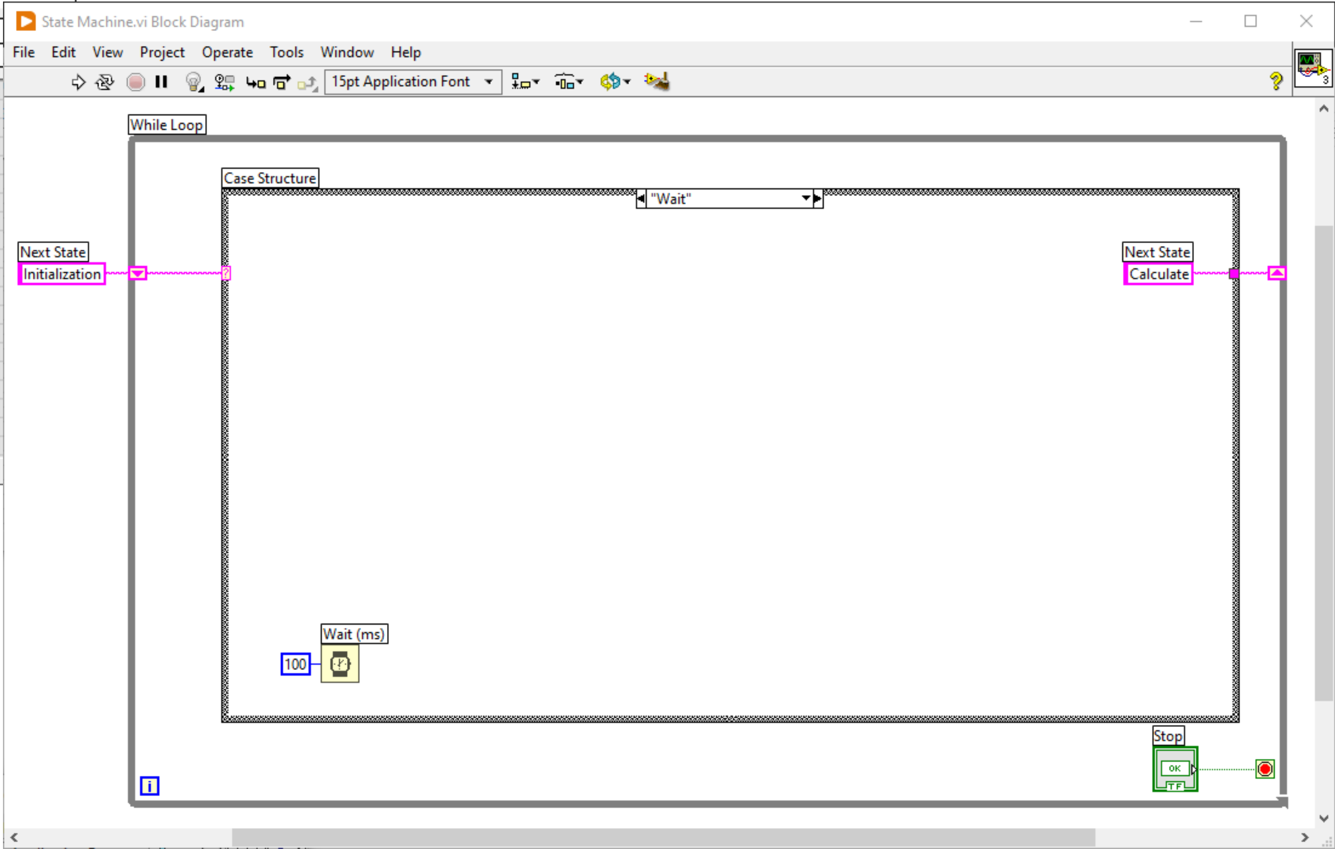
Case Structure Example 2


Case structure Example - Calculator.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Mode
Add
Number1
2
Answer
5
Number2
3
Case structure Example - Calculator.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Case Structure
"Add", Default
Mode
Enum
Number1
1.23
DBL
Number2
Answer
"Multiply"

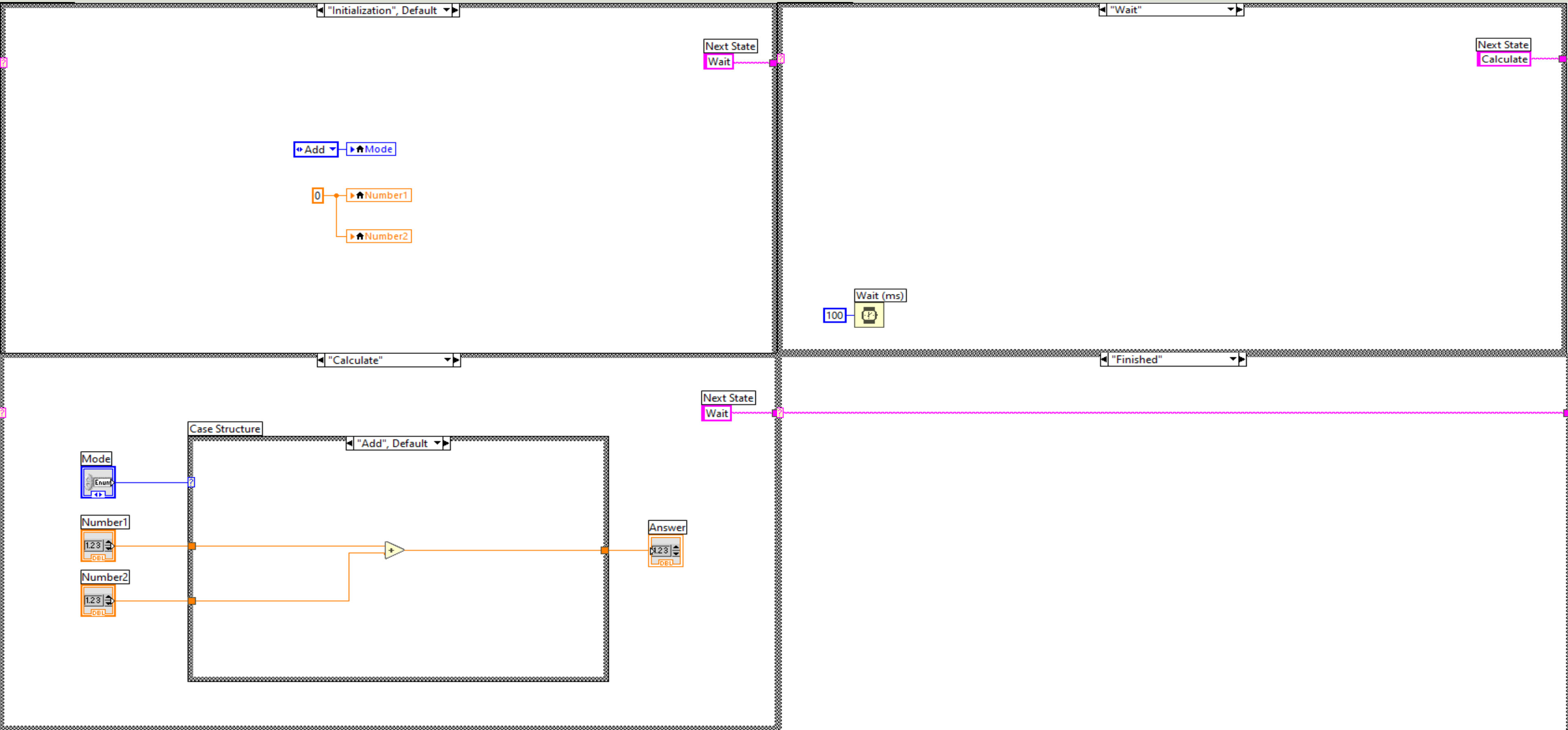
Here we have created a basic “Calculator” using a Case structure

State Machine

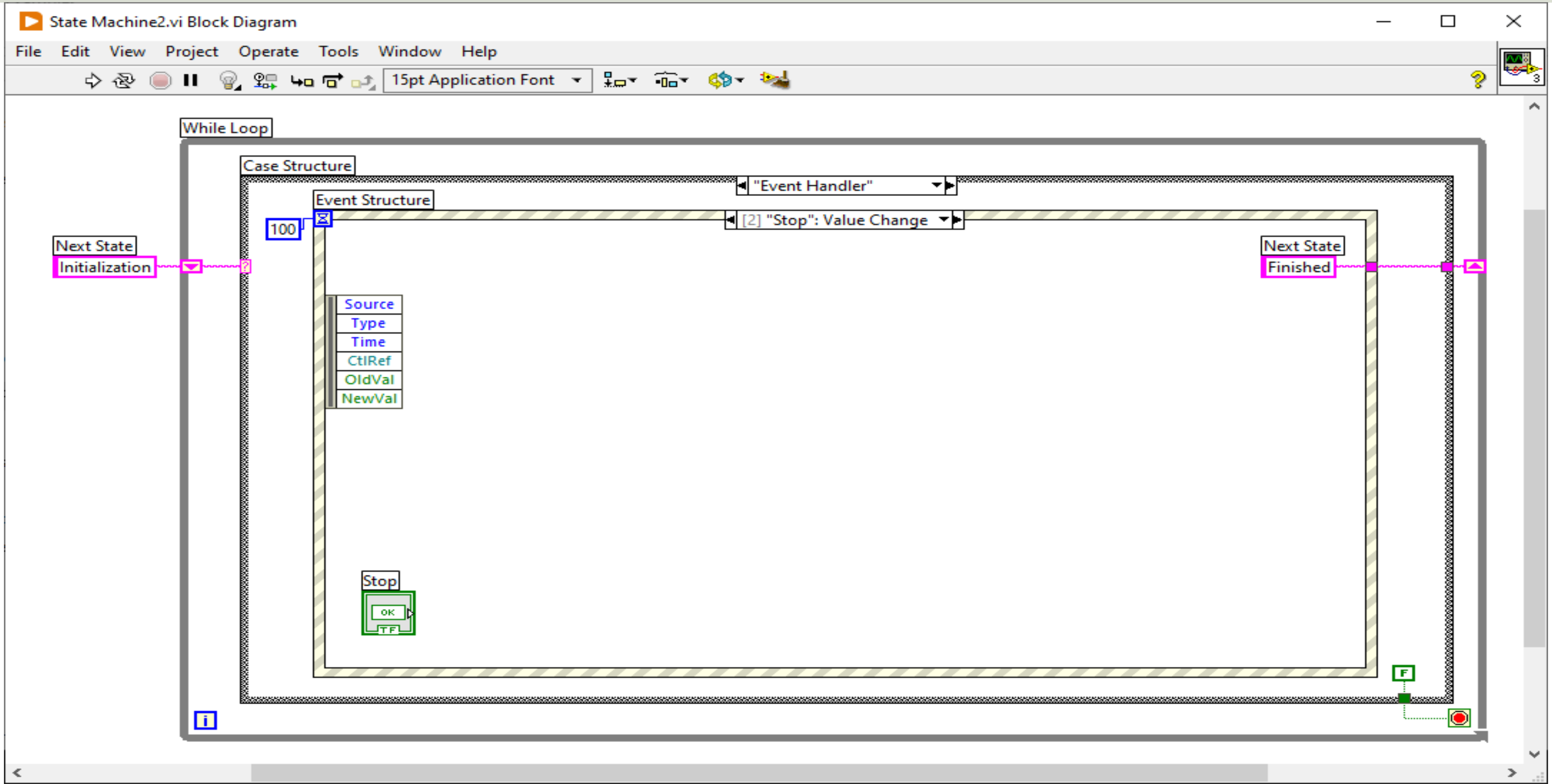

State Machine.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Mode
Multiply
Number1
2
Number2
3
Answer
6
Stop



State Machine



Improved State Machine

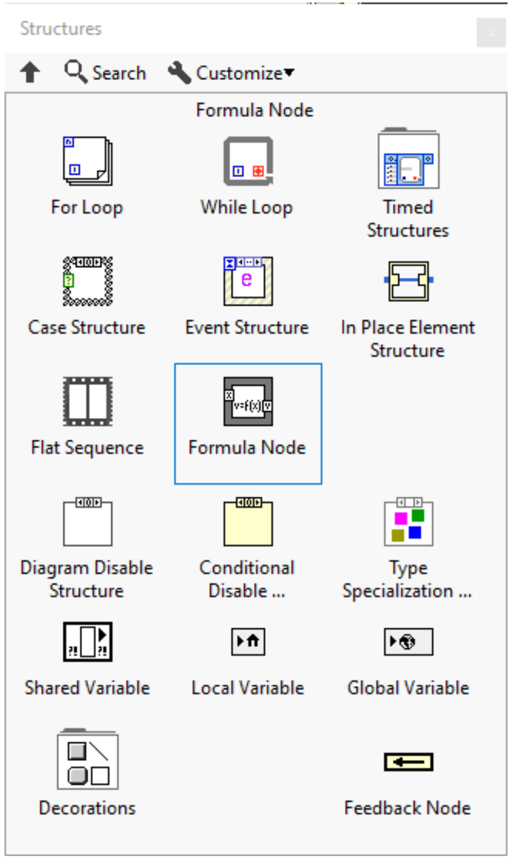


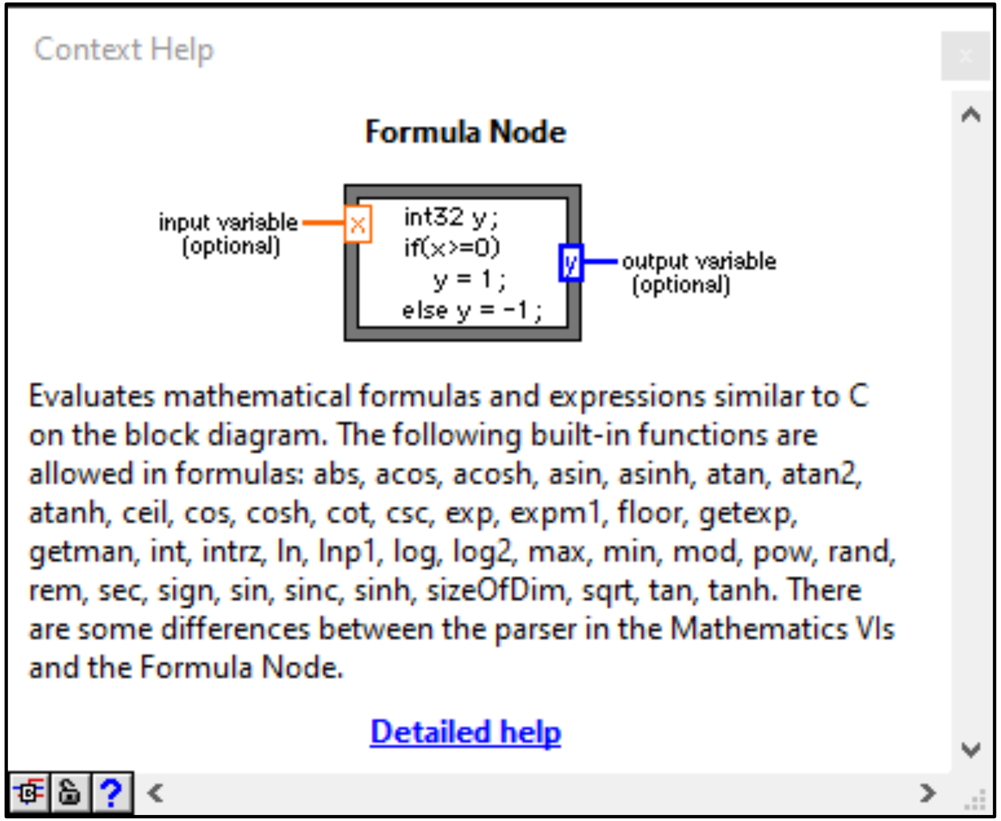
LabVIEW Fundamentals

Formula Nodes in LabVIEW

Hans-Petter Halvorsen

Formula Node





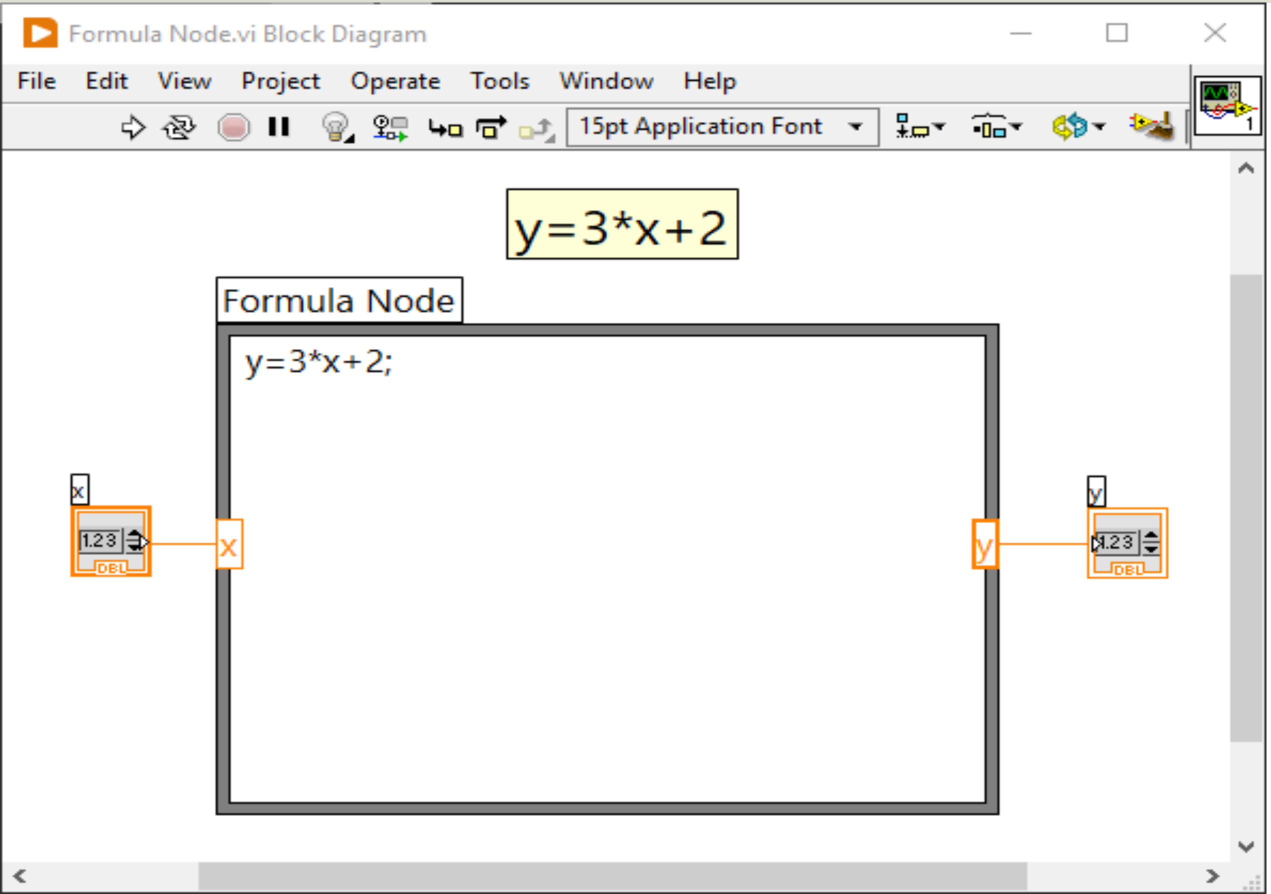
Formula Node

```
int32 y;
if(x>=0)
  y = 1;
else y = -1;
```

Evaluates mathematical formulas and expressions similar to C on the block diagram. The following built-in functions are allowed in formulas: abs, acos, acosh, asin, asinh, atan, atan2, atanh, ceil, cos, cosh, cot, csc, exp, expm1, floor, getexp, getman, int, intrz, ln, lnp1, log, log2, max, min, mod, pow, rand, rem, sec, sign, sin, sinc, sinh, sizeOfDim, sqrt, tan, tanh. There are some differences between the parser in the Mathematics VIs and the Formula Node.

Detailed help

Formula Node Example


Formula Node.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
x
1.23
DBL
y

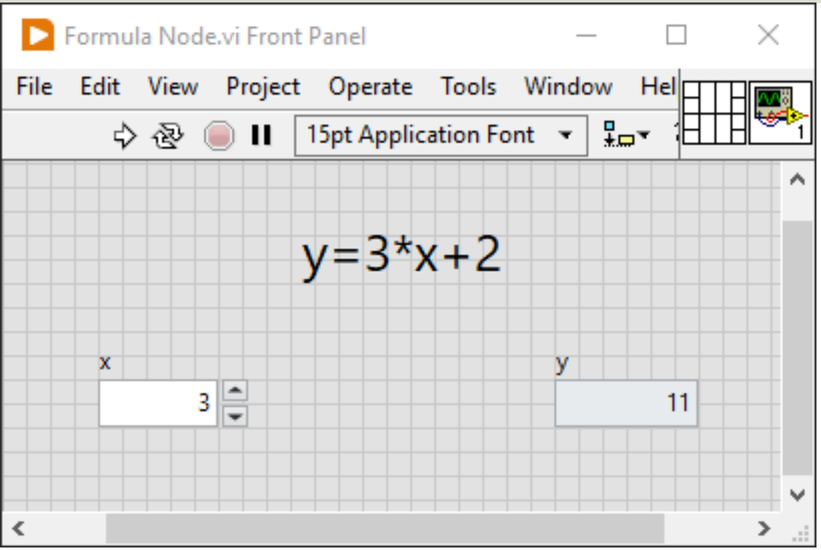
y=3*x+2

Formula Node

```
y=3*x+2;
```

- We use the Formula Node to create equations, calculations, mathematical expressions, etc.
- Simulations, etc.
- The Formula Node uses C syntax

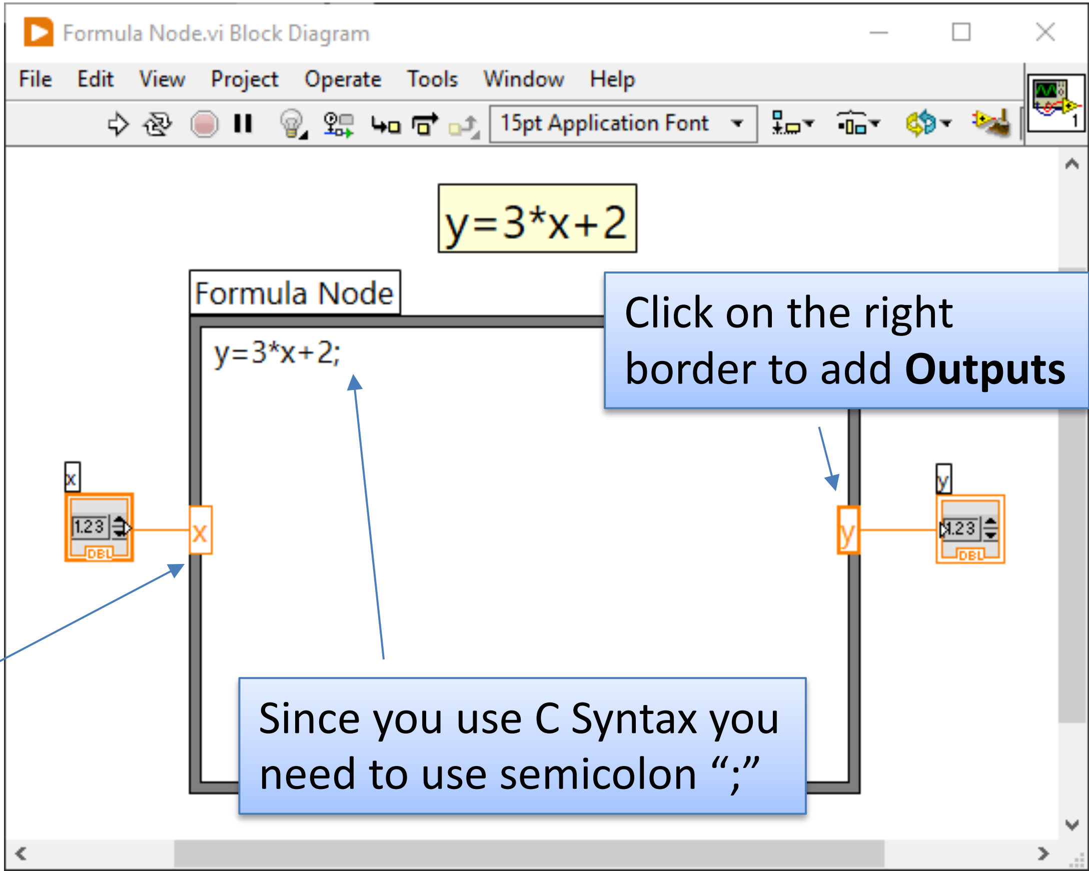
Formula Node Example


Formula Node.vi Front Panel
File Edit View Project Operate Tools Window Hel
15pt Application Font

y=3*x+2

x: 3

y: 11


Formula Node.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Formula Node
1.23
DBL

y=3*x+2

```
y=3*x+2;
```

Click on the right border to add **Outputs**

Click on the left border to add **Inputs**

Since you use C Syntax you need to use semicolon “;”

Implementing an Advanced Equation

Advanced Mathematical Formula:

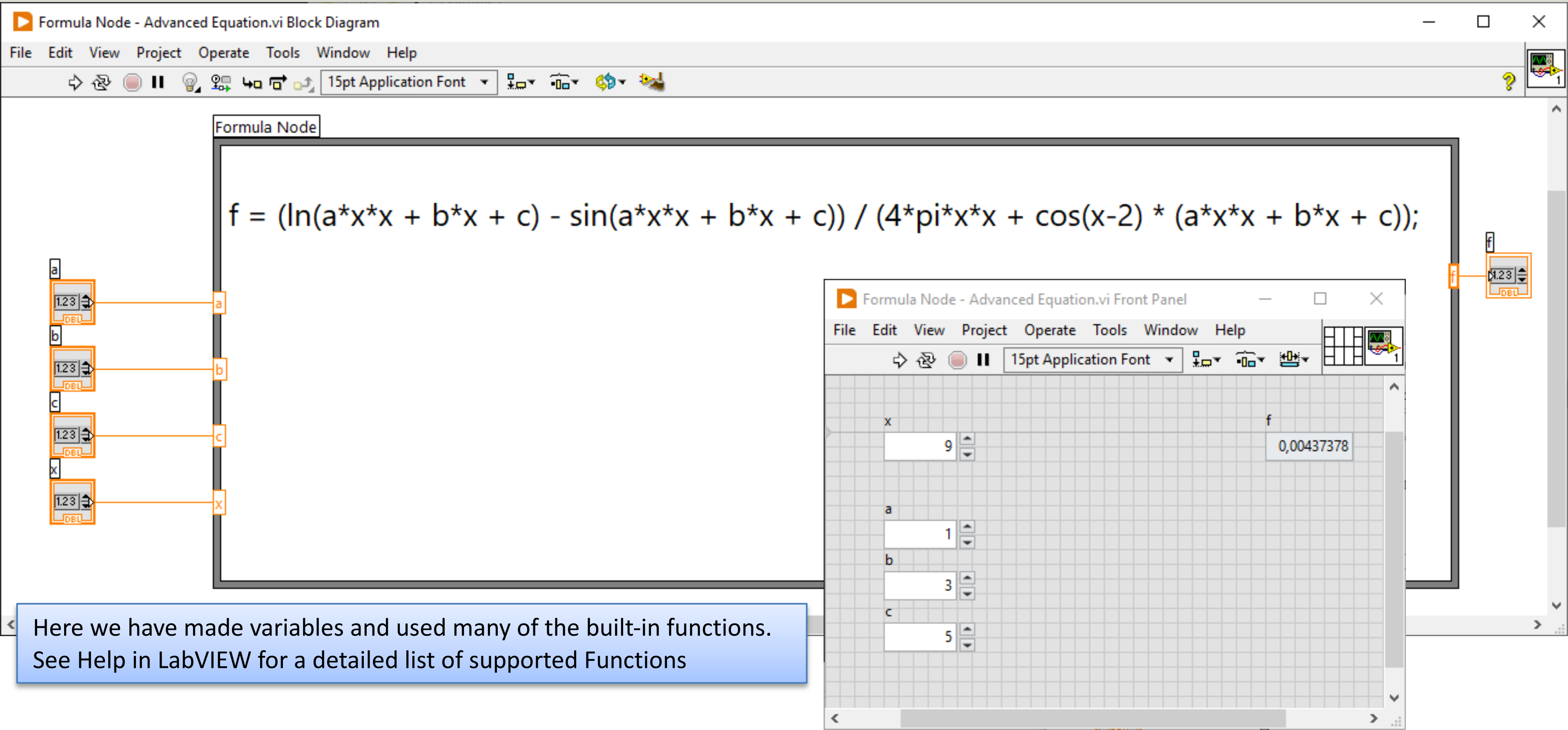
$$f(x) = \frac{\ln(ax^2 + bx + c) - \sin(ax^2 + bx + c)}{4\pi x^2 + \cos(x - 2)(ax^2 + bx + c)}$$

We will use the LabVIEW Formula Node to implement this formula

Given $a = 1, b = 3, c = 5$

Find $f(9)$ (The answer should be $f(9) = 0.0044$)

Implementing an Advanced Equation


Formula Node - Advanced Equation.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Formula Node
a
b
c
x
f
Formula Node - Advanced Equation.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font

```
f = (ln(a*x*x + b*x + c) - sin(a*x*x + b*x + c)) / (4*pi*x*x + cos(x-2) * (a*x*x + b*x + c));
```

x: 9
f: 0,00437378
a: 1
b: 3
c: 5

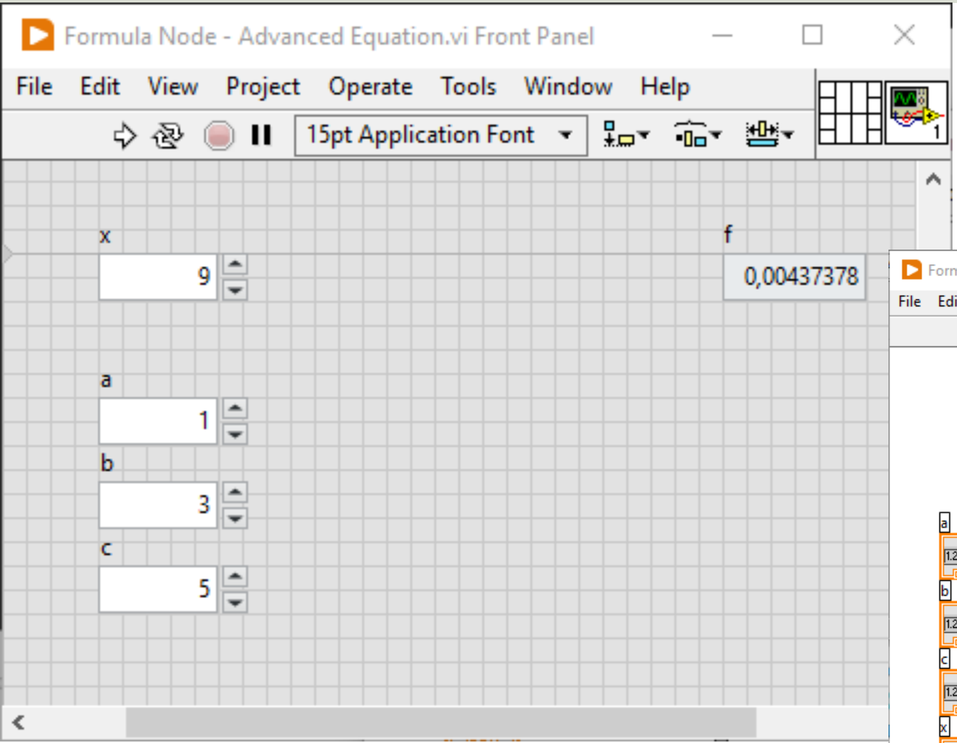
Here we have made variables and used many of the built-in functions. See Help in LabVIEW for a detailed list of supported Functions

Implementing an Advanced Equation

We can simplify the equation:

$$f(x) = \frac{\ln\overbrace{(ax^2 + bx + c)}^{g(x)} - \sin\overbrace{(ax^2 + bx + c)}^{g(x)}}{4\pi x^2 + \cos(x - 2)\underbrace{(ax^2 + bx + c)}_{g(x)}}$$

Implementing an Advanced Equation


Formula Node - Advanced Equation.vi Front Panel

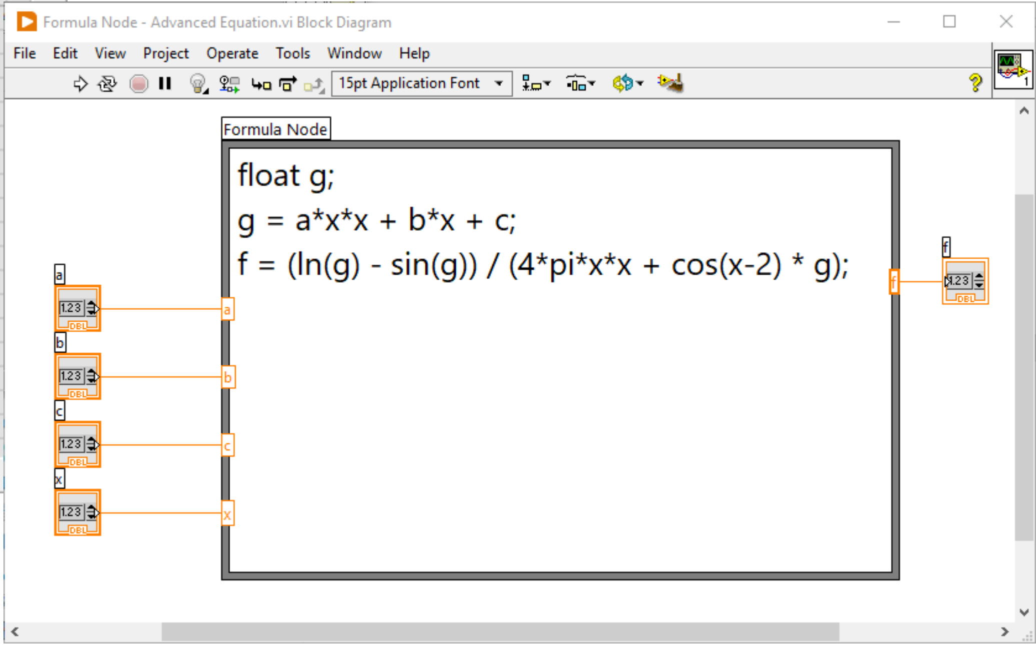
x: 9

a: 1

b: 3

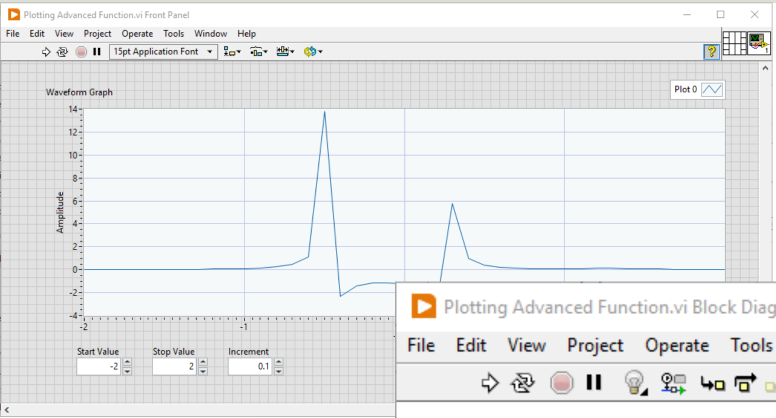
c: 5

f: 0,00437378


Formula Node - Advanced Equation.vi Block Diagram

```
float g;
g = a*x*x + b*x + c;
f = (ln(g) - sin(g)) / (4*pi*x*x + cos(x-2) * g);
```

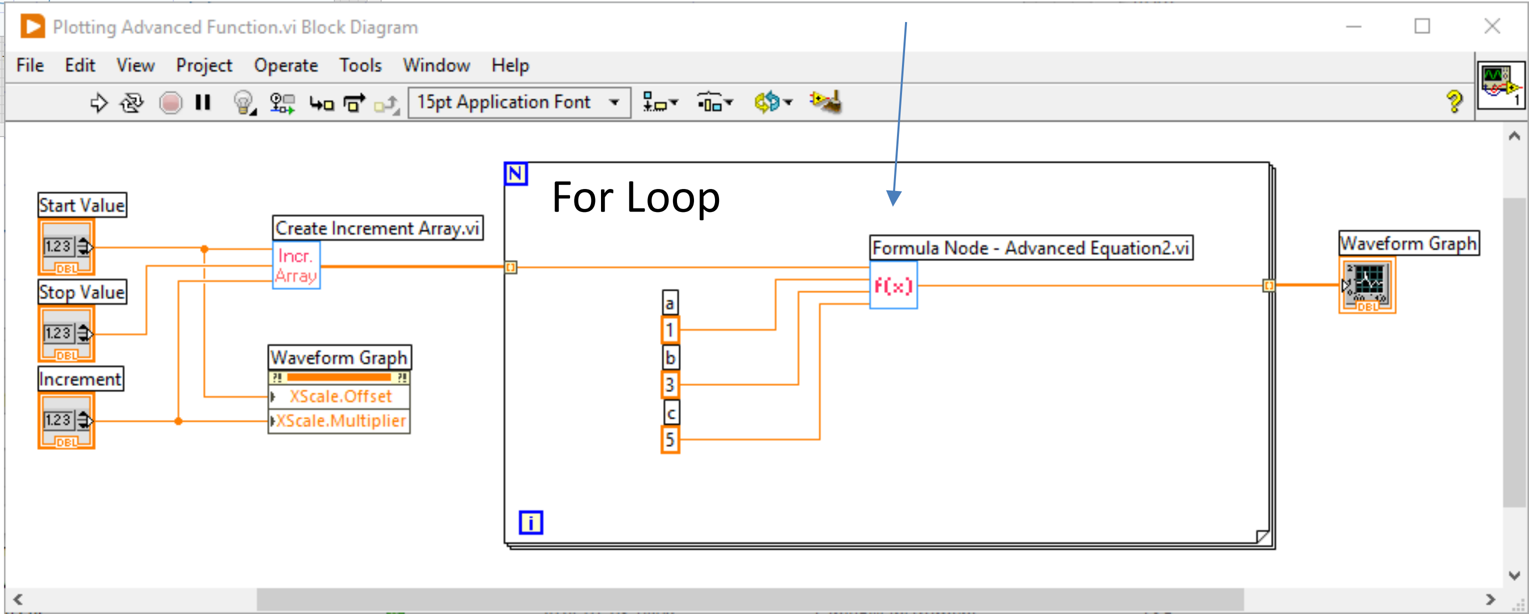
Plotting Advanced Equation



Plotting Advanced Function.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Waveform Graph
Plot 0
Amplitude
14
12
10
8
6
4
2
0
-2
-4
-2
-1

Start Value: -2
Stop Value: 2
Increment: 0.1

We have made a SubVI for the Advanced Equation which we use in our main program



Plotting Advanced Function.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Start Value
Stop Value
Increment
DBL
Create Increment Array.vi
Incr. Array
Waveform Graph
XScale.Offset
XScale.Multiplier
For Loop
N
i
Formula Node - Advanced Equation2.vi
f(x)
a
1
b
3
c
5
Waveform Graph

Simulation of 1.order system

- In this example we will use the following 1. order differential equation:

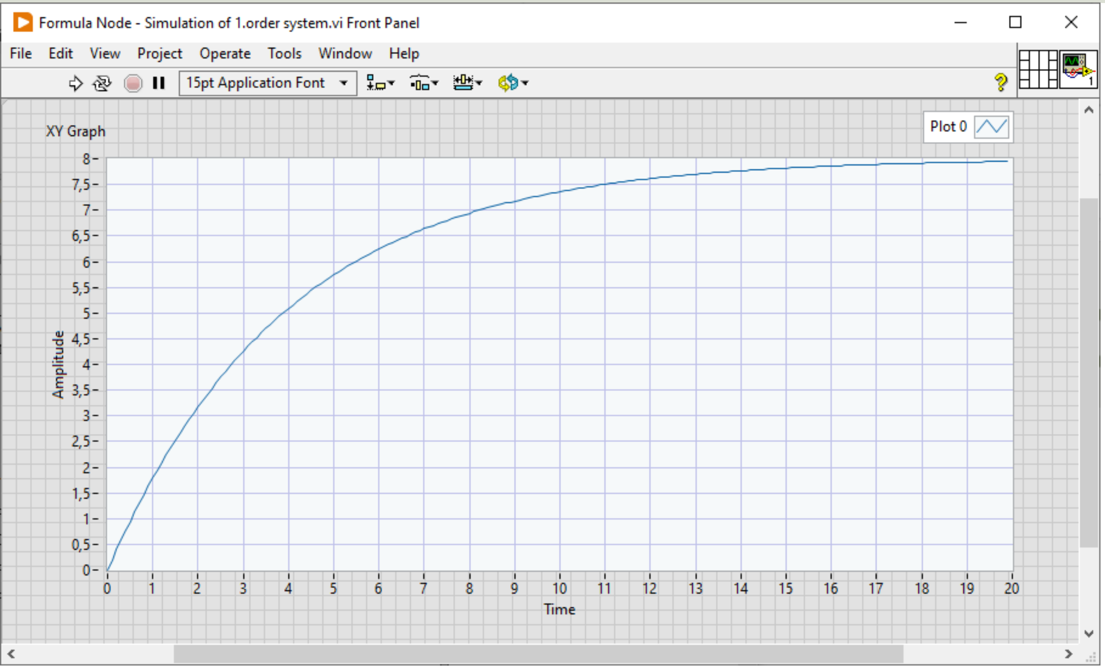
$$\dot{x} = -ax + bu$$

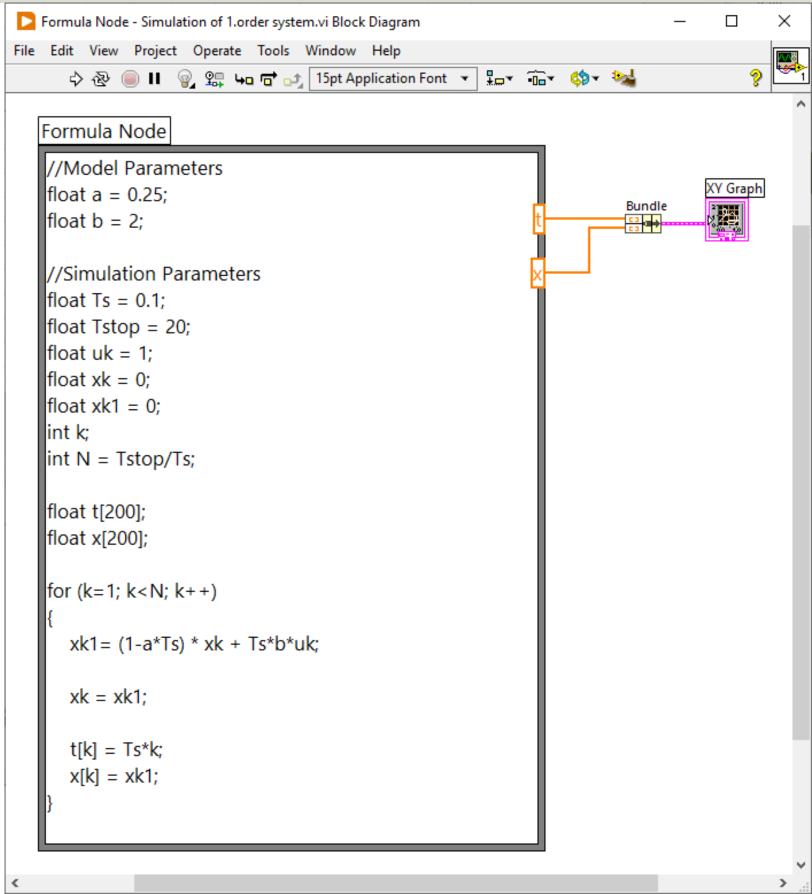
- We can set, e.g., $a = 0.25$ and $b = 2$ in the simulations
- We want to simulate this differential equation by applying a step in the input signal $u = 1$ at $t = 0s$
- Then we will observe the simulation results (Step Response) by plotting the results

Discrete Model

- We have the continuous differential equation: $\dot{x} = -ax + bu$
- We apply Euler: $\dot{x} \approx \frac{x(k+1)-x(k)}{T_s}$
- Then we get:
$$\frac{x(k+1) - x(k)}{T_s} = -ax(k) + bu(k)$$
- This gives the following discrete differential equation (difference equation):
- $x(k+1) = (1 - T_s a)x(k) + T_s bu(k)$
- This equation can easily be implemented in any text-based programming language or in a Formula Node in LabVIEW

LabVIEW Application




Formula Node - Simulation of 1.order system.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Formula Node
t
x
Bundle
XY Graph

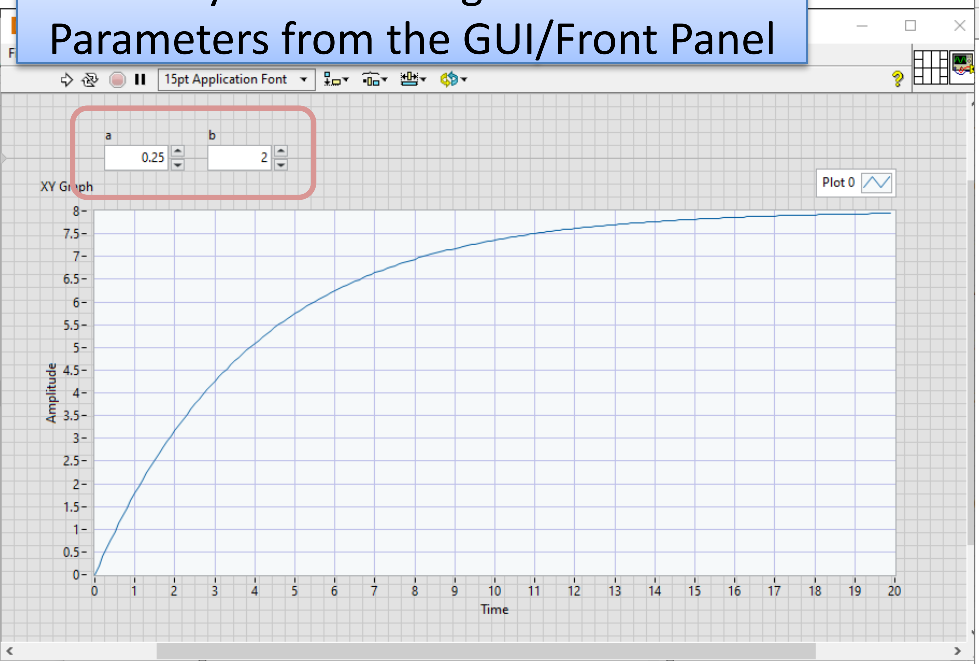
```
//Model Parameters
float a = 0.25;
float b = 2;

//Simulation Parameters
float Ts = 0.1;
float Tstop = 20;
float uk = 1;
float xk = 0;
float xk1 = 0;
int k;
int N = Tstop/Ts;

float t[200];
float x[200];

for (k=1; k<N; k++)
{
   xk1= (1-a*Ts) * xk + Ts*b*uk;

   xk = xk1;

   t[k] = Ts*k;
   x[k] = xk1;
}
```

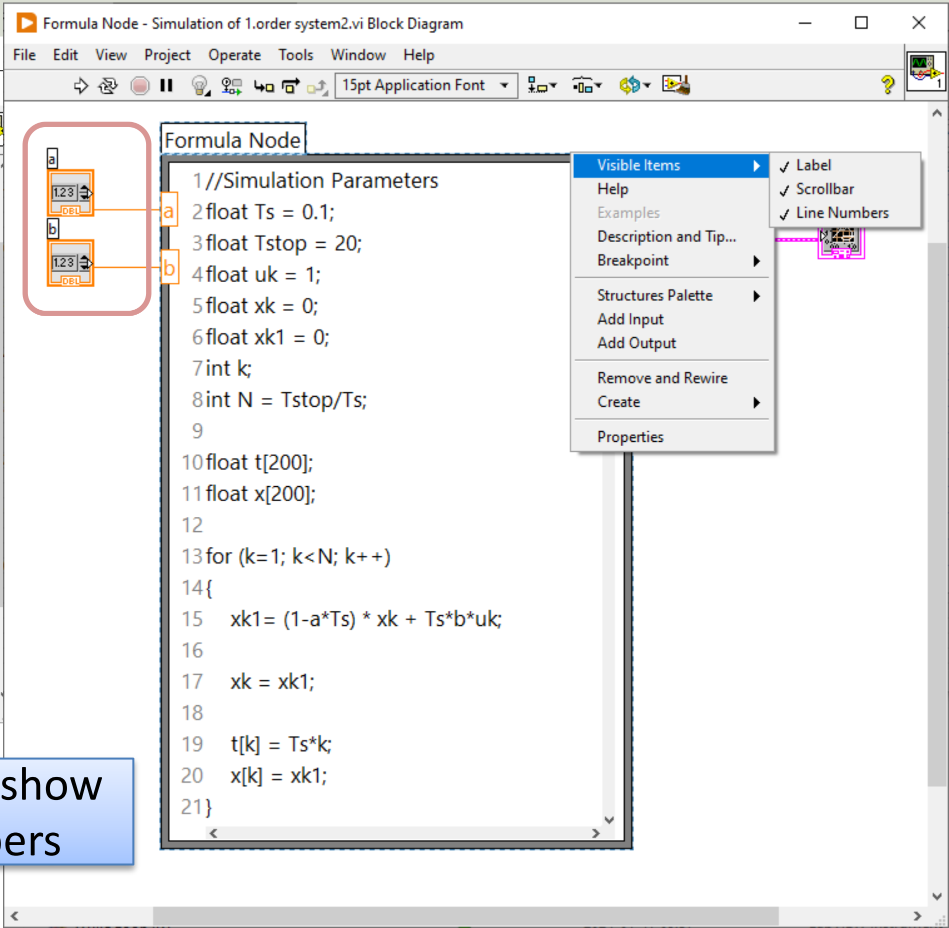
Script Code

```
//Model Parameters
float a = 0.25;
float b = 2;

//Simulation Parameters
float Ts = 0.1;
float Tstop = 20;
float uk = 1;
float xk = 0;
float xk1 = 0;
int k;
int N = Tstop/Ts;

float t[200];
float x[200];

for (k=1; k<N; k++)
{
    xk1= (1-a*Ts) * xk + Ts*b*uk;

    xk = xk1;

    t[k] = Ts*k;
    x[k] = xk1;
}
```

Improvements

Now you can change the Model Parameters from the GUI/Front Panel



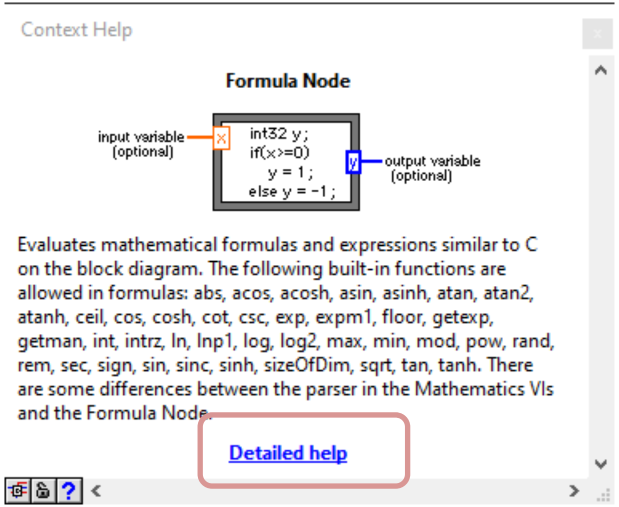


```
 1//Simulation Parameters
 2float Ts = 0.1;
 3float Tstop = 20;
 4float uk = 1;
 5float xk = 0;
 6float xk1 = 0;
 7int k;
 8int N = Tstop/Ts;
 9
10float t[200];
11float x[200];
12
13for (k=1; k<N; k++)
14{
15   xk1= (1-a*Ts) * xk + Ts*b*uk;
16
17   xk = xk1;
18
19   t[k] = Ts*k;
20   x[k] = xk1;
21}
```

You can also show Line Numbers

Detailed Help

Ctrl + H



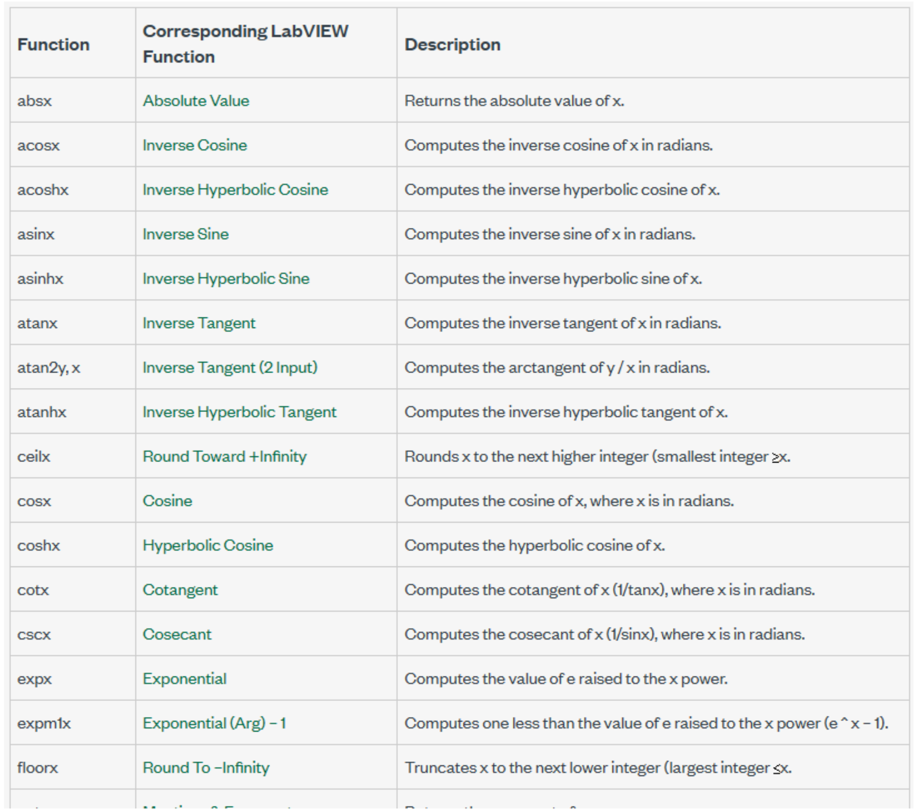
Context Help

Formula Node

```
int32 y;
if(x>=0)
  y = 1;
else y = -1;
```

Evaluates mathematical formulas and expressions similar to C on the block diagram. The following built-in functions are allowed in formulas: abs, acos, acosh, asin, asinh, atan, atan2, atanh, ceil, cos, cosh, cot, csc, exp, expm1, floor, getexp, getman, int, intrz, ln, lnp1, log, log2, max, min, mod, pow, rand, rem, sec, sign, sin, sinc, sinh, sizeOfDim, sqrt, tan, tanh. There are some differences between the parser in the Mathematics VIs and the Formula Node.

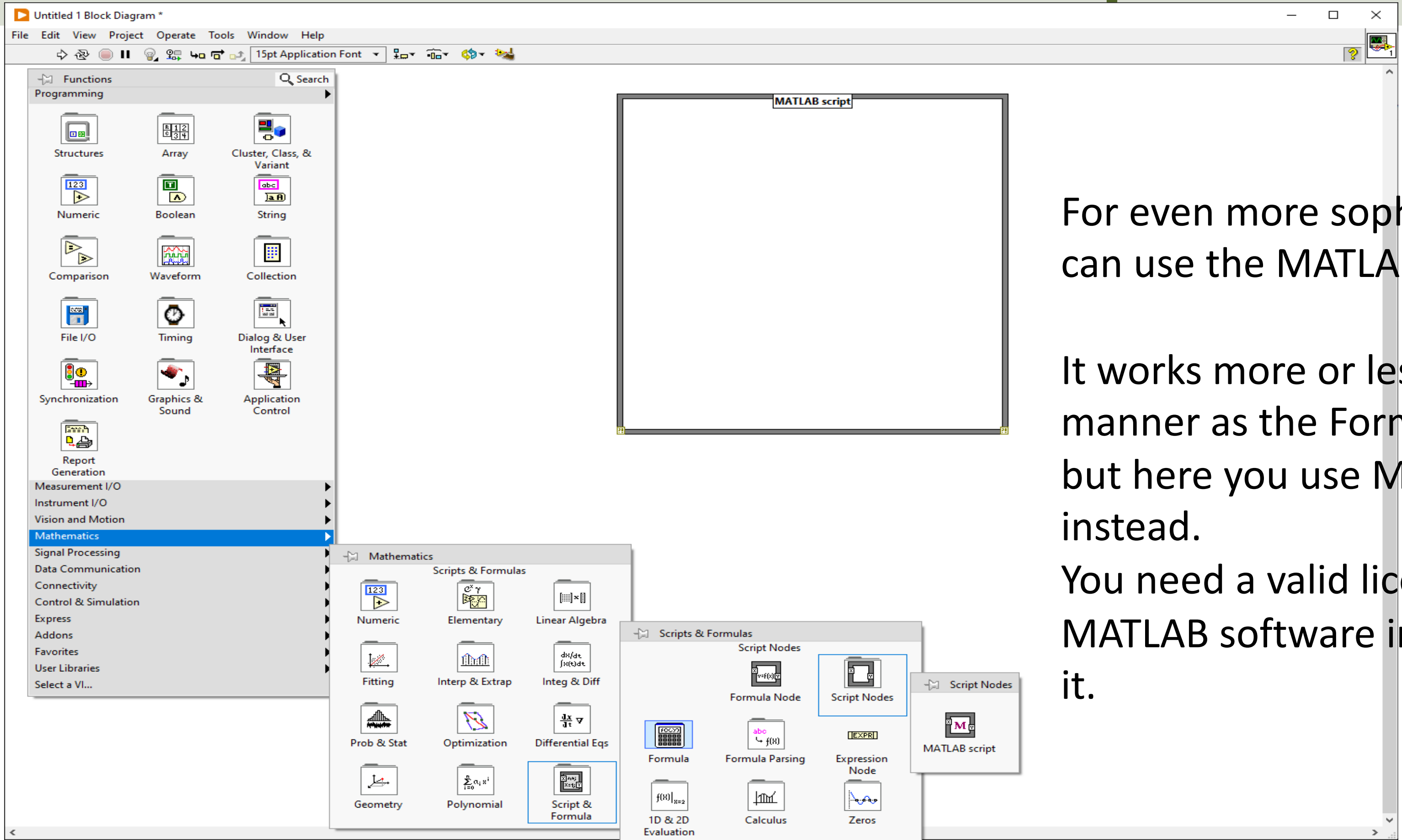
Detailed help



Function	Corresponding LabVIEW Function	Description
absx	Absolute Value	Returns the absolute value of x.
acosx	Inverse Cosine	Computes the inverse cosine of x in radians.
acoshx	Inverse Hyperbolic Cosine	Computes the inverse hyperbolic cosine of x.
asinx	Inverse Sine	Computes the inverse sine of x in radians.
asinhx	Inverse Hyperbolic Sine	Computes the inverse hyperbolic sine of x.
atanx	Inverse Tangent	Computes the inverse tangent of x in radians.
atan2y, x	Inverse Tangent (2 Input)	Computes the arctangent of y / x in radians.
atanhx	Inverse Hyperbolic Tangent	Computes the inverse hyperbolic tangent of x.
ceilx	Round Toward +Infinity	Rounds x to the next higher integer (smallest integer ≥x.
cosx	Cosine	Computes the cosine of x, where x is in radians.
coshx	Hyperbolic Cosine	Computes the hyperbolic cosine of x.
cotx	Cotangent	Computes the cotangent of x (1/tanx), where x is in radians.
cscx	Cosecant	Computes the cosecant of x (1/sinx), where x is in radians.
expx	Exponential	Computes the value of e raised to the x power.
expm1x	Exponential (Arg) - 1	Computes one less than the value of e raised to the x power (e ^ x - 1).
floorx	Round To -Infinity	Truncates x to the next lower integer (largest integer ≤x.

+++

MATLAB Script



For even more sophisticated you can use the MATLAB Script .

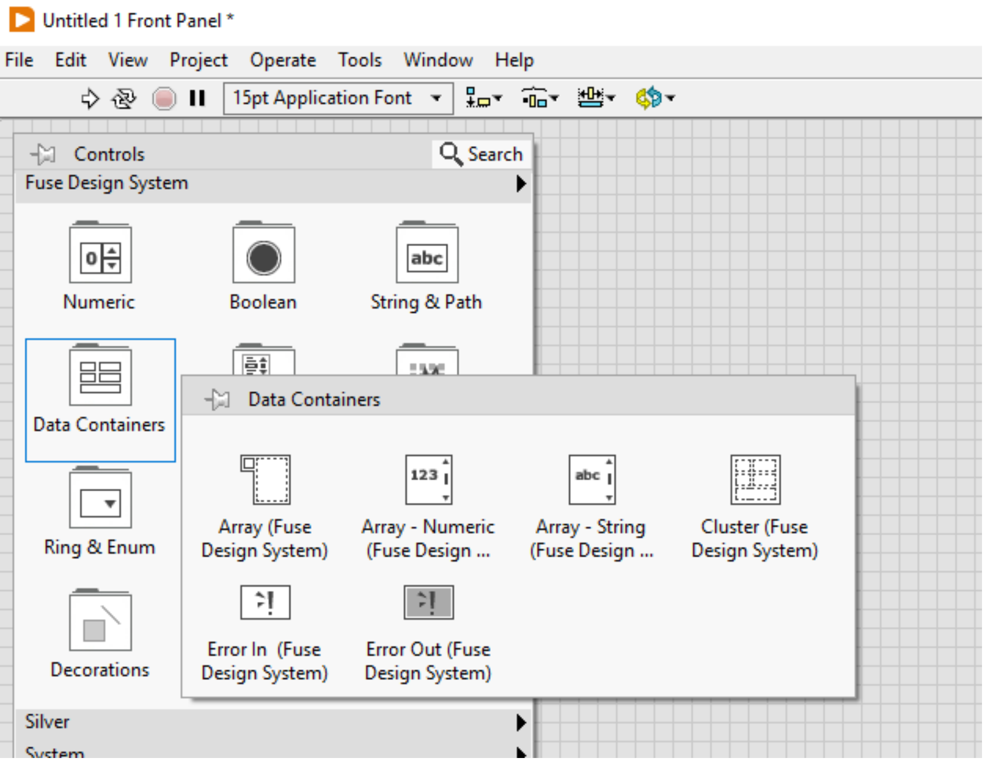
It works more or less in the same manner as the Formula Node, but here you use MATLAB syntax instead.
You need a valid license od the MATLAB software in order to use it.

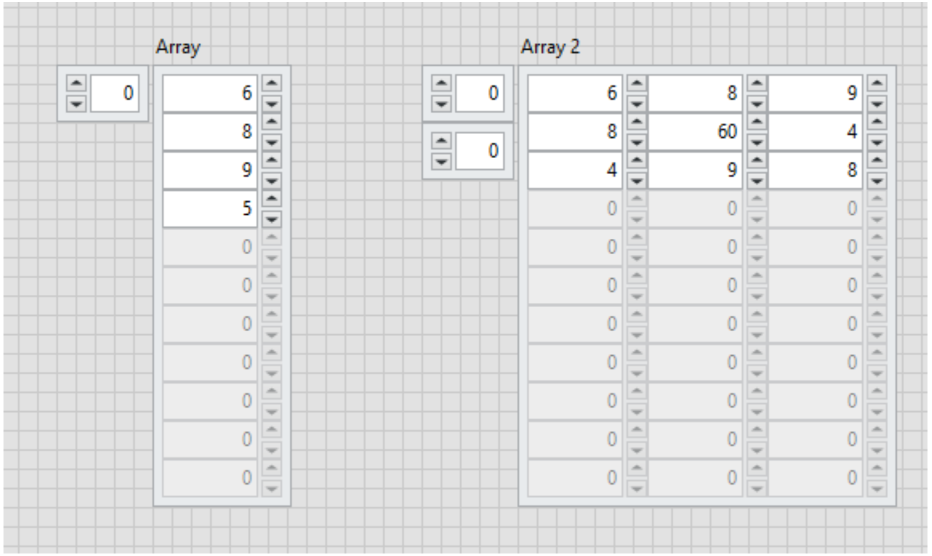
LabVIEW Fundamentals

Arrays in LabVIEW

Hans-Petter Halvorsen

Arrays in LabVIEW


Untitled 1 Front Panel *
File Edit View Project Operate Tools Window Help
15pt Application Font
Controls
Search
Fuse Design System
Numeric
Boolean
String & Path
Data Containers
Ring & Enum
Decorations
Data Containers
Array (Fuse Design System)
Array - Numeric (Fuse Design ...
Array - String (Fuse Design ...
Cluster (Fuse Design System)
Error In (Fuse Design System)
Error Out (Fuse Design System)
Silver
System



Array (index 0):

6
8
9
5
0
0
0
0
0
0
0

Array 2 (indices 0, 0):

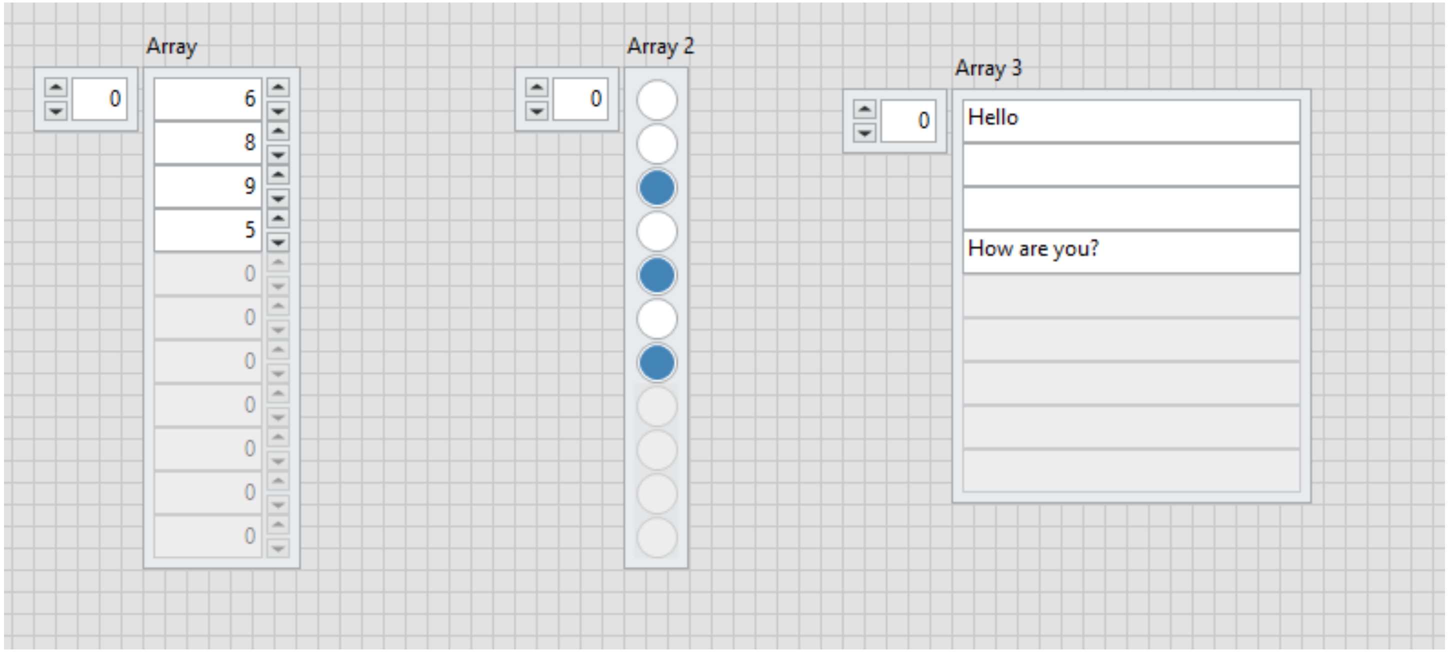
6	8	9
8	60	4
4	9	8
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Arrays of different Data Types

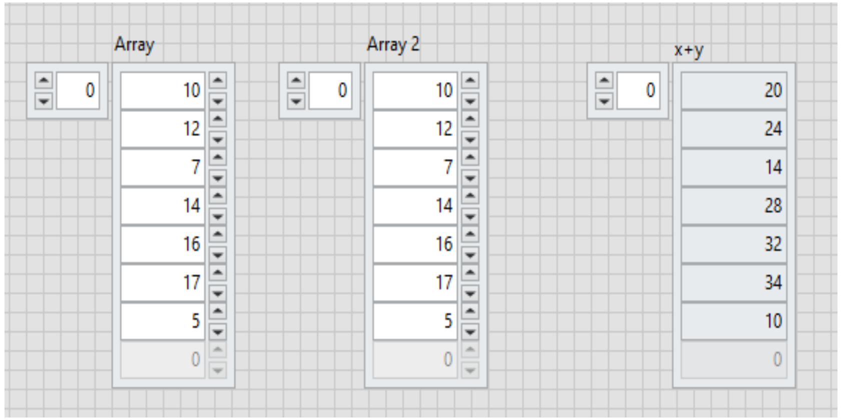
Numeric

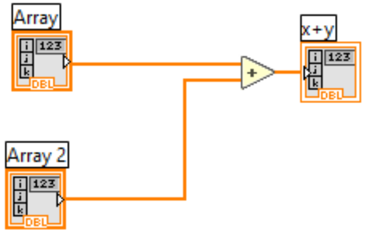
Boolean

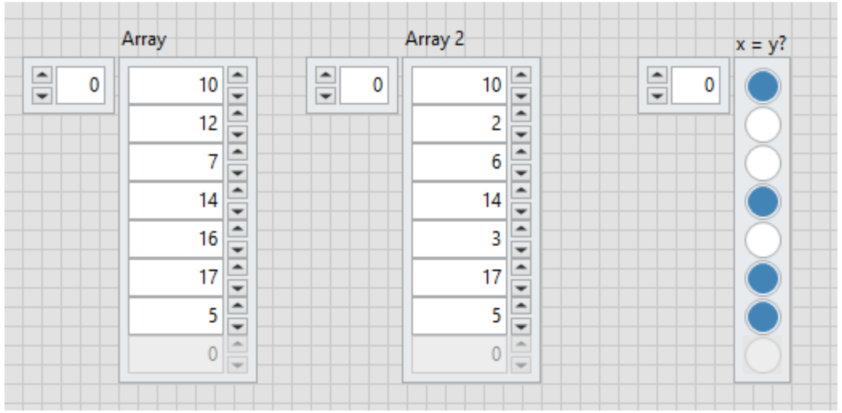
String

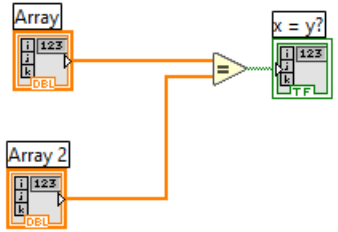


Most of the built-in Functions supports Arrays

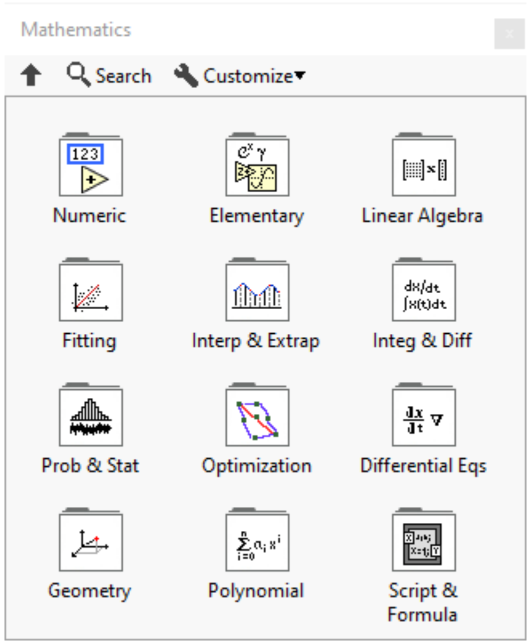


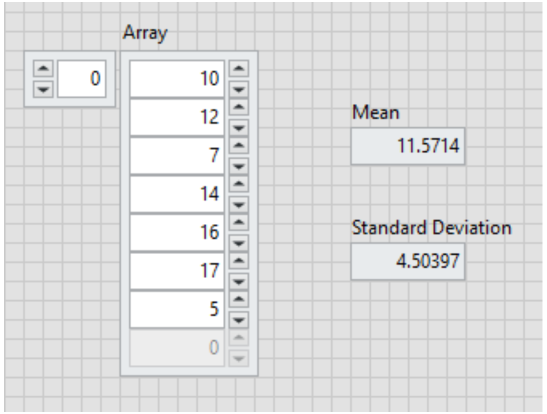


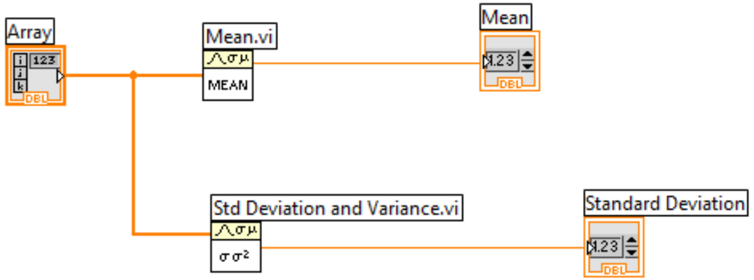




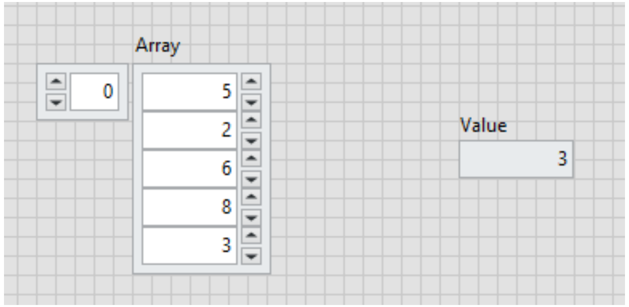
Mathematics Array Functions

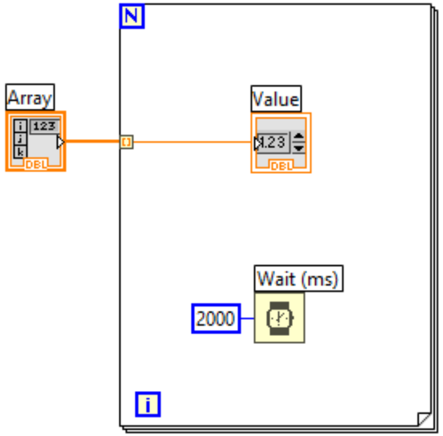


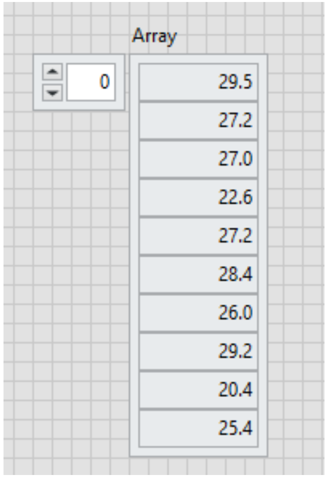


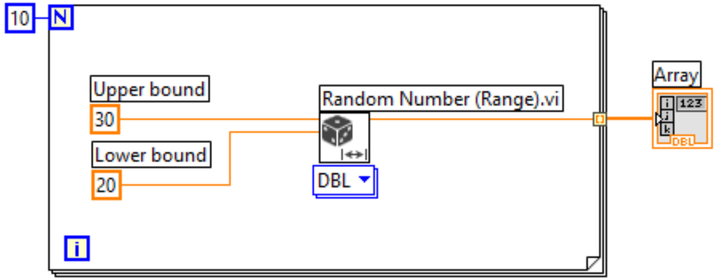


Arrays and For Loop

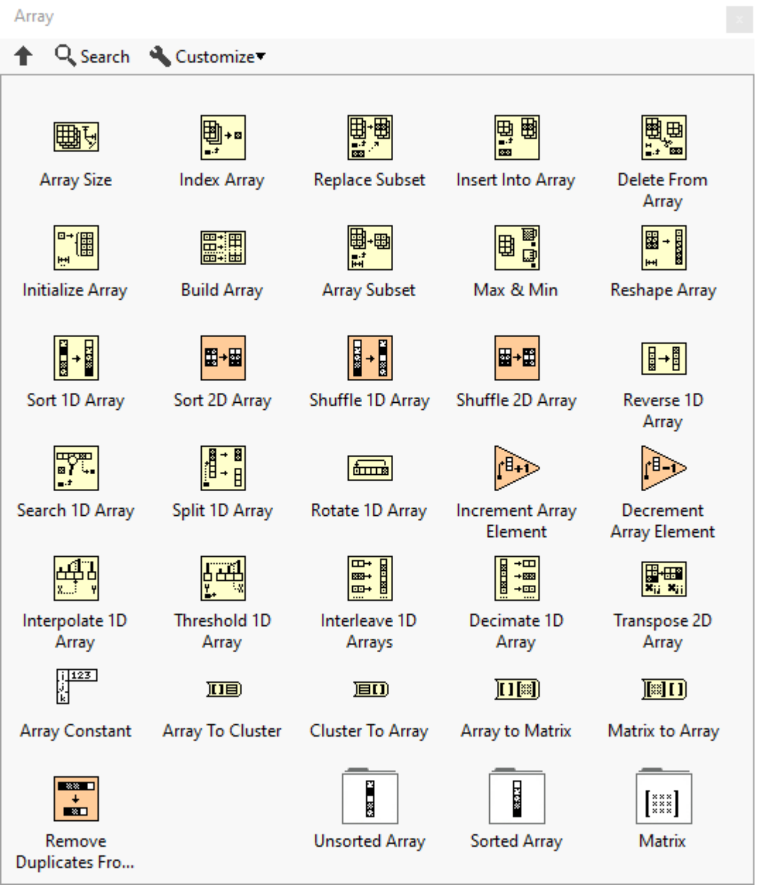




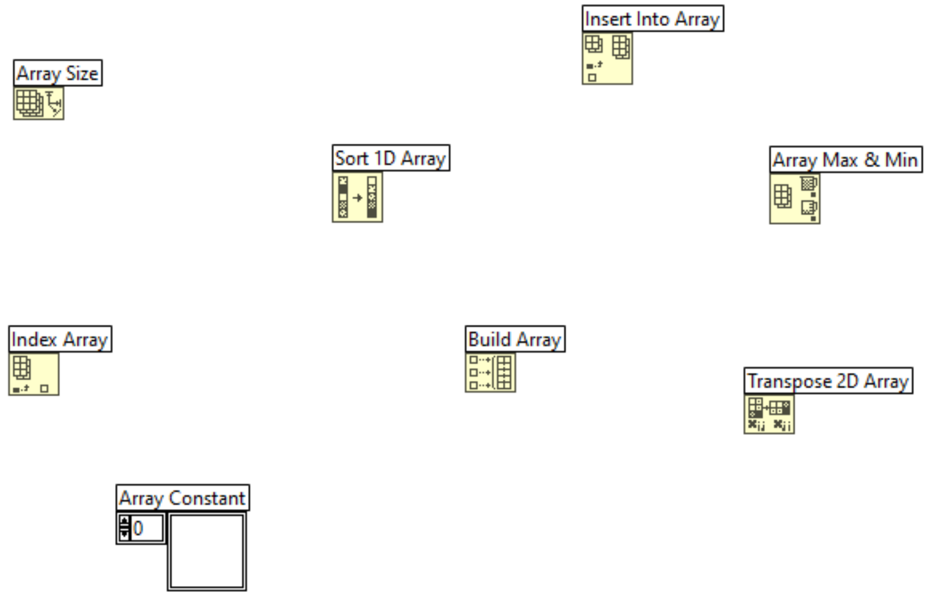




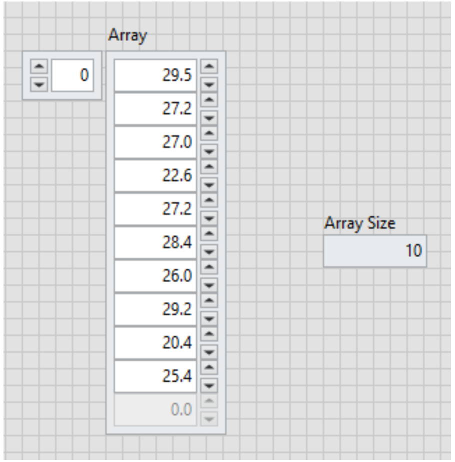
Array Functions in LabVIEW

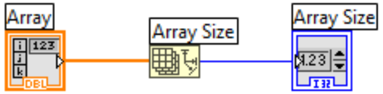


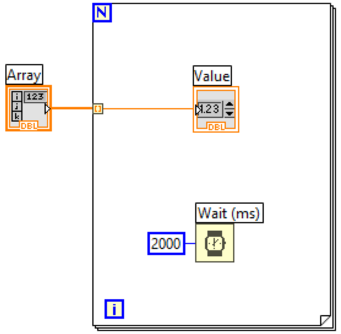
Some of the most used Array Functions:

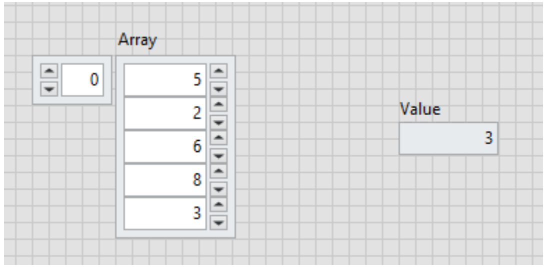


Array Functions in LabVIEW

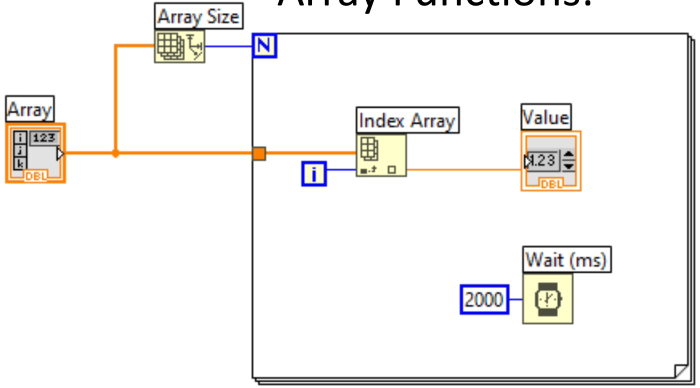




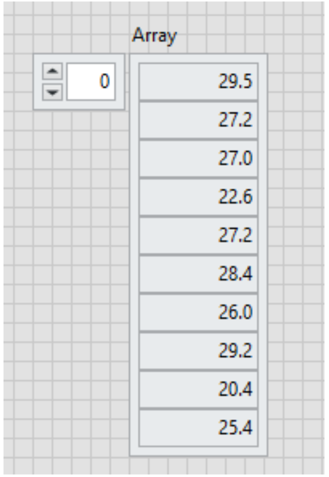


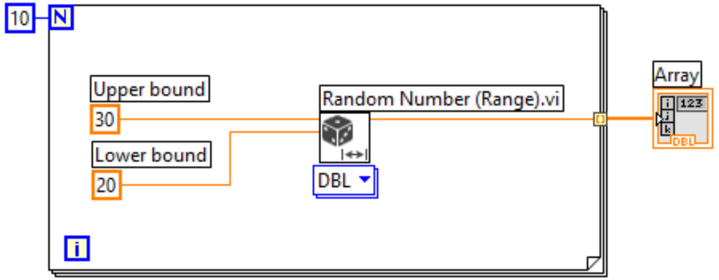


Same code using some Array Functions:

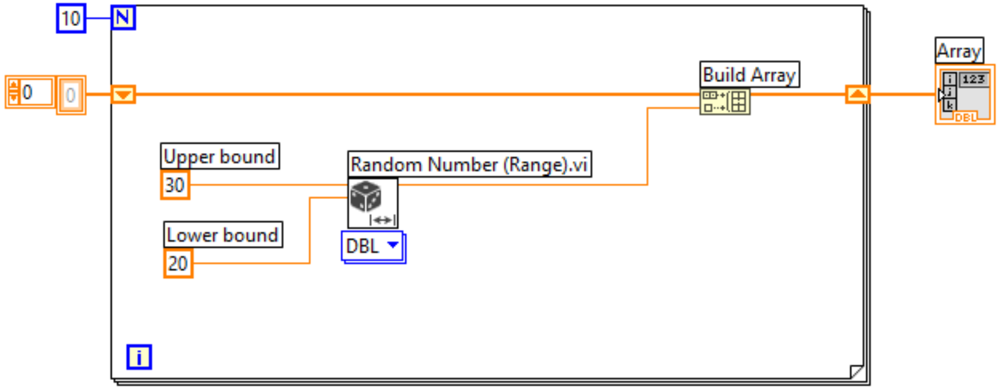


Array Functions in LabVIEW





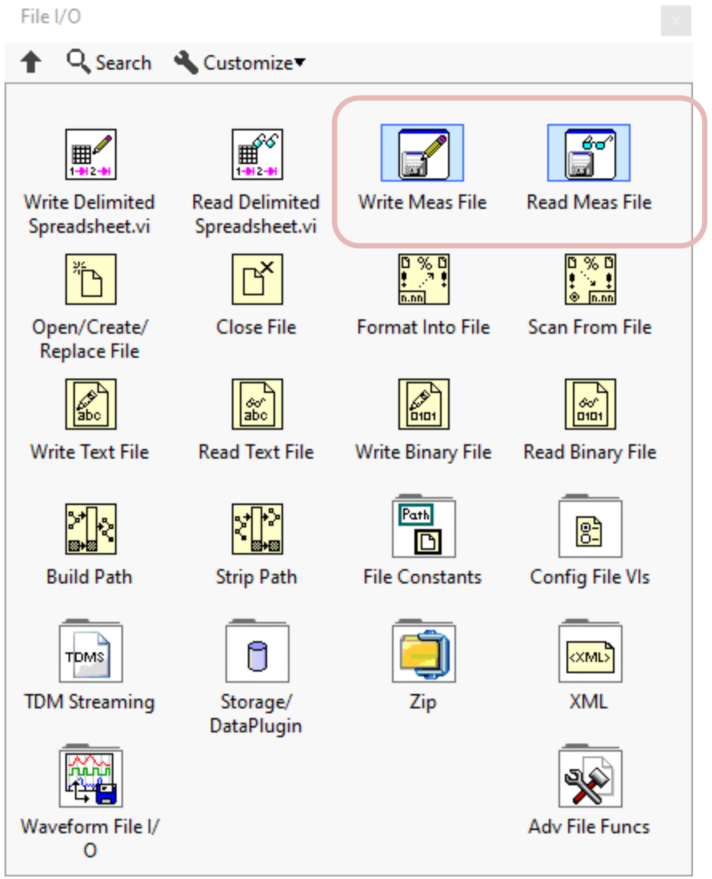
Same code using **Build Array**:



LabVIEW Fundamentals

Write and Read Data Files in LabVIEW

Writing/Reading to/from Files


File I/O

Search

Customize

Write Delimited Spreadsheet.vi

Read Delimited Spreadsheet.vi

Write Meas File

Read Meas File

Open/Create/ Replace File

Close File

Format Into File

Scan From File

Write Text File

Read Text File

Write Binary File

Read Binary File

Build Path

Strip Path

File Constants

Config File VIs

TDM Streaming

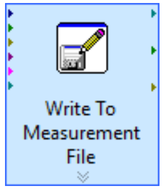
Storage/ DataPlugin

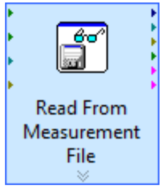
Zip

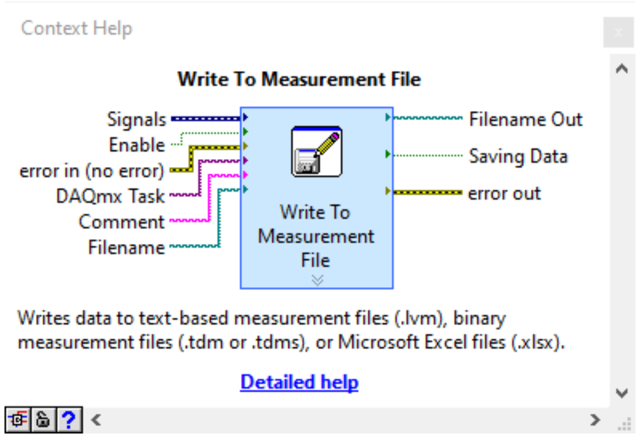
XML

Waveform File I/ O

Adv File Funcs



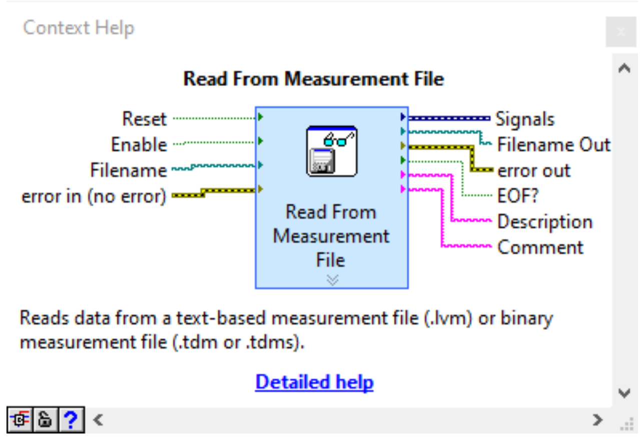



Context Help

Write To Measurement File

Writes data to text-based measurement files (.lvm), binary measurement files (.tdm or .tdms), or Microsoft Excel files (.xlsx).

Detailed help

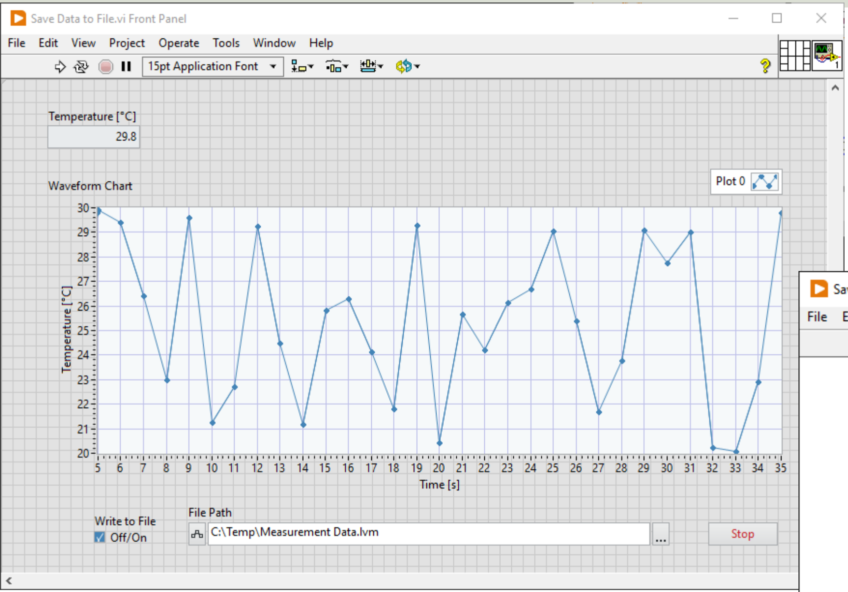

Context Help

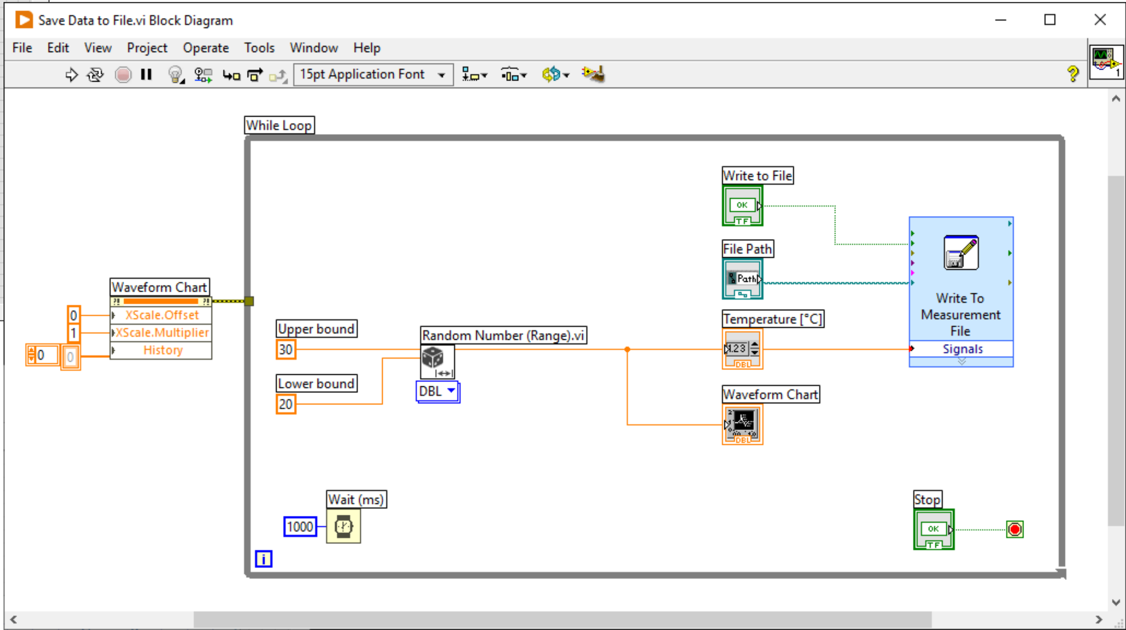
Read From Measurement File

Reads data from a text-based measurement file (.lvm) or binary measurement file (.tdm or .tdms).

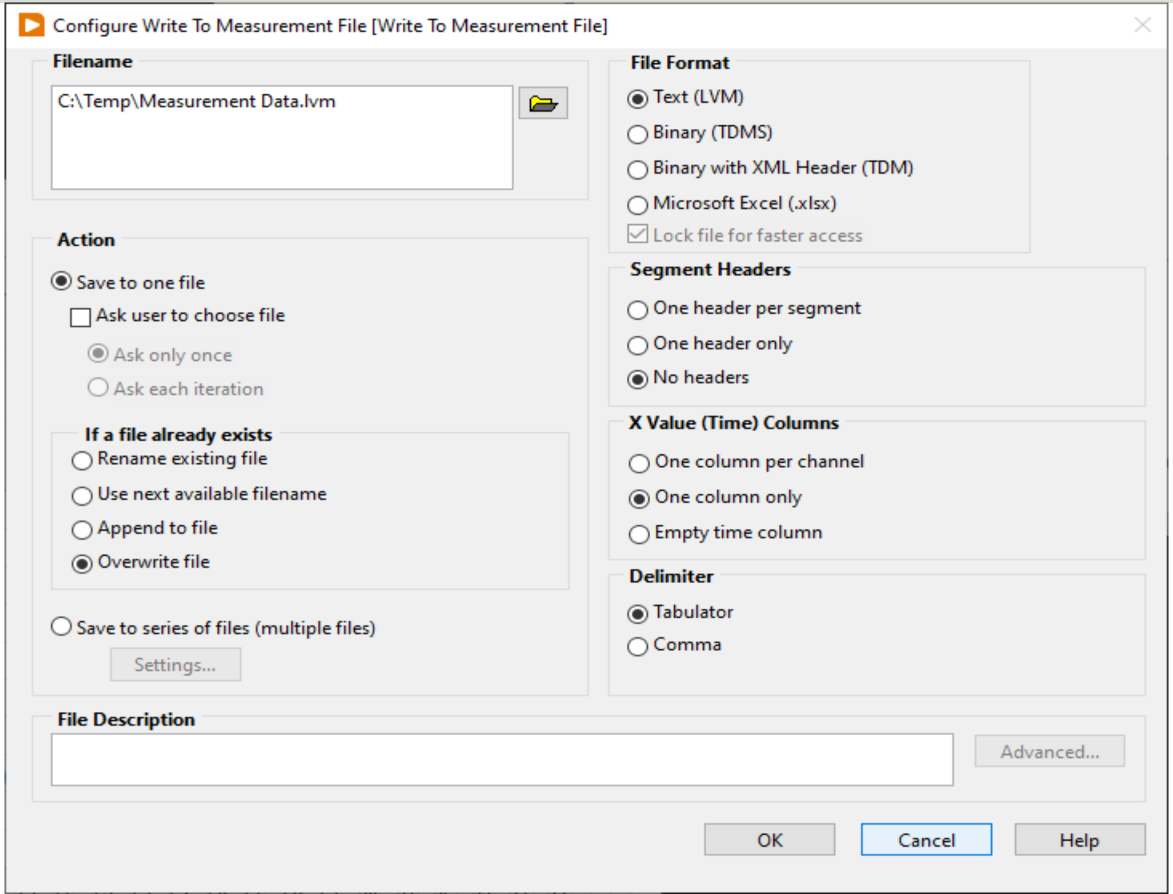
Detailed help

Writing Data to File Example





Configuration



Configure Write To Measurement File [Write To Measurement File]

Filename

C:\Temp\Measurement Data.lvm

Action

[✓ Save to one file]

Ask user to choose file

[✓ Ask only once]

Ask each iteration

If a file already exists

Rename existing file

Use next available filename

Append to file

[✓ Overwrite file]

Save to series of files (multiple files)

Settings...

File Description

Advanced...

File Format

[✓ Text (LVM)]

Binary (TDMS)

Binary with XML Header (TDM)

Microsoft Excel (.xlsx)

[✓ Lock file for faster access]

Segment Headers

One header per segment

One header only

[✓ No headers]

X Value (Time) Columns

One column per channel

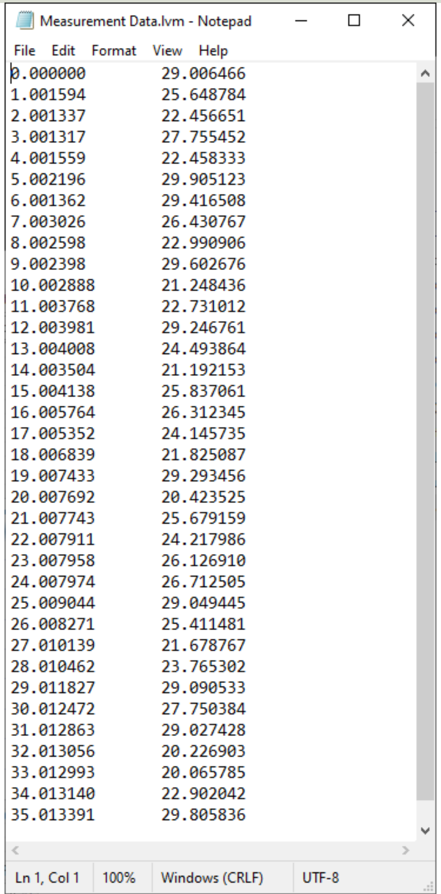
[✓ One column only]

Empty time column

Delimiter

[✓ Tabulator]

Comma

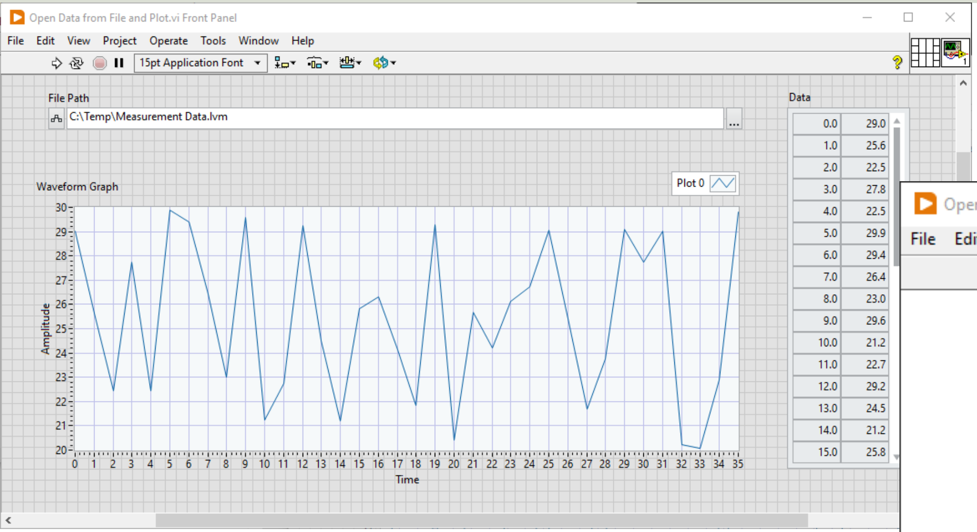


Measurement Data.lvm - Notepad

```
0.000000	29.006466
1.001594	25.648784
2.001337	22.456651
3.001317	27.755452
4.001559	22.458333
5.002196	29.905123
6.001362	29.416508
7.003026	26.430767
8.002598	22.990906
9.002398	29.602676
10.002888	21.248436
11.003768	22.731012
12.003981	29.246761
13.004008	24.493864
14.003504	21.192153
15.004138	25.837061
16.005764	26.312345
17.005352	24.145735
18.006839	21.825087
19.007433	29.293456
20.007692	20.423525
21.007743	25.679159
22.007911	24.217986
23.007958	26.126910
24.007974	26.712505
25.009044	29.049445
26.008271	25.411481
27.010139	21.678767
28.010462	23.765302
29.011827	29.090533
30.012472	27.750384
31.012863	29.027428
32.013056	20.226903
33.012993	20.065785
34.013140	22.902042
35.013391	29.805836
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8

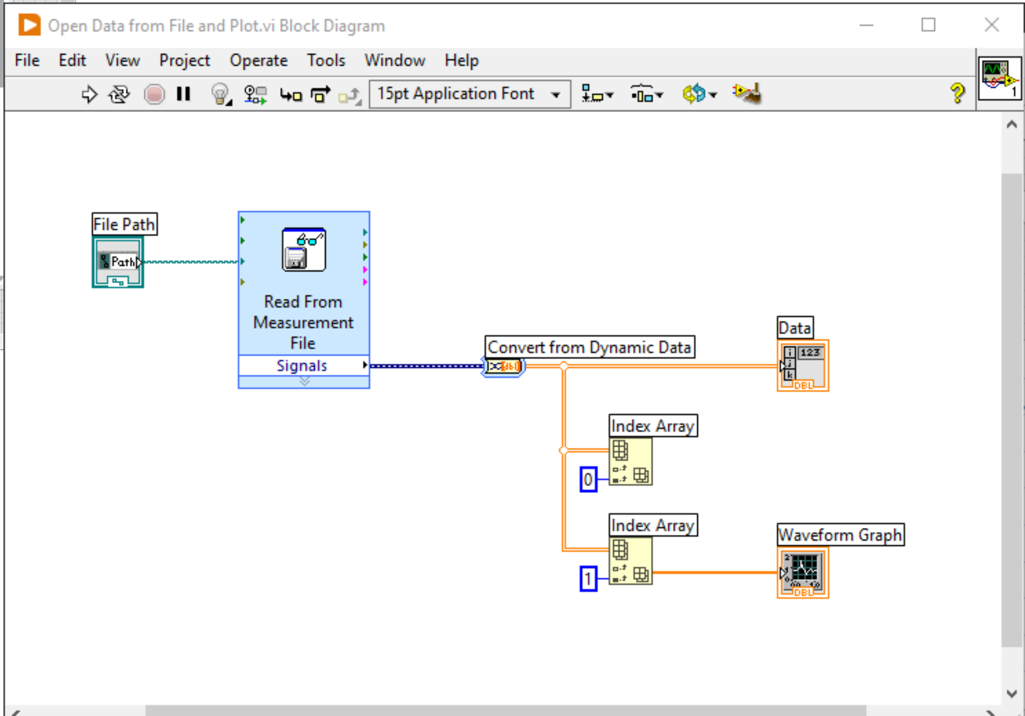
Open Data from File Example


Open Data from File and Plot.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Waveform Graph
Plot 0
Amplitude
Time

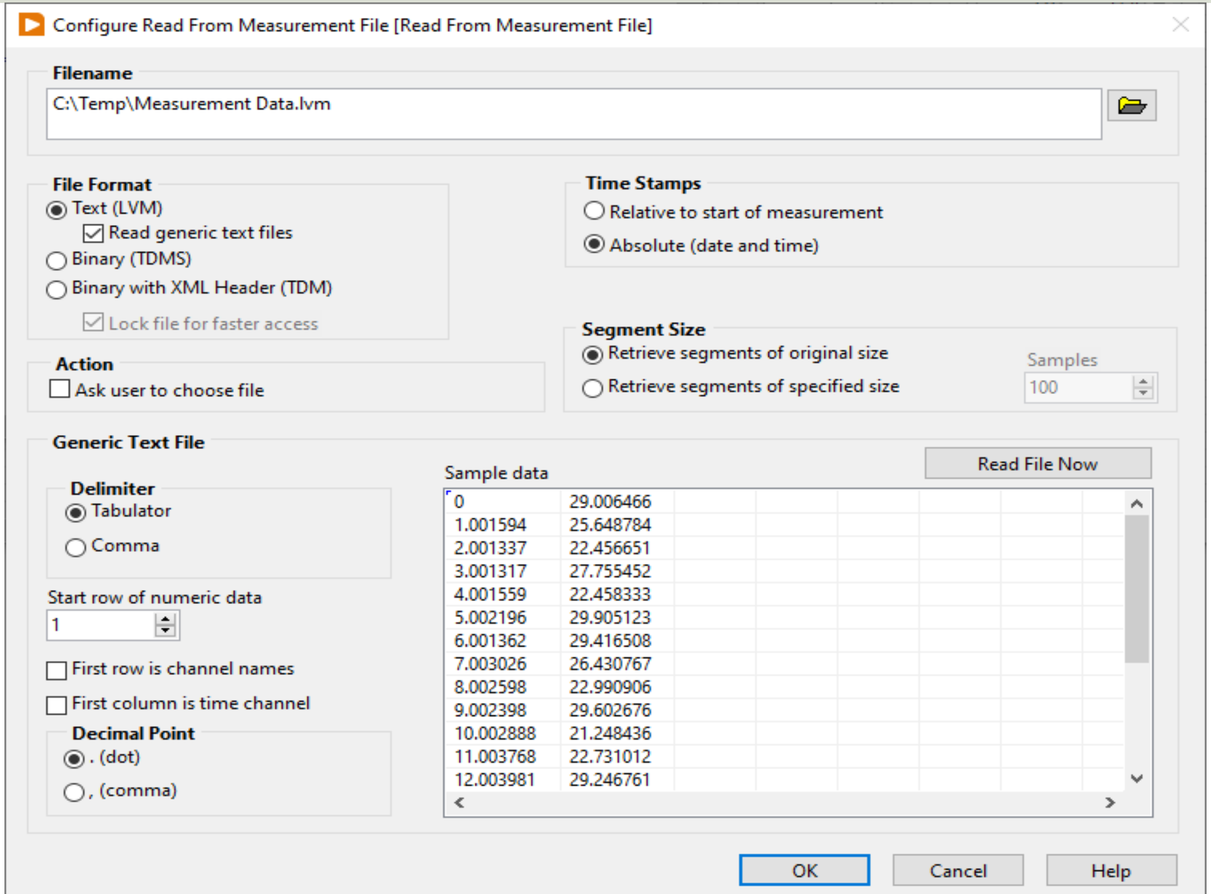
File Path: C:\Temp\Measurement Data.lvm

Data

0.0	29.0
1.0	25.6
2.0	22.5
3.0	27.8
4.0	22.5
5.0	29.9
6.0	29.4
7.0	26.4
8.0	23.0
9.0	29.6
10.0	21.2
11.0	22.7
12.0	29.2
13.0	24.5
14.0	21.2
15.0	25.8


Open Data from File and Plot.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
File Path
Path
Read From Measurement File
Signals
Convert from Dynamic Data
Data
Index Array
0
Index Array
1
Waveform Graph

Configuration



Configure Read From Measurement File [Read From Measurement File]

Filename

C:\Temp\Measurement Data.lvm

File Format

- [✓ Text (LVM)]
  - [✓ Read generic text files]
- Binary (TDMS)
- Binary with XML Header (TDM)
  - [✓ Lock file for faster access]

Time Stamps

- Relative to start of measurement
- [✓ Absolute (date and time)]

Action

- Ask user to choose file

Segment Size

- [✓ Retrieve segments of original size]
- Retrieve segments of specified size

Samples: 100

Generic Text File

Delimiter

- [✓ Tabulator]
- Comma

Start row of numeric data: 1

- First row is channel names
- First column is time channel

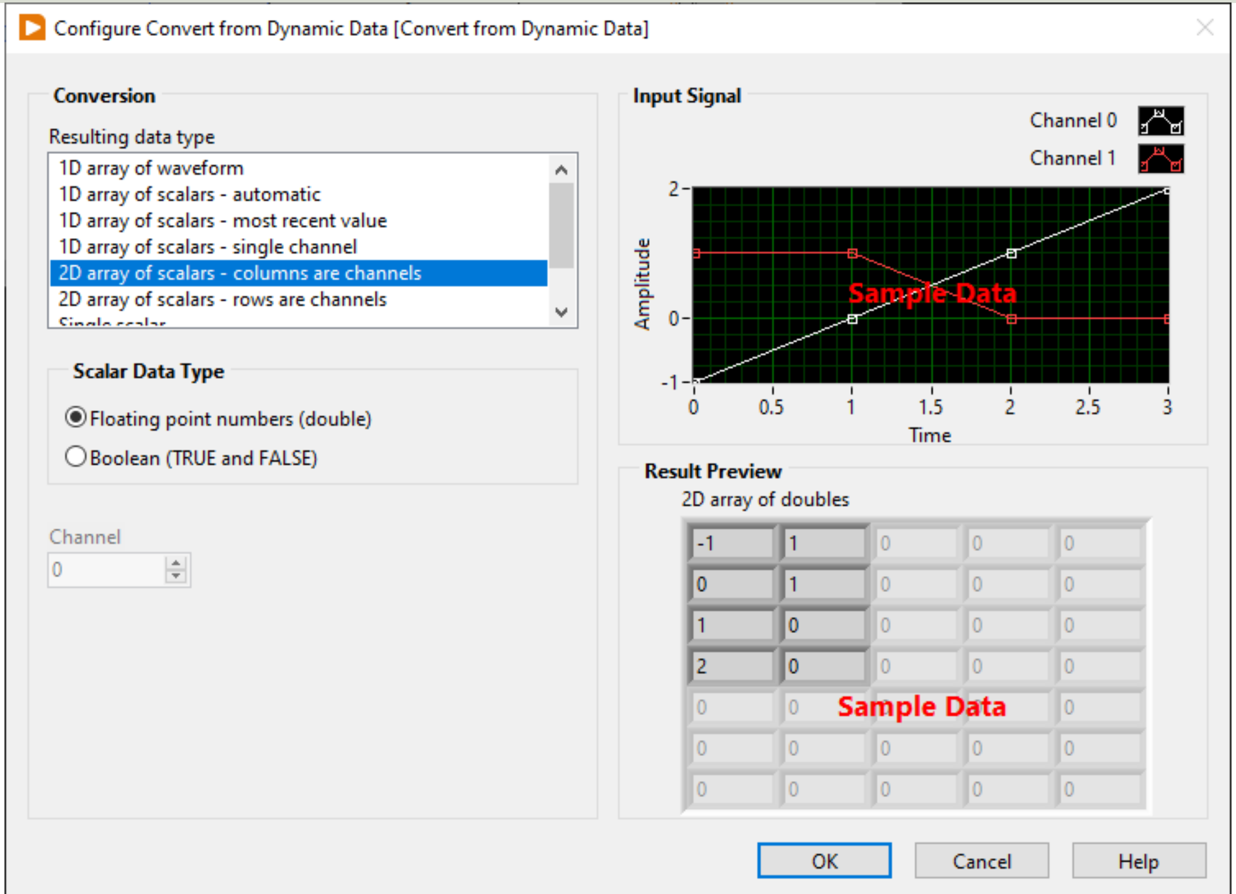
Decimal Point

- [✓ . (dot)]
- , (comma)

Sample data

0	29.006466
1.001594	25.648784
2.001337	22.456651
3.001317	27.755452
4.001559	22.458333
5.002196	29.905123
6.001362	29.416508
7.003026	26.430767
8.002598	22.990906
9.002398	29.602676
10.002888	21.248436
11.003768	22.731012
12.003981	29.246761

Convert from Dynamic Data



Configure Convert from Dynamic Data [Convert from Dynamic Data]

Conversion

Resulting data type

- 1D array of waveform
- 1D array of scalars - automatic
- 1D array of scalars - most recent value
- 1D array of scalars - single channel
- [✓ 2D array of scalars - columns are channels]
- 2D array of scalars - rows are channels
- Single scalar

Scalar Data Type

[✓ Floating point numbers (double)]
Boolean (TRUE and FALSE)

Channel
0

Input Signal

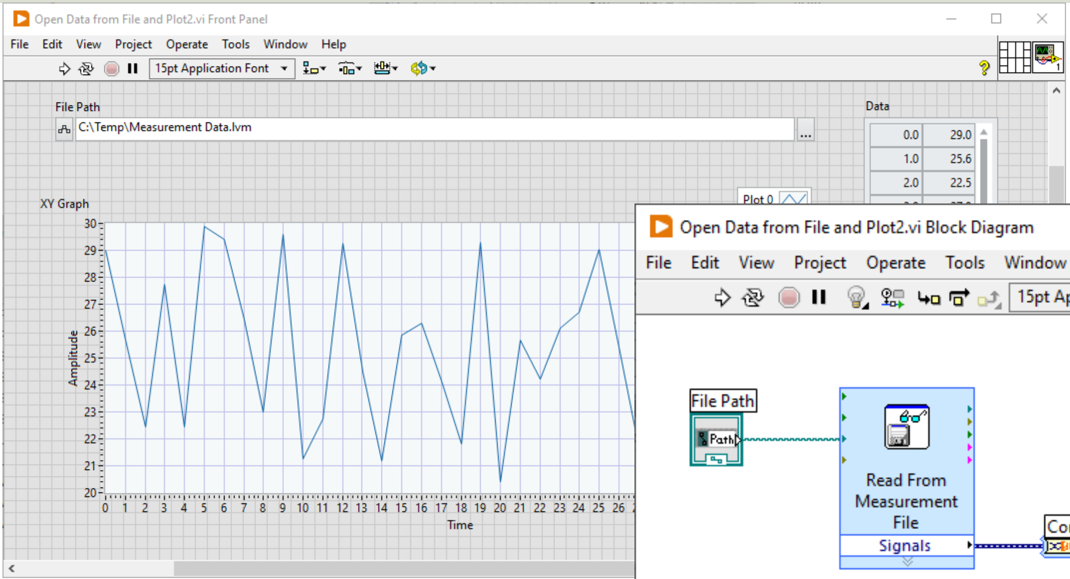
Result Preview

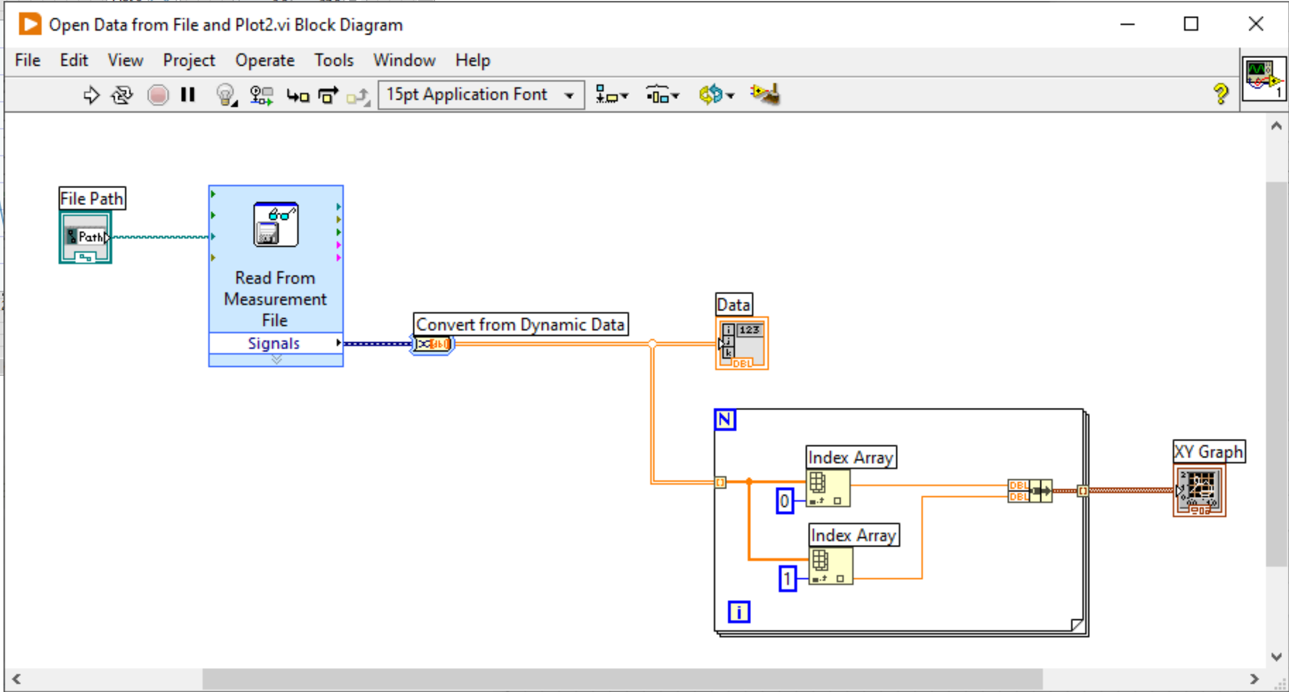
2D array of doubles

-1	1	0	0	0
0	1	0	0	0
1	0	0	0	0
2	0	0	0	0
0	0	Sample Data		0
0	0	0	0	0
0	0	0	0	0

OK Cancel Help

Open Data from File – XY Graph


Open Data from File and Plot2.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
File Path
C:\Temp\Measurement Data.lvm
Data
0.0 29.0
1.0 25.6
2.0 22.5
Plot 0
XY Graph
Amplitude
Time


Open Data from File and Plot2.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
File Path
Path
Read From Measurement File
Signals
Convert from Dynamic Data
Data
N
Index Array
0
Index Array
1
i
XY Graph

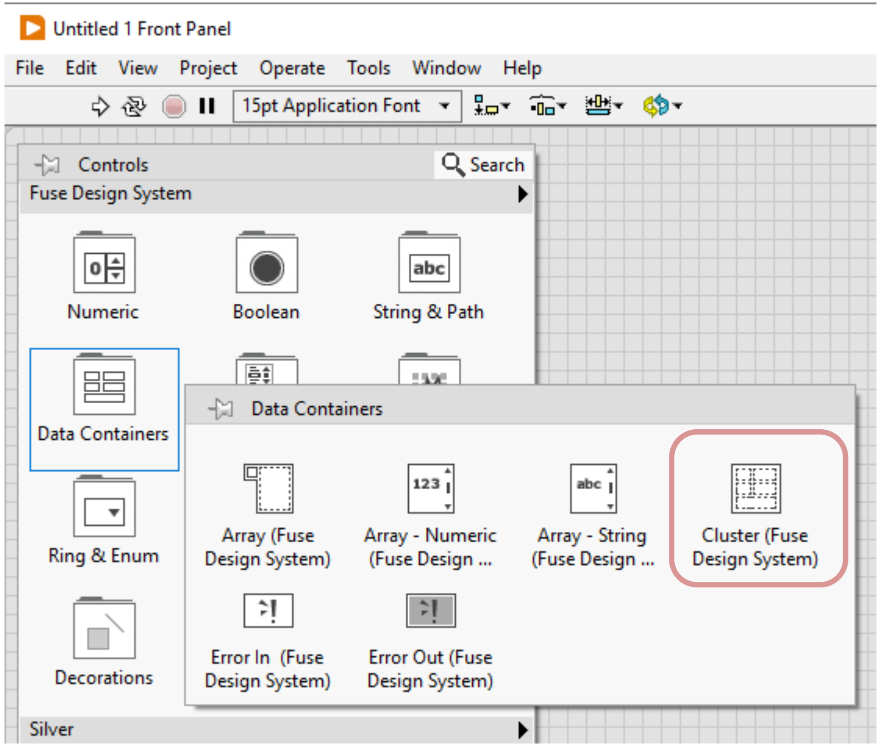
LabVIEW Fundamentals

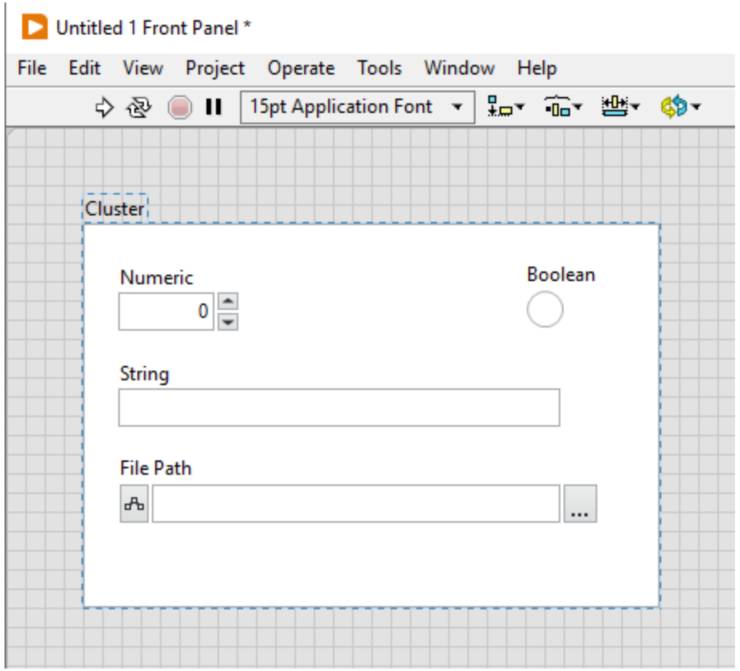
Clusters in LabVIEW

Hans-Petter Halvorsen

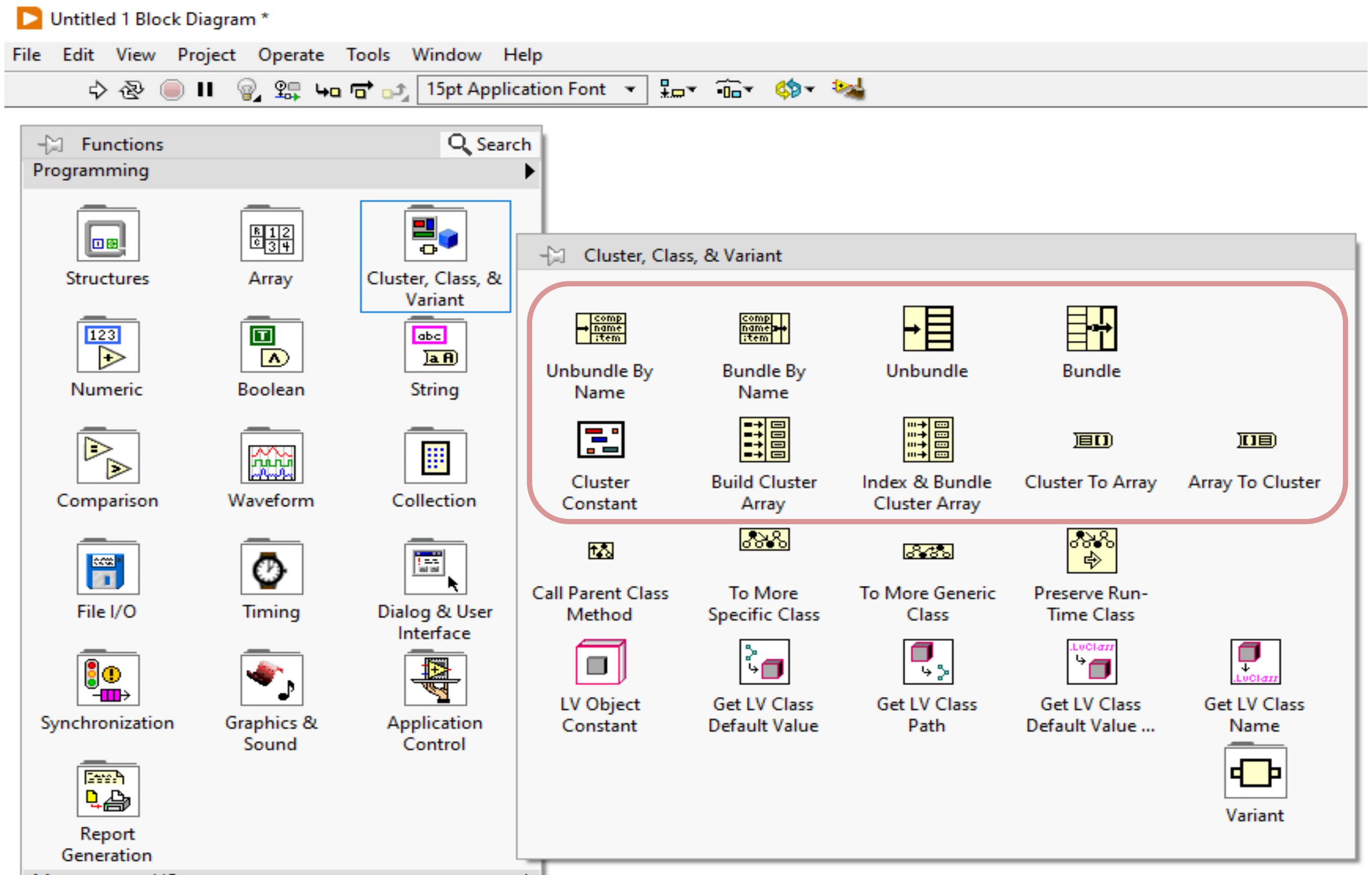
Clusters

A cluster is a container where you can add different controls and they can also have different data types

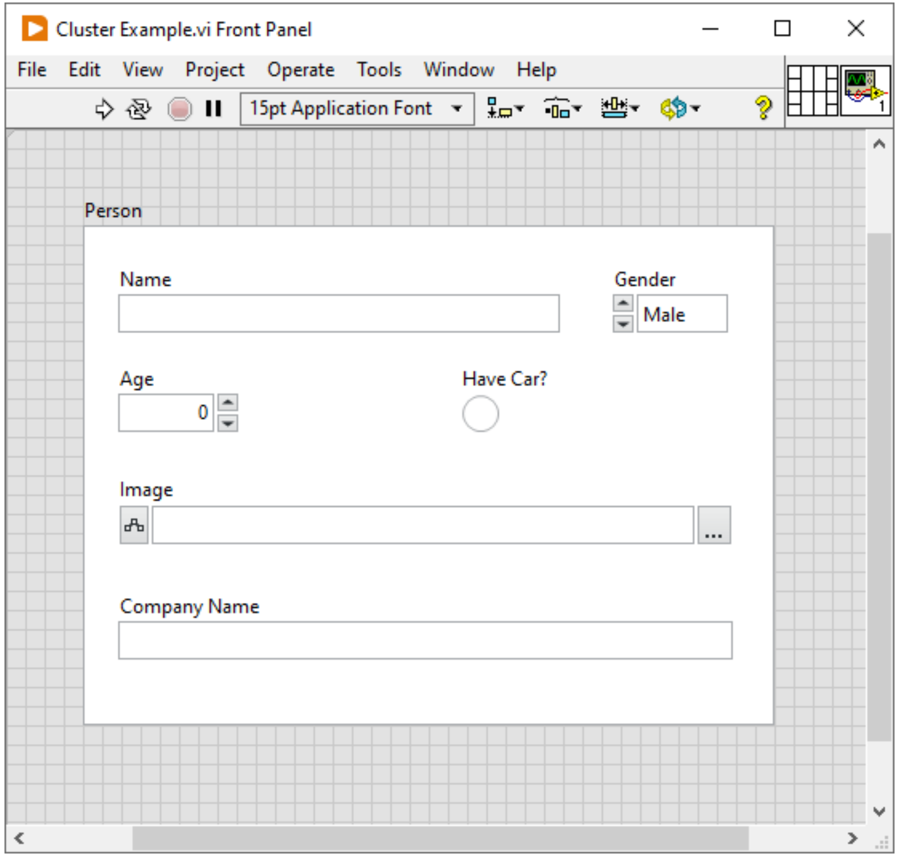

Untitled 1 Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Controls
Search
Fuse Design System
Numeric
Boolean
String & Path
Data Containers
Ring & Enum
Decorations
Silver
Data Containers
Array (Fuse Design System)
Array - Numeric (Fuse Design ...
Array - String (Fuse Design ...
Cluster (Fuse Design System)
Error In (Fuse Design System)
Error Out (Fuse Design System)



Cluster Functions



Cluster Example



Cluster Example.vi Front Panel

File Edit View Project Operate Tools Window Help

15pt Application Font

Person

Name

Gender: Male

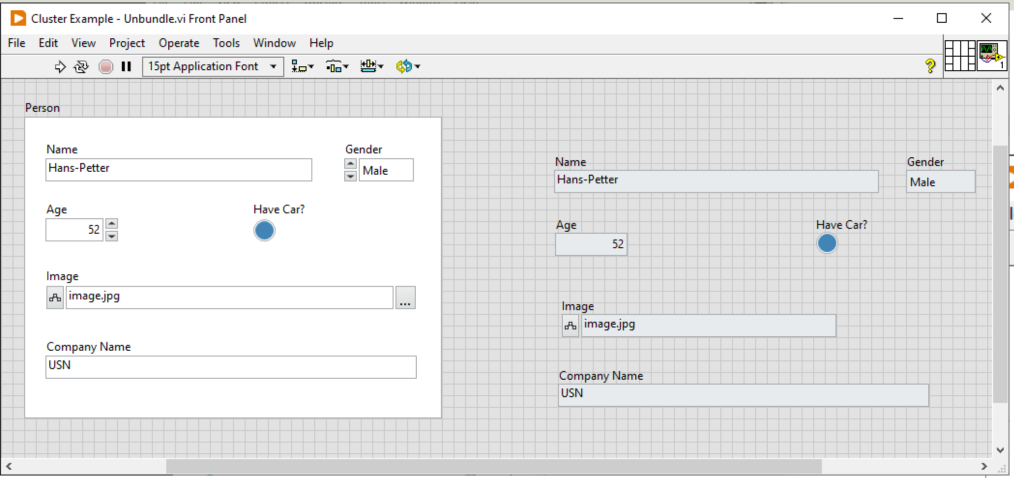
Age: 0

Have Car?

Image

Company Name

Unbundle



Cluster Example - Unbundle.vi Front Panel

File Edit View Project Operate Tools Window Help

15pt Application Font

Person

Name: Hans-Petter

Gender: Male

Age: 52

Have Car?

Image: image.jpg

Company Name: USN

Name: Hans-Petter

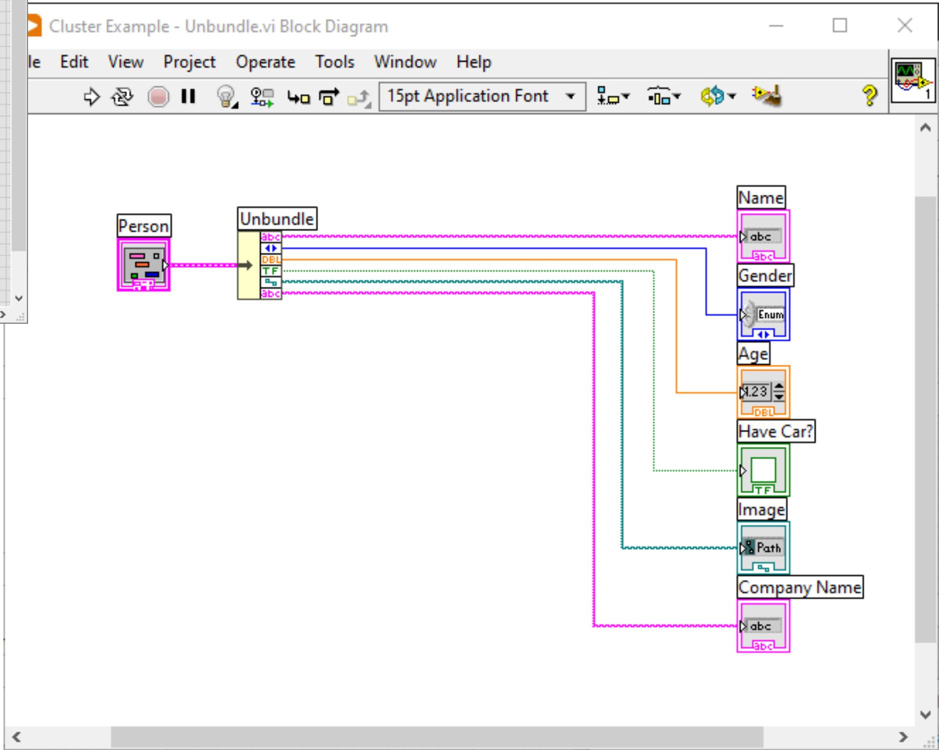
Gender: Male

Age: 52

Have Car?

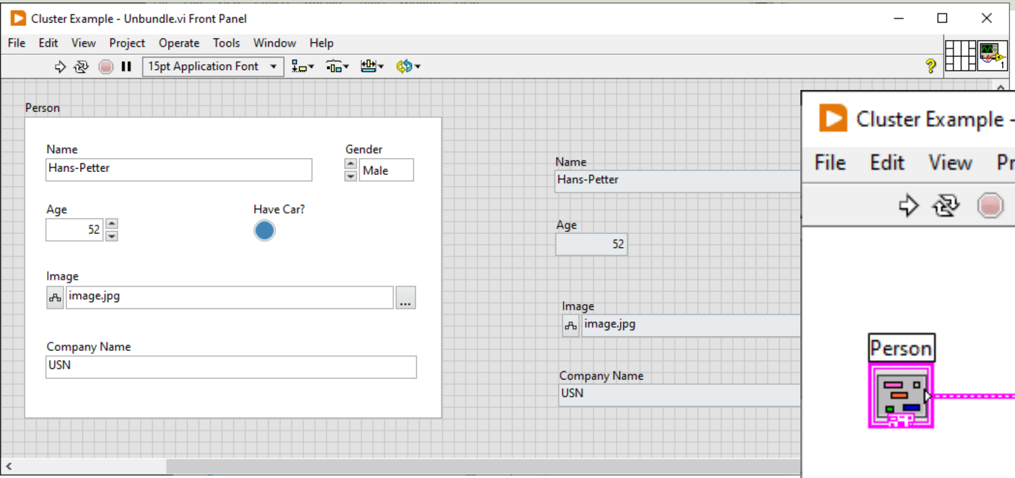
Image: image.jpg

Company Name: USN



Cluster Example - Unbundle.vi Block Diagram

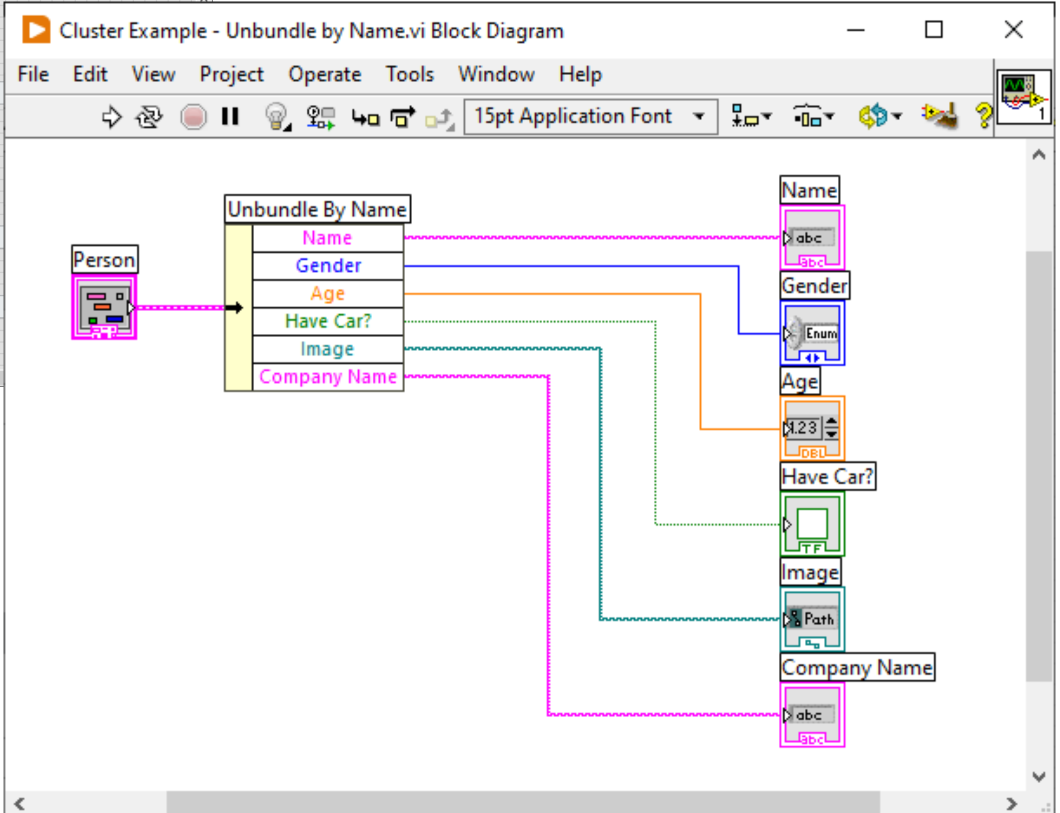
Unbundle by Name


Cluster Example - Unbundle.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font

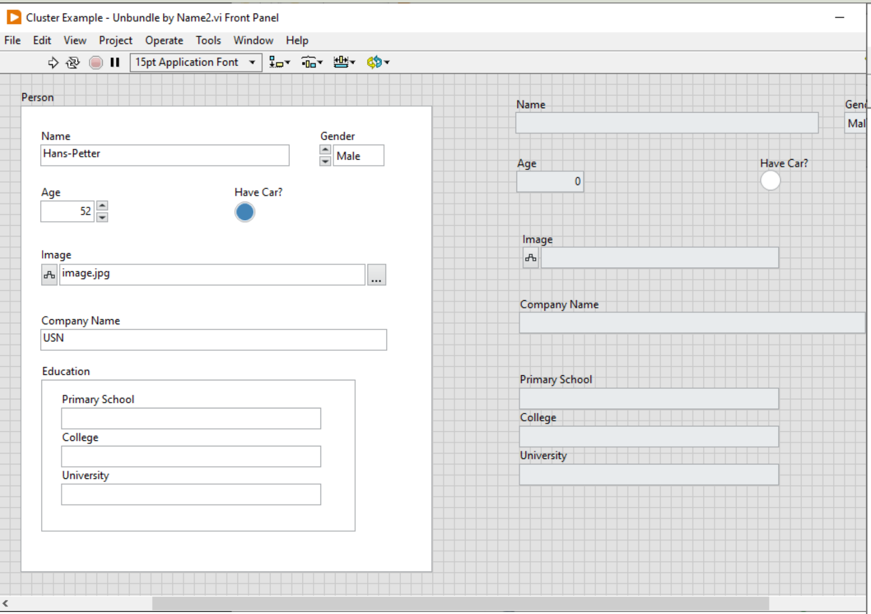
Person

Name: Hans-Petter
Gender: Male
Age: 52
Have Car?
Image: image.jpg
Company Name: USN

Name: Hans-Petter
Age: 52
Image: image.jpg
Company Name: USN



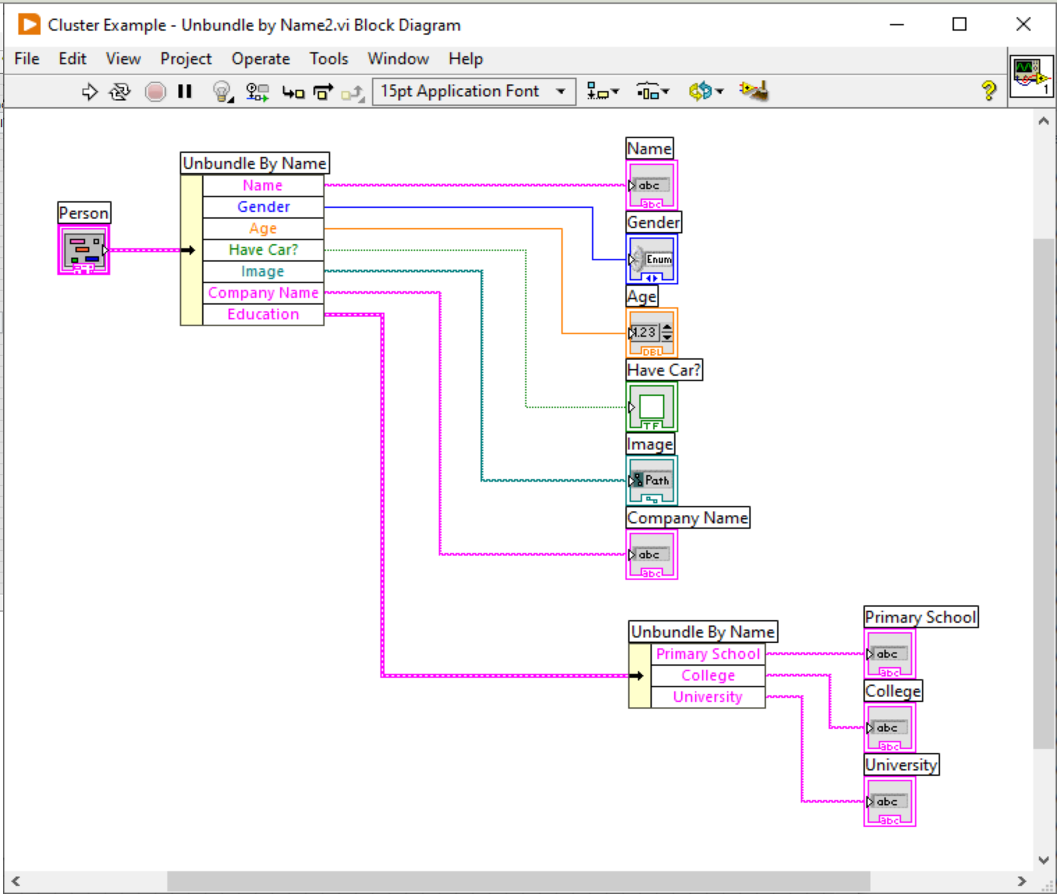
Cluster within Clusters


Cluster Example - Unbundle by Name2.vi Front Panel

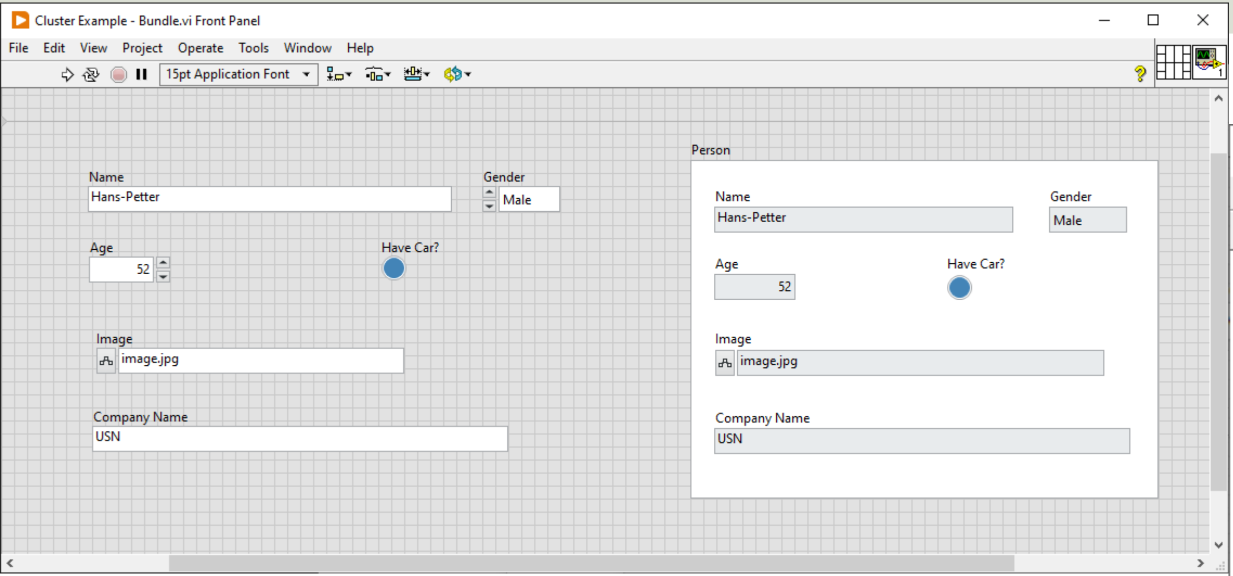
Person

Name: Hans-Petter
Gender: Male
Age: 52
Have Car?
Image: image.jpg
Company Name: USN
Education
Primary School
College
University

Name
Gen
Mal
Age: 0
Have Car?
Image
Company Name
Primary School
College
University


Cluster Example - Unbundle by Name2.vi Block Diagram

Bundle

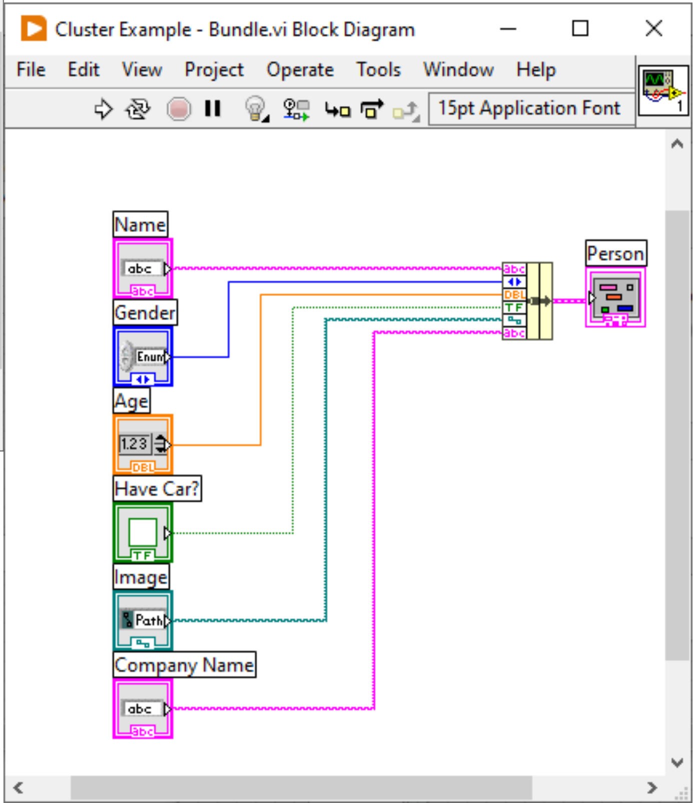

Cluster Example - Bundle.vi Front Panel

File Edit View Project Operate Tools Window Help

15pt Application Font

Name: Hans-Petter
Gender: Male
Age: 52
Have Car?
Image: image.jpg
Company Name: USN

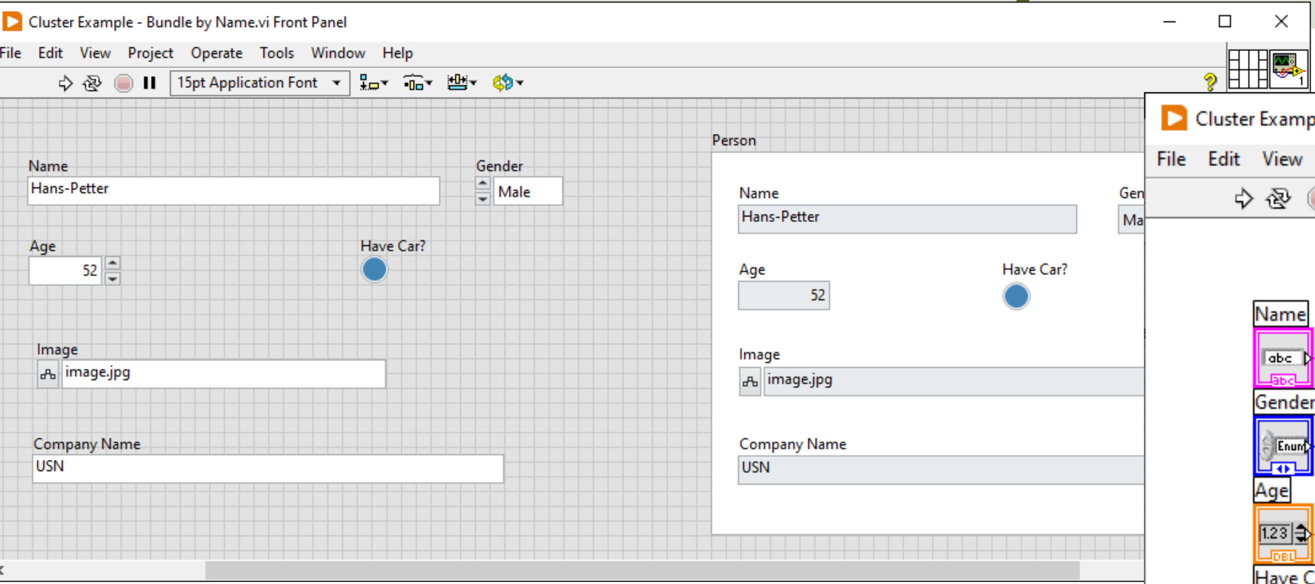
Person
Name: Hans-Petter
Gender: Male
Age: 52
Have Car?
Image: image.jpg
Company Name: USN


Cluster Example - Bundle.vi Block Diagram

File Edit View Project Operate Tools Window Help

15pt Application Font

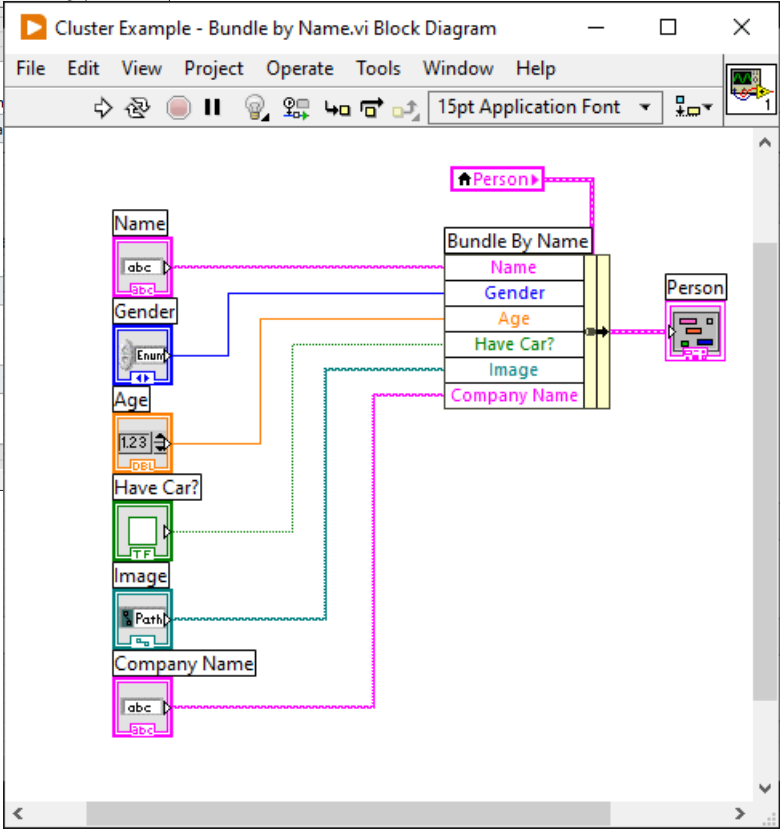
Bundle by Name



Cluster Example - Bundle by Name.vi Front Panel

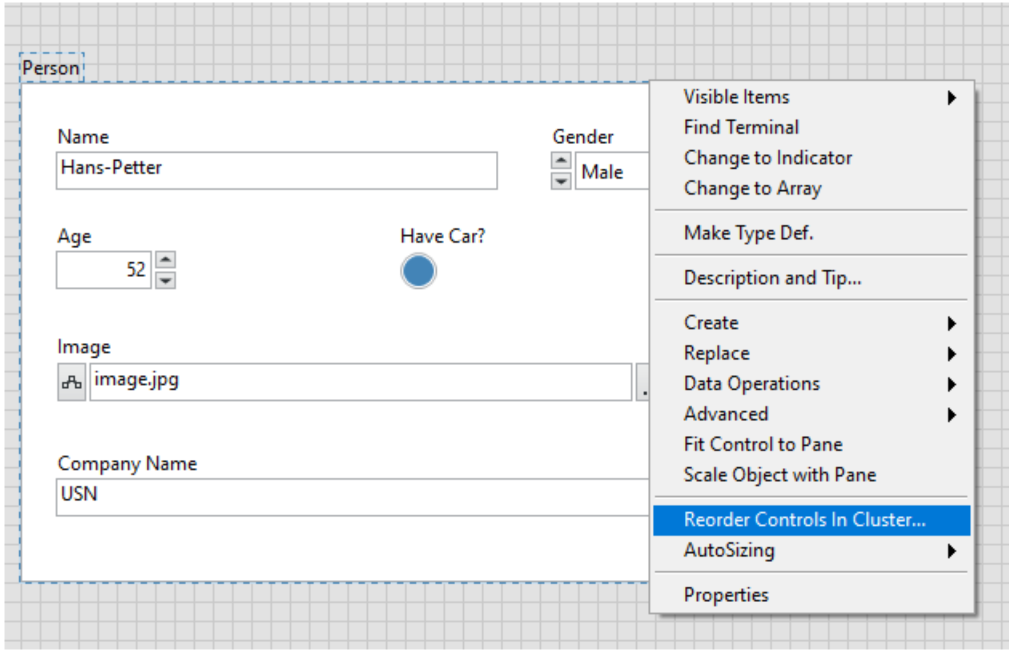
Name: Hans-Petter
Gender: Male
Age: 52
Have Car?
Image: image.jpg
Company Name: USN

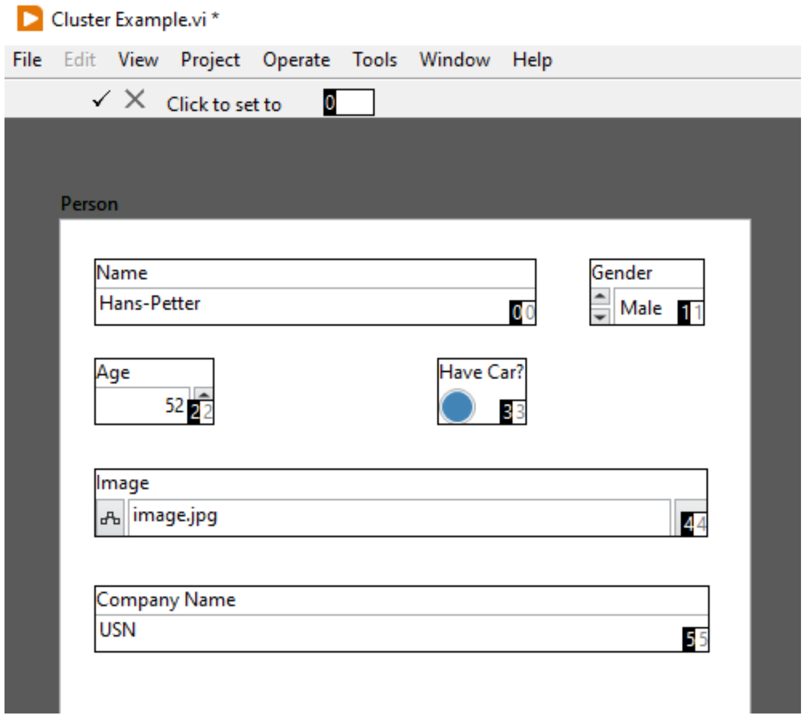
Person
Name: Hans-Petter
Gen: Ma
Age: 52
Have Car?
Image: image.jpg
Company Name: USN



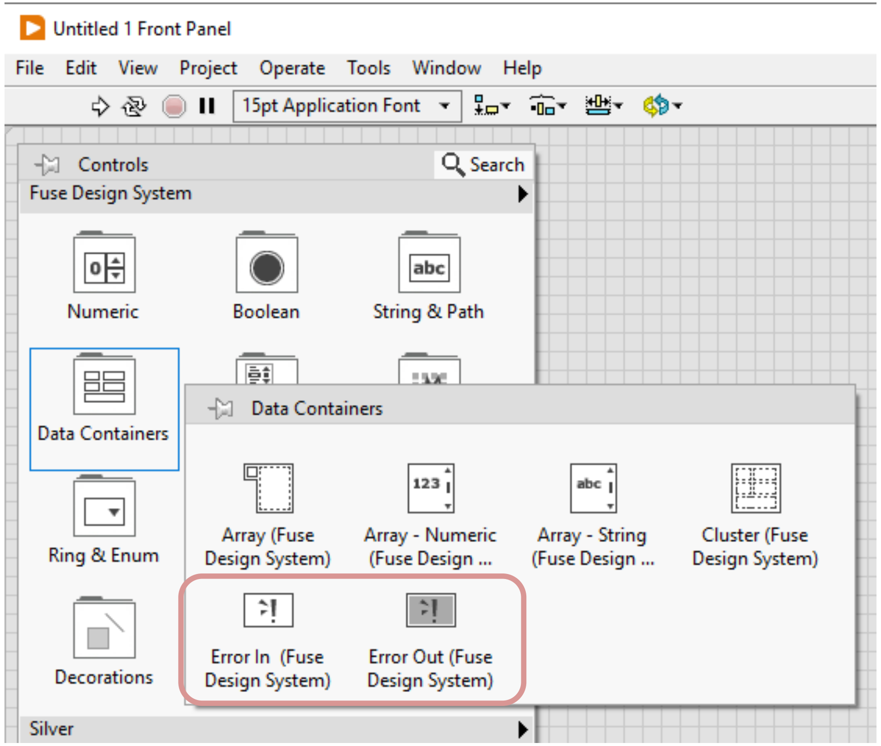
Cluster Example - Bundle by Name.vi Block Diagram

Cluster Order

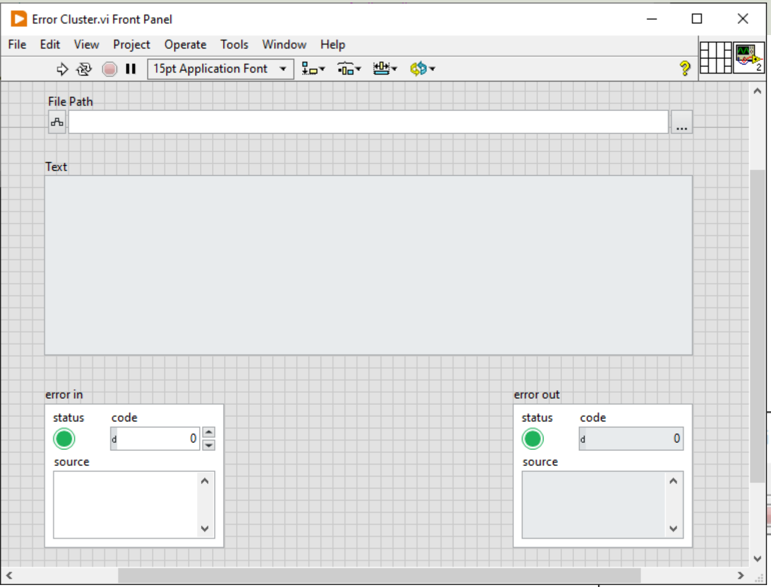


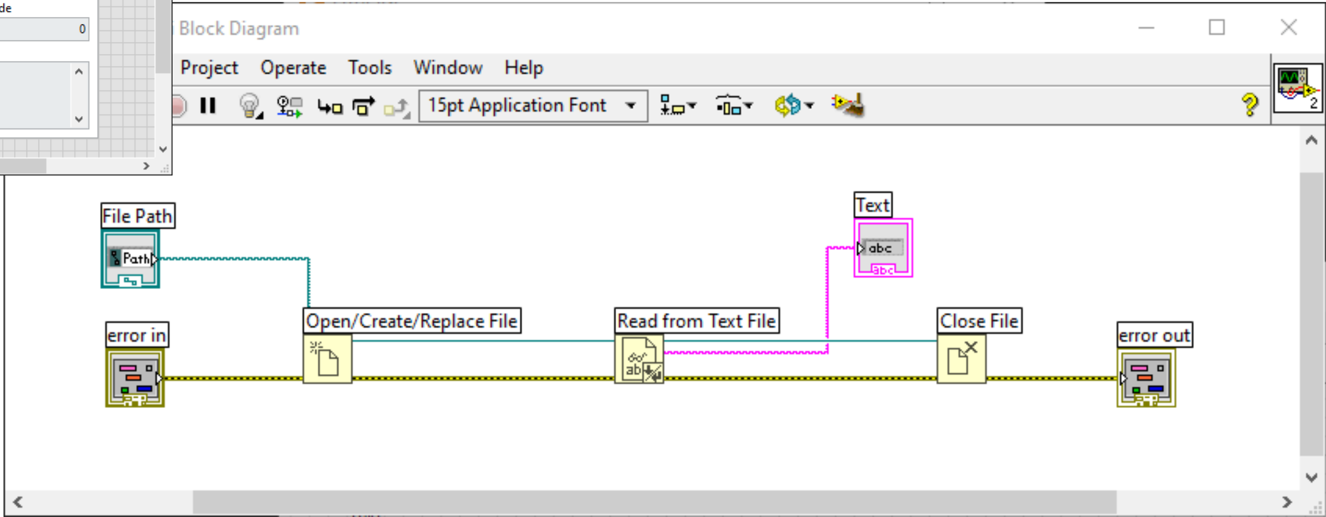


Error Cluster

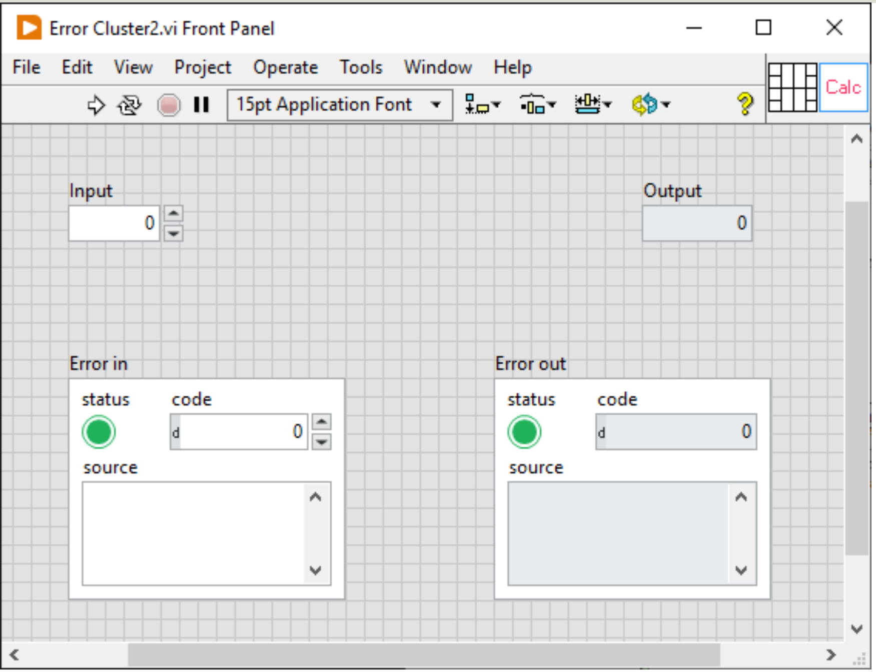

Untitled 1 Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Controls
Search
Fuse Design System
Numeric
Boolean
String & Path
Data Containers
Ring & Enum
Decorations
Silver
Data Containers
Array (Fuse Design System)
Array - Numeric (Fuse Design ...
Array - String (Fuse Design ...
Cluster (Fuse Design System)
Error In (Fuse Design System)
Error Out (Fuse Design System)

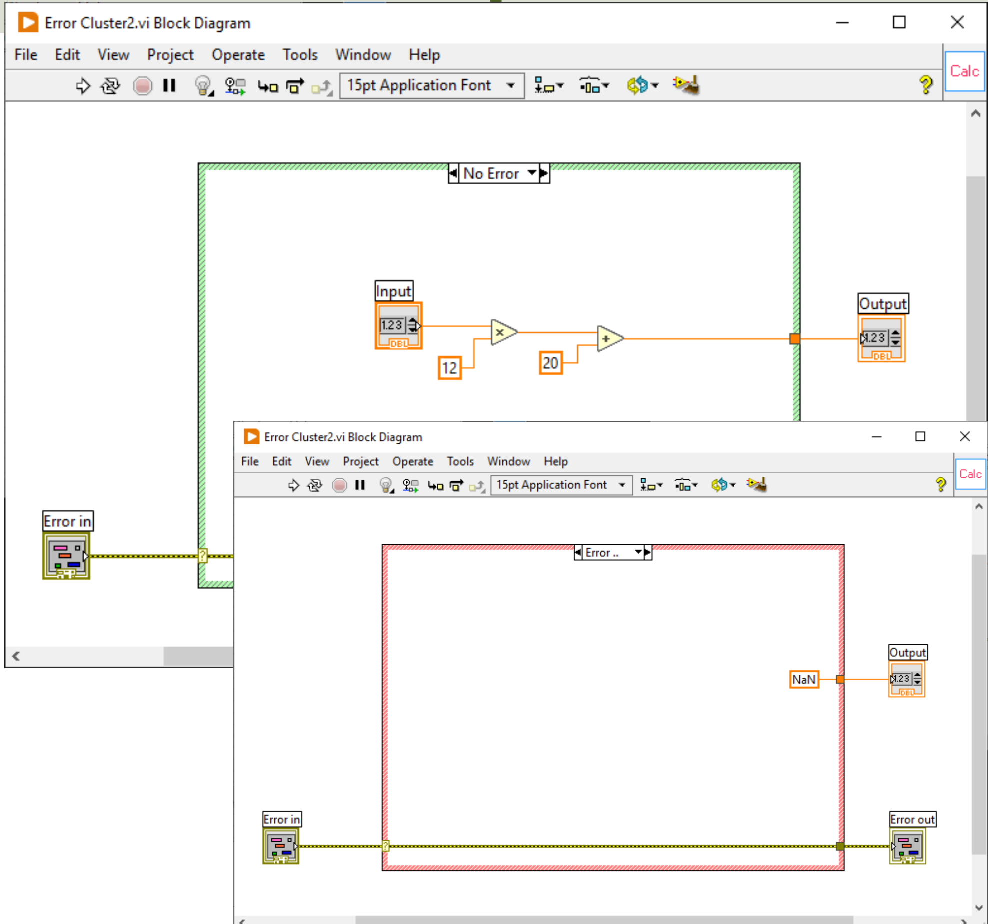
Error Cluster Example



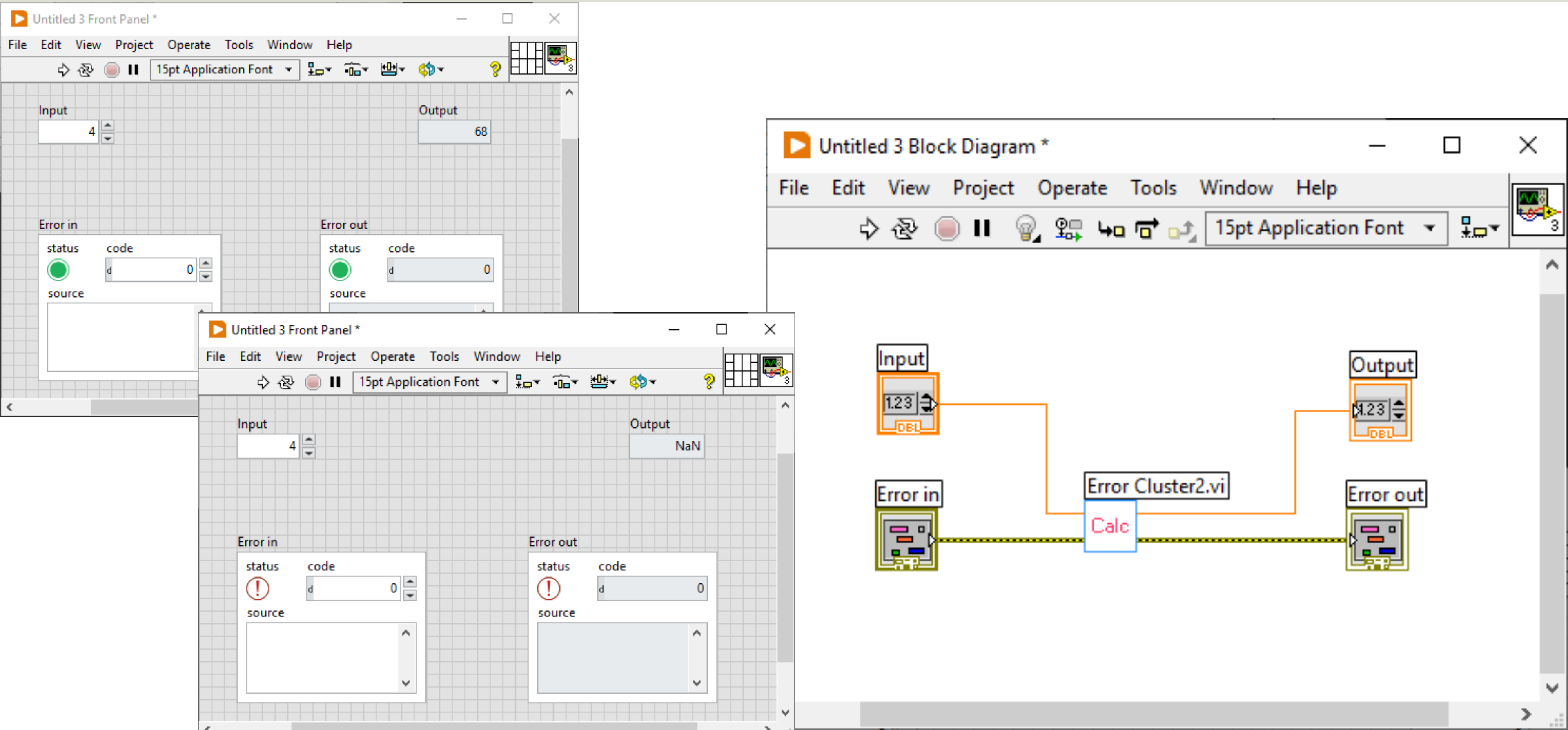


Error Cluster Example




Error Cluster2.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Calc
No Error
Input
1.23
DBL
12
20
Output
Error in
Error Cluster2.vi Block Diagram
Error ..
NaN
Output
Error in
Error out

Error Cluster Example



https://www.halvorsen.blog

LabVIEW Fundamentals

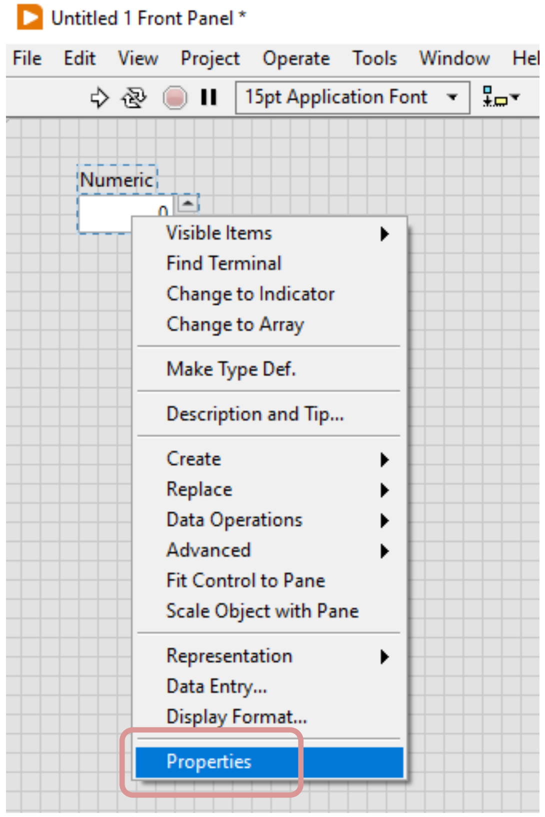
Property Nodes in LabVIEW

Hans-Petter Halvorsen

Property Nodes

- Every Program or Programming IDEs and Programming Languages have Properties
- Typically, you can either set the Properties from a Configuration Window or you can set them from your Code
- Typical Properties can e.g., be Color, Title, Text, Disable, ...
- In, e.g., Visual Studio we have the Properties window where we can set Properties for the selected object/control. These Properties can also be set from the Code
- In LabVIEW we can set Properties in the same way, i.e., we can right-click on different objects – or we can set them from code using so-called Property Nodes

Properties


Untitled 1 Front Panel *

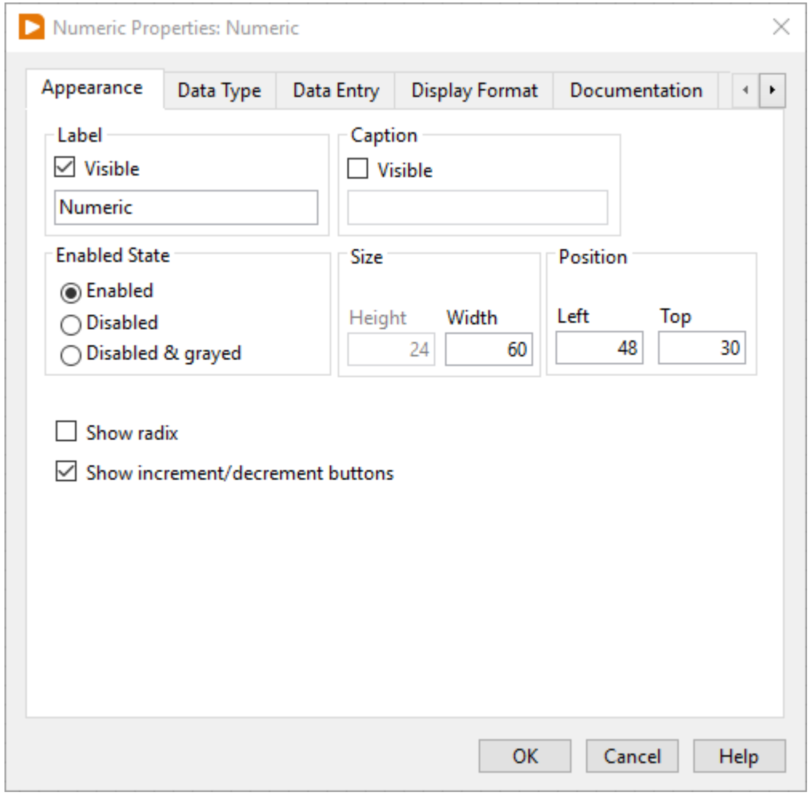
File Edit View Project Operate Tools Window Hel

15pt Application Font

Numeric

- Visible Items
- Find Terminal
- Change to Indicator
- Change to Array
- Make Type Def.
- Description and Tip...
- Create
- Replace
- Data Operations
- Advanced
- Fit Control to Pane
- Scale Object with Pane
- Representation
- Data Entry...
- Display Format...
- Properties

Properties window


Numeric Properties: Numeric

Appearance | Data Type | Data Entry | Display Format | Documentation

Label
- [✓] Visible
- Numeric

Caption
- [ ] Visible

Enabled State
- (●) Enabled
- ( ) Disabled
- ( ) Disabled & grayed

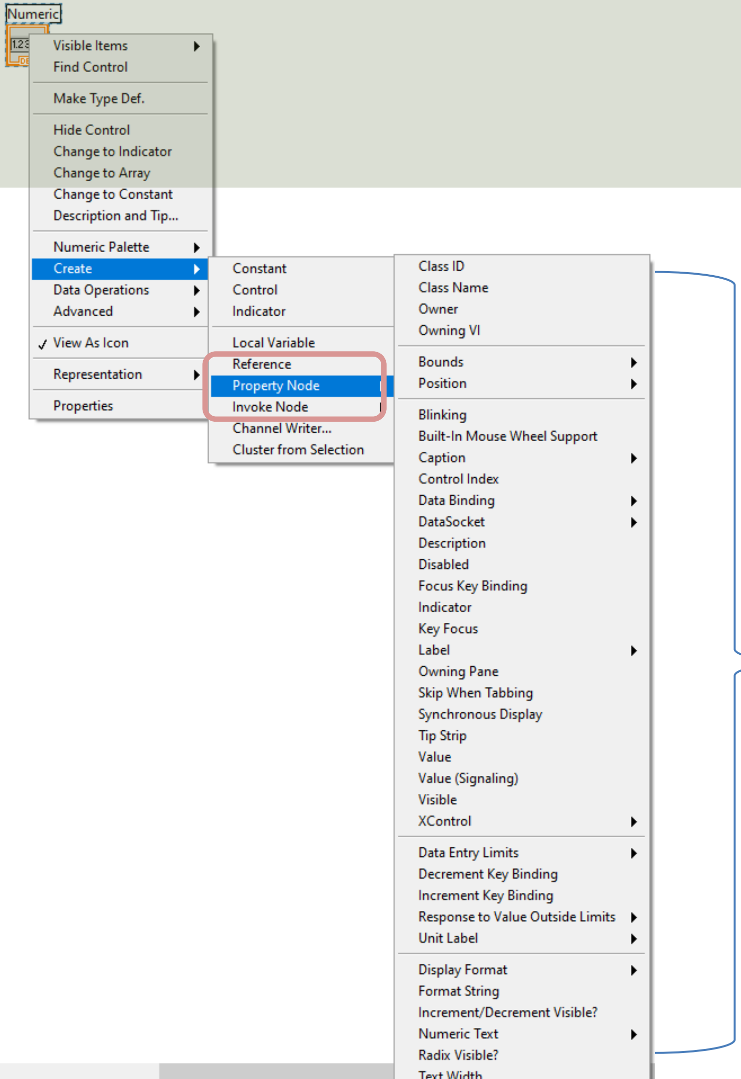
Size
- Height: 24
- Width: 60

Position
- Left: 48
- Top: 30

- [ ] Show radix
- [✓] Show increment/decrement buttons

OK | Cancel | Help

Property Nodes

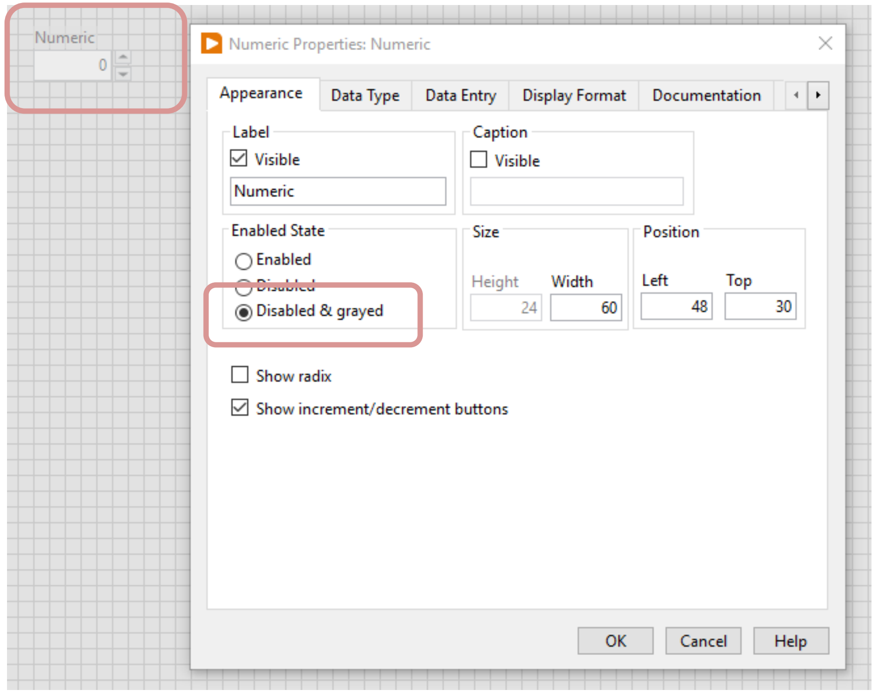


We can set the same Properties as I the Properties window from Code (the LabVIEW Block Diagram) by creating one or more Property Nodes

List of available Properties for selected item

Basic Example

Setting a specific Property from Properties window



Numeric Properties: Numeric

Appearance | Data Type | Data Entry | Display Format | Documentation

Label
[✓] Visible
Numeric

Caption
[ ] Visible

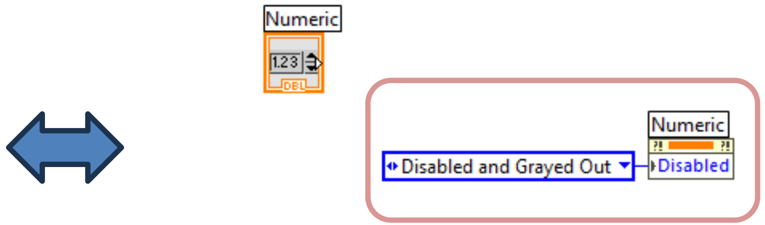
Enabled State
○ Enabled
○ Disabled
● Disabled & grayed

Size
Height 24
Width 60

Position
Left 48
Top 30

[ ] Show radix
[✓] Show increment/decrement buttons

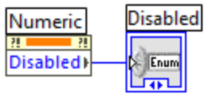
OK | Cancel | Help

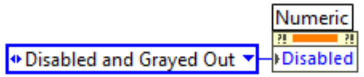


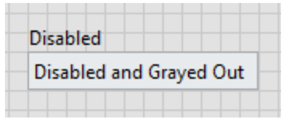
Setting the same Property using a Property Node in Code/Block Diagram

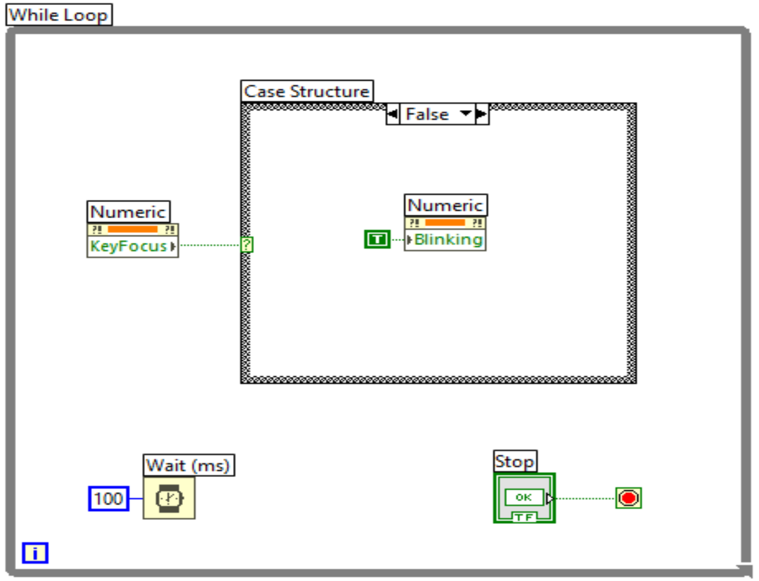
Properties Write/Read

Most of the Properties are both Readable and Writable



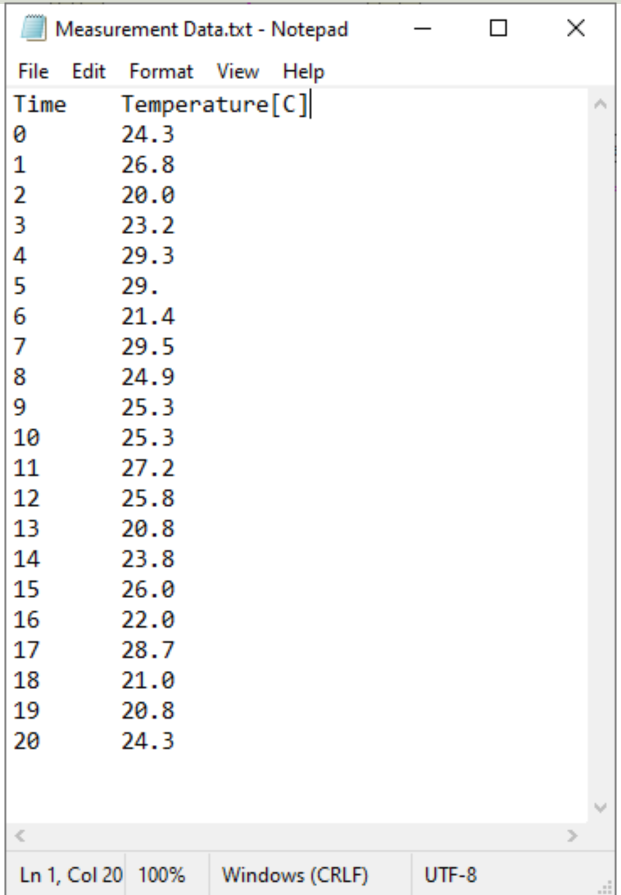






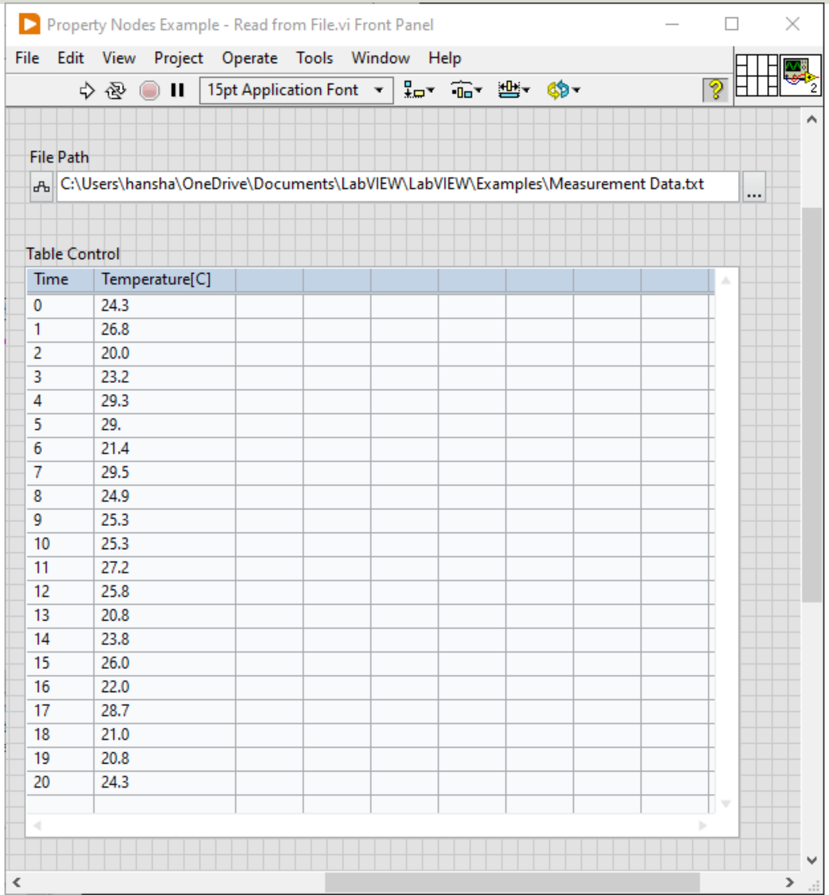
Can be shown on the Front Panel or used in different ways in Code/Block Diagram logic

Property Nodes Example



Measurement Data.txt - Notepad

```
Time	Temperature[C]
0	24.3
1	26.8
2	20.0
3	23.2
4	29.3
5	29.
6	21.4
7	29.5
8	24.9
9	25.3
10	25.3
11	27.2
12	25.8
13	20.8
14	23.8
15	26.0
16	22.0
17	28.7
18	21.0
19	20.8
20	24.3
```



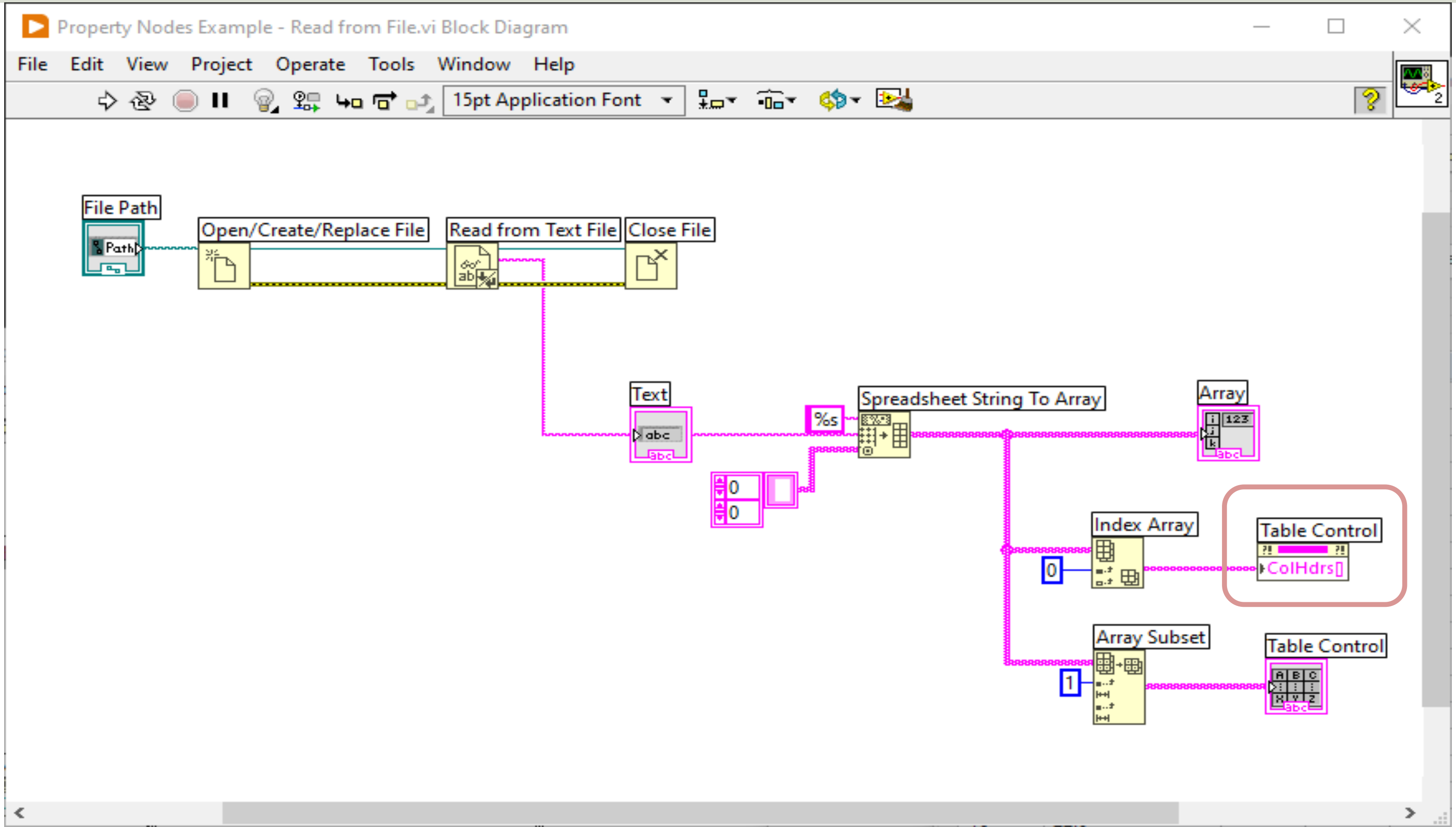
Property Nodes Example - Read from File.vi Front Panel

File Path: C:\Users\hansha\OneDrive\Documents\LabVIEW\LabVIEW\Examples\Measurement Data.txt

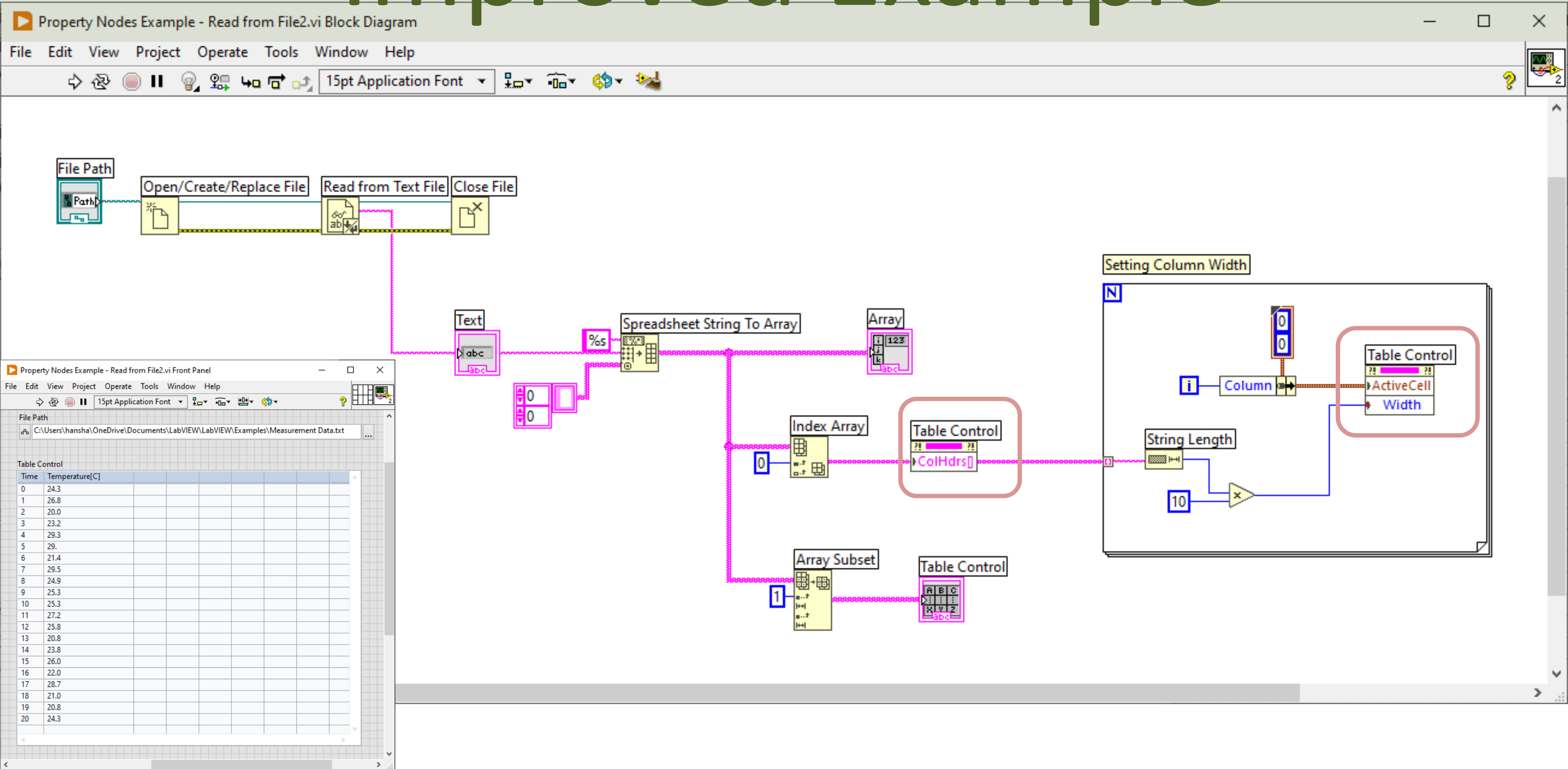
Table Control

Time	Temperature[C]
0	24.3
1	26.8
2	20.0
3	23.2
4	29.3
5	29.
6	21.4
7	29.5
8	24.9
9	25.3
10	25.3
11	27.2
12	25.8
13	20.8
14	23.8
15	26.0
16	22.0
17	28.7
18	21.0
19	20.8
20	24.3

Property Nodes Example



Improved Example



File Path

C:\Users\hansha\OneDrive\Documents\LabVIEW\LabVIEW\Examples\Measurement Data.txt

Table Control

Time	Temperature[C]
0	24.3
1	26.8
2	20.0
3	23.2
4	29.3
5	29.
6	21.4
7	29.5
8	24.9
9	25.3
10	25.3
11	27.2
12	25.8
13	20.8
14	23.8
15	26.0
16	22.0
17	28.7
18	21.0
19	20.8
20	24.3

Chart Properties

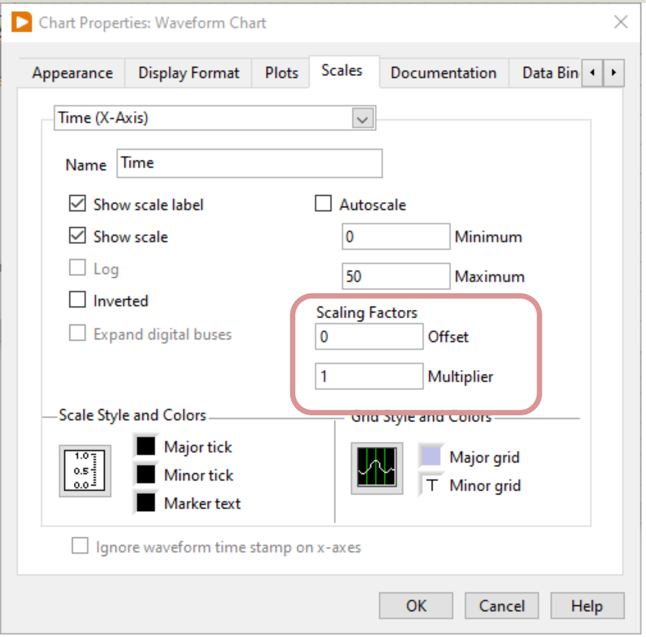


Chart Properties: Waveform Chart

Time (X-Axis)

Name: Time

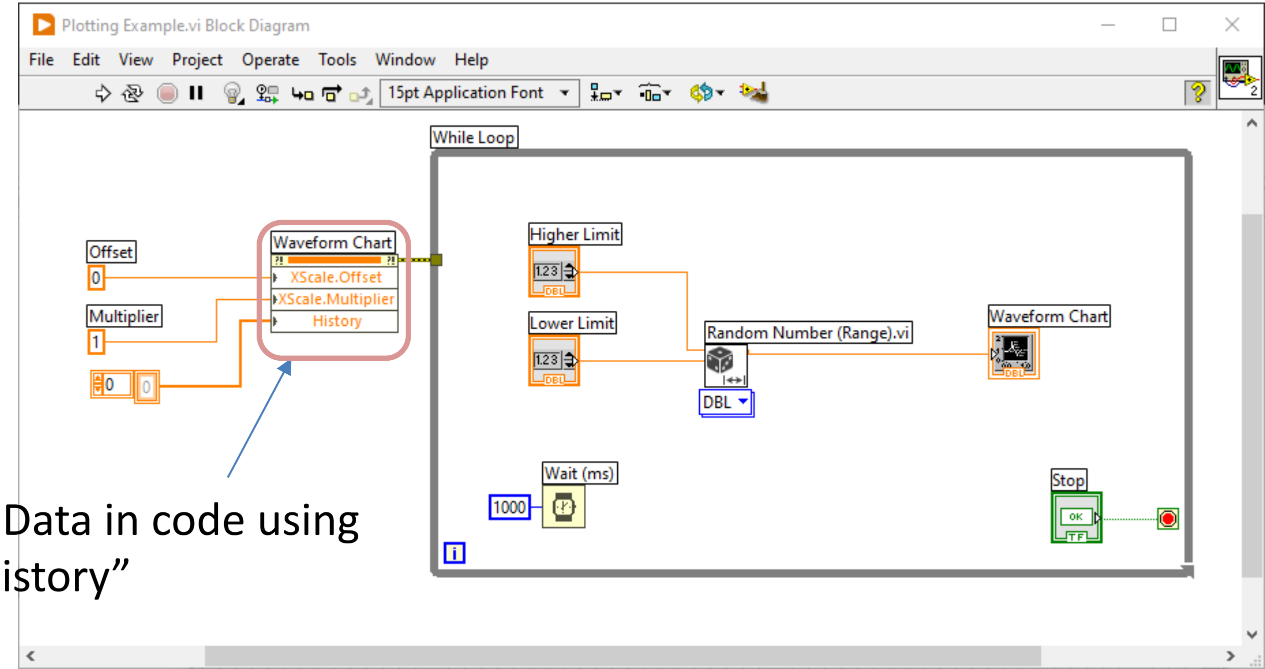
- [✓] Show scale label
- [✓] Show scale
- [ ] Log
- [ ] Inverted
- [ ] Expand digital buses
- [ ] Autoscale

Minimum: 0
Maximum: 50

Scaling Factors
Offset: 0
Multiplier: 1

- [ ] Ignore waveform time stamp on x-axes

You can set these settings either in the GUI or in code using Property Nodes. Offset is typically set equal to zero (starting value on the x-axis), while Multiplier is the interval between to values, e.g., if you plot a new data point every second, you set Multiplier=1, etc.

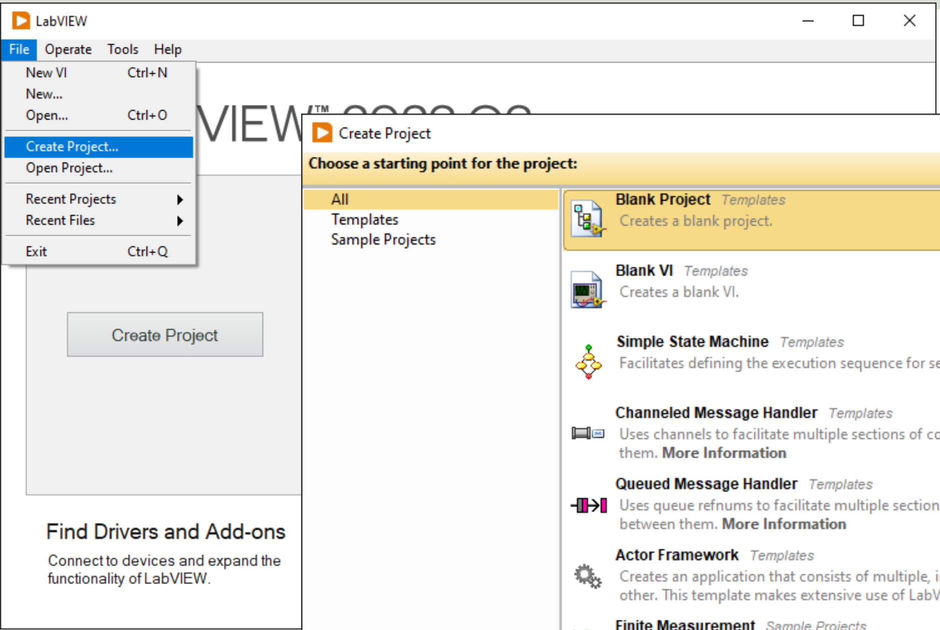


Plotting Example.vi Block Diagram

You can also clear the Chart Data in code using the Property Node called "History"

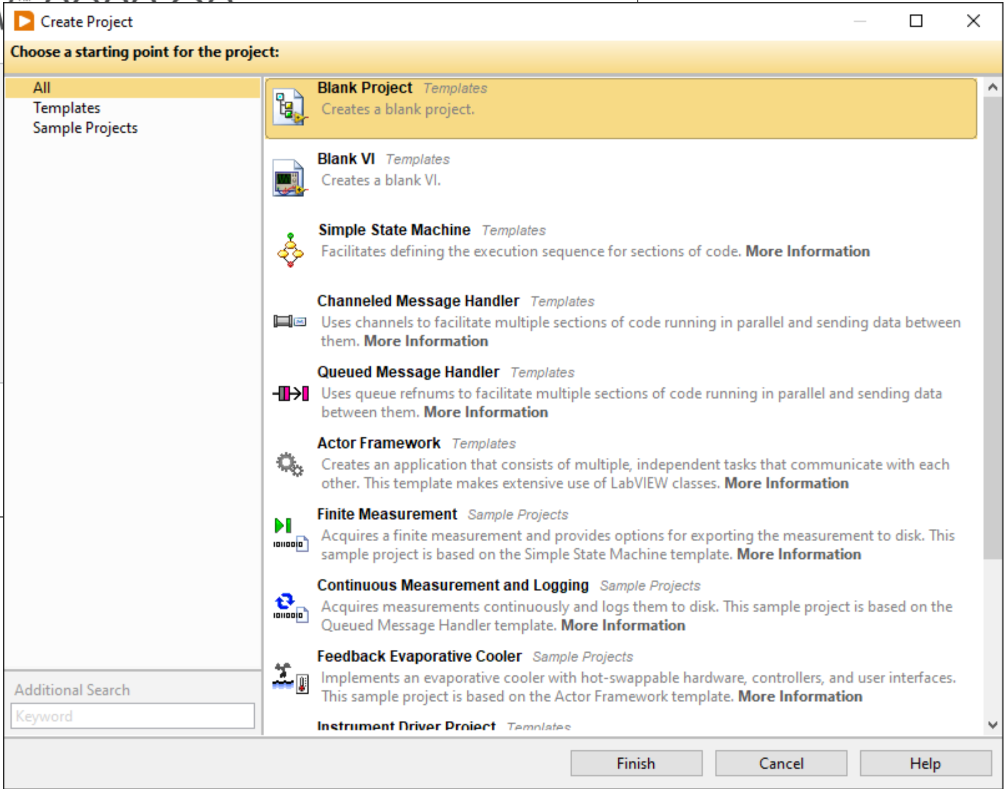
Project Explorer in LabVIEW

Project Explorer



Find Drivers and Add-ons

Connect to devices and expand the functionality of LabVIEW.



Choose a starting point for the project:

Blank Project *Templates*
Creates a blank project.

Blank VI *Templates*
Creates a blank VI.

Simple State Machine *Templates*
Facilitates defining the execution sequence for sections of code. **More Information**

Channeled Message Handler *Templates*
Uses channels to facilitate multiple sections of code running in parallel and sending data between them. **More Information**

Queued Message Handler *Templates*
Uses queue refnums to facilitate multiple sections of code running in parallel and sending data between them. **More Information**

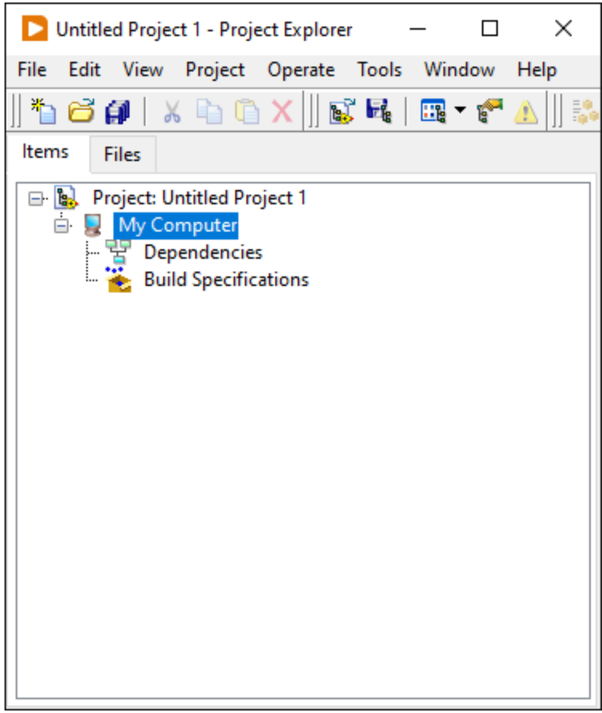
Actor Framework *Templates*
Creates an application that consists of multiple, independent tasks that communicate with each other. This template makes extensive use of LabVIEW classes. **More Information**

Finite Measurement *Sample Projects*
Acquires a finite measurement and provides options for exporting the measurement to disk. This sample project is based on the Simple State Machine template. **More Information**

Continuous Measurement and Logging *Sample Projects*
Acquires measurements continuously and logs them to disk. This sample project is based on the Queued Message Handler template. **More Information**

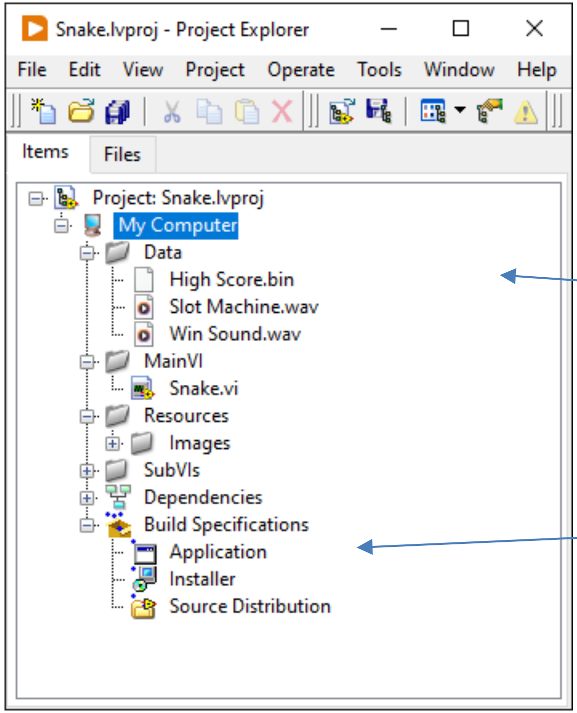
Feedback Evaporative Cooler *Sample Projects*
Implements an evaporative cooler with hot-swappable hardware, controllers, and user interfaces. This sample project is based on the Actor Framework template. **More Information**

Instrument Driver Project *Templates*


Untitled Project 1 - Project Explorer
File Edit View Project Operate Tools Window Help
Items
Files
Project: Untitled Project 1
My Computer
Dependencies
Build Specifications

Project Explorer Example

It is good practice to use the Project Explorer when creating your LabVIEW Applications.
If you have worked with, e.g., Visual Studio, you should already be familiar with the concept (in Visual Studio they call it Solution Explorer).

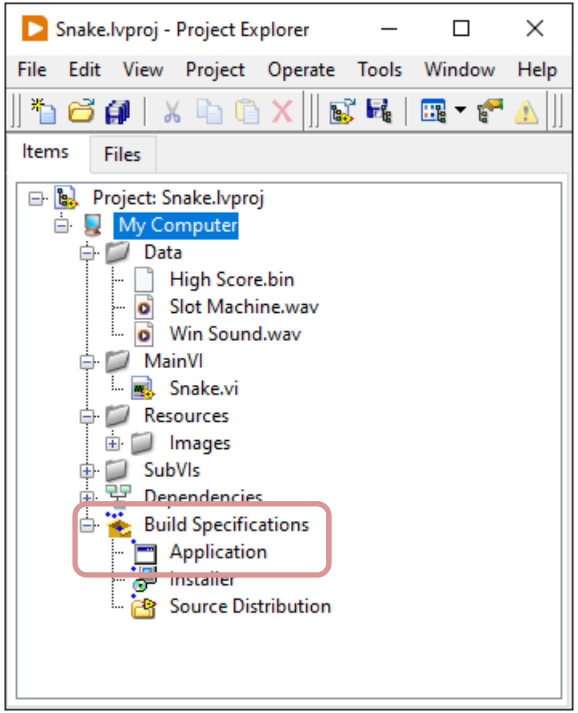

Snake.lvproj - Project Explorer
File Edit View Project Operate Tools Window Help
Items Files

```
Project: Snake.lvproj
  My Computer
    Data
      High Score.bin
      Slot Machine.wav
      Win Sound.wav
    MainVI
      Snake.vi
    Resources
      Images
    SubVIs
    Dependencies
    Build Specifications
      Application
      Installer
      Source Distribution
```

You can structure the different Files in your Application

You need to use the Project Explorer when creating Executable Applications (.exe)

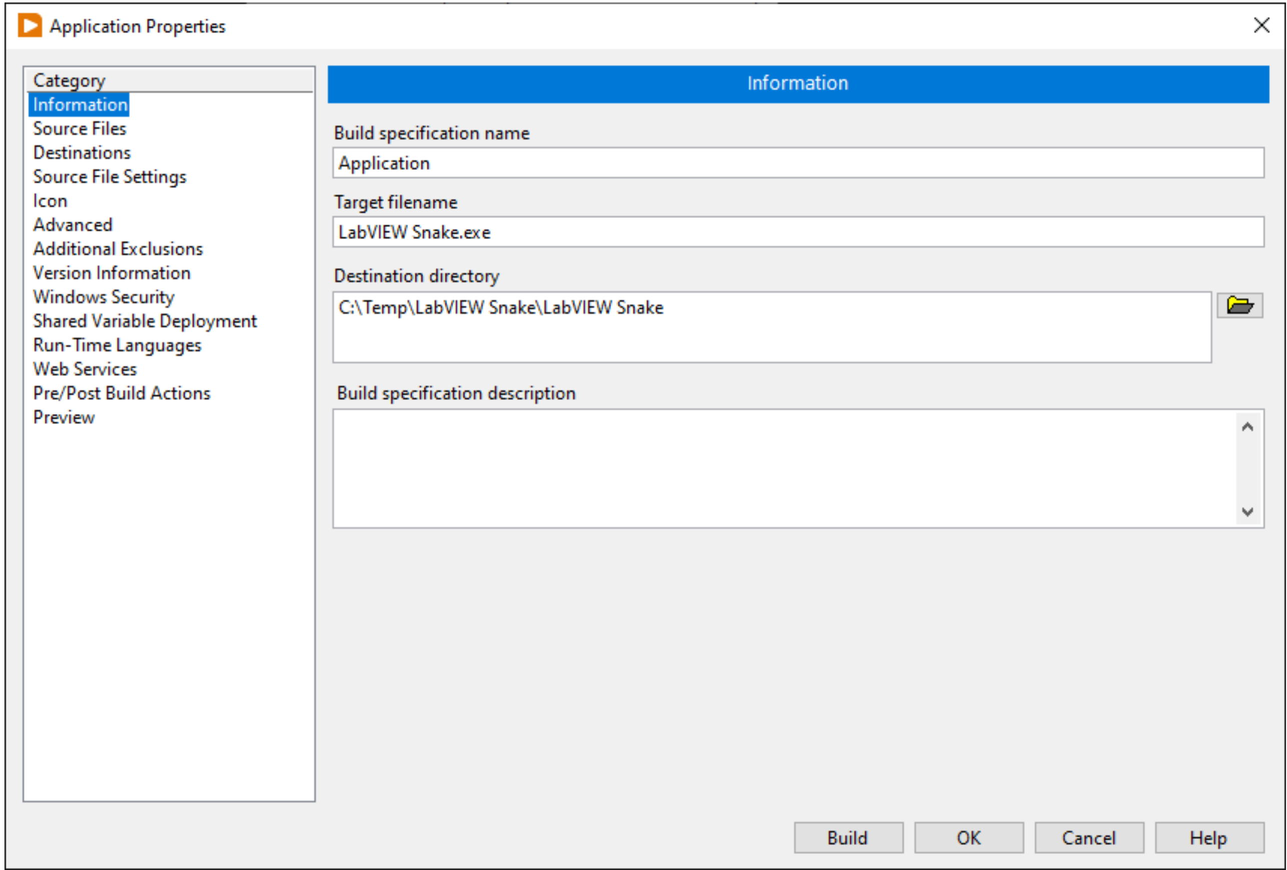
Executable Applications


Snake.lvproj - Project Explorer

File Edit View Project Operate Tools Window Help

Items Files

- Project: Snake.lvproj
  - My Computer
    - Data
      - High Score.bin
      - Slot Machine.wav
      - Win Sound.wav
    - MainVI
      - Snake.vi
    - Resources
      - Images
    - SubVIs
    - Dependencies
    - Build Specifications
      - Application
      - Installer
      - Source Distribution


Application Properties

Category: Information, Source Files, Destinations, Source File Settings, Icon, Advanced, Additional Exclusions, Version Information, Windows Security, Shared Variable Deployment, Run-Time Languages, Web Services, Pre/Post Build Actions, Preview

Information

Build specification name: Application

Target filename: LabVIEW Snake.exe

Destination directory: C:\Temp\LabVIEW Snake\LabVIEW Snake

Build specification description:

Build OK Cancel Help

Debugging Techniques in LabVIEW

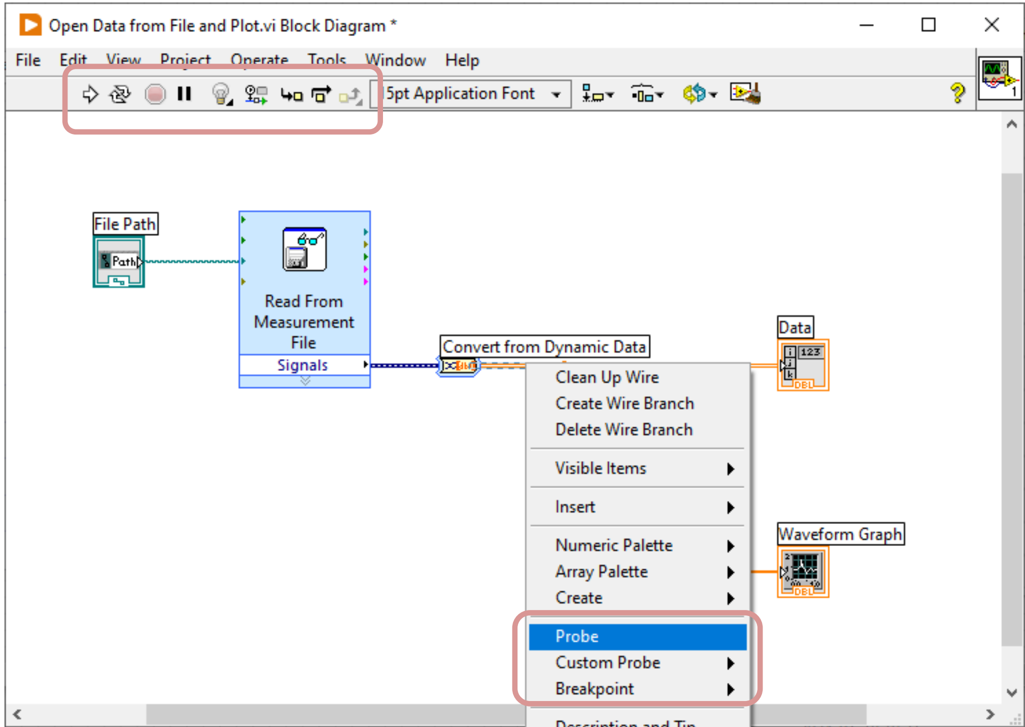
Hans-Petter Halvorsen

Debugging

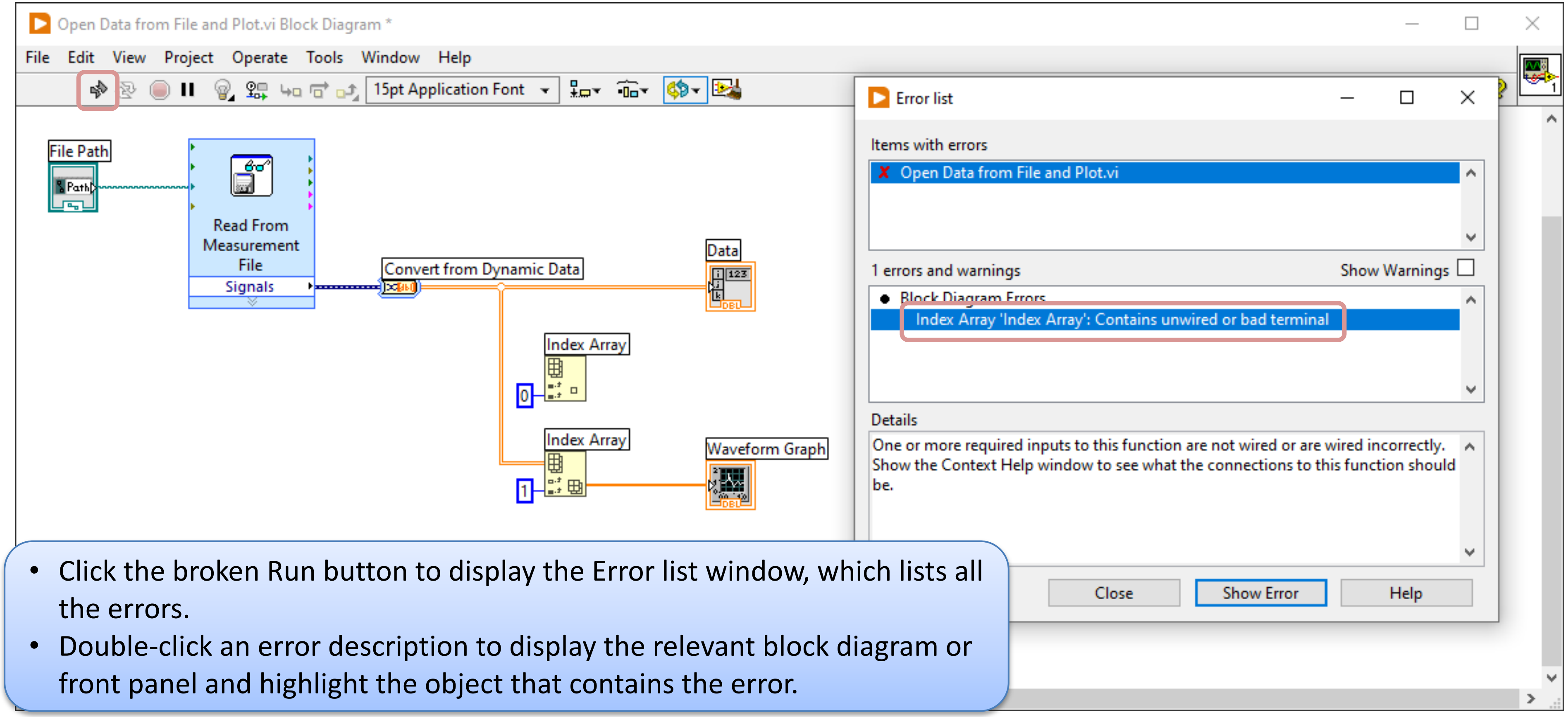
- Debugging is the process of locating and fixing bugs/errors in your computer program
- LabVIEW has powerful debugging techniques for debugging your code
- The sooner you learn to use these debugging techniques the better

Debugging Techniques

- Broken Run Arrow
- Highlight Execution
- Probes
- Pause Execution
- Breakpoints
- ..



Broken Run Arrow



Items with errors

Open Data from File and Plot.vi

1 errors and warnings   Show Warnings

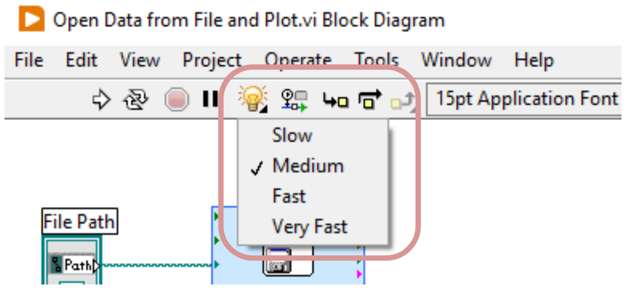
- Block Diagram Errors
  - Index Array 'Index Array': Contains unwired or bad terminal

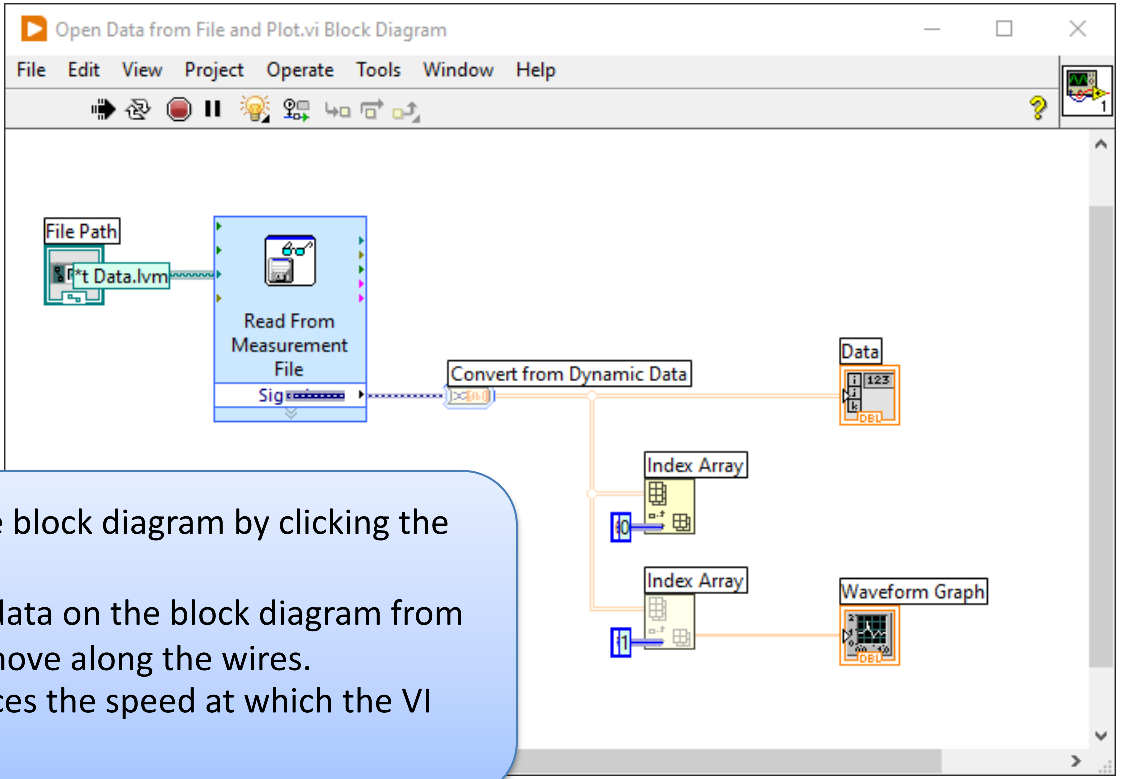
Details

One or more required inputs to this function are not wired or are wired incorrectly. Show the Context Help window to see what the connections to this function should be.

- Click the broken Run button to display the Error list window, which lists all the errors.
- Double-click an error description to display the relevant block diagram or front panel and highlight the object that contains the error.

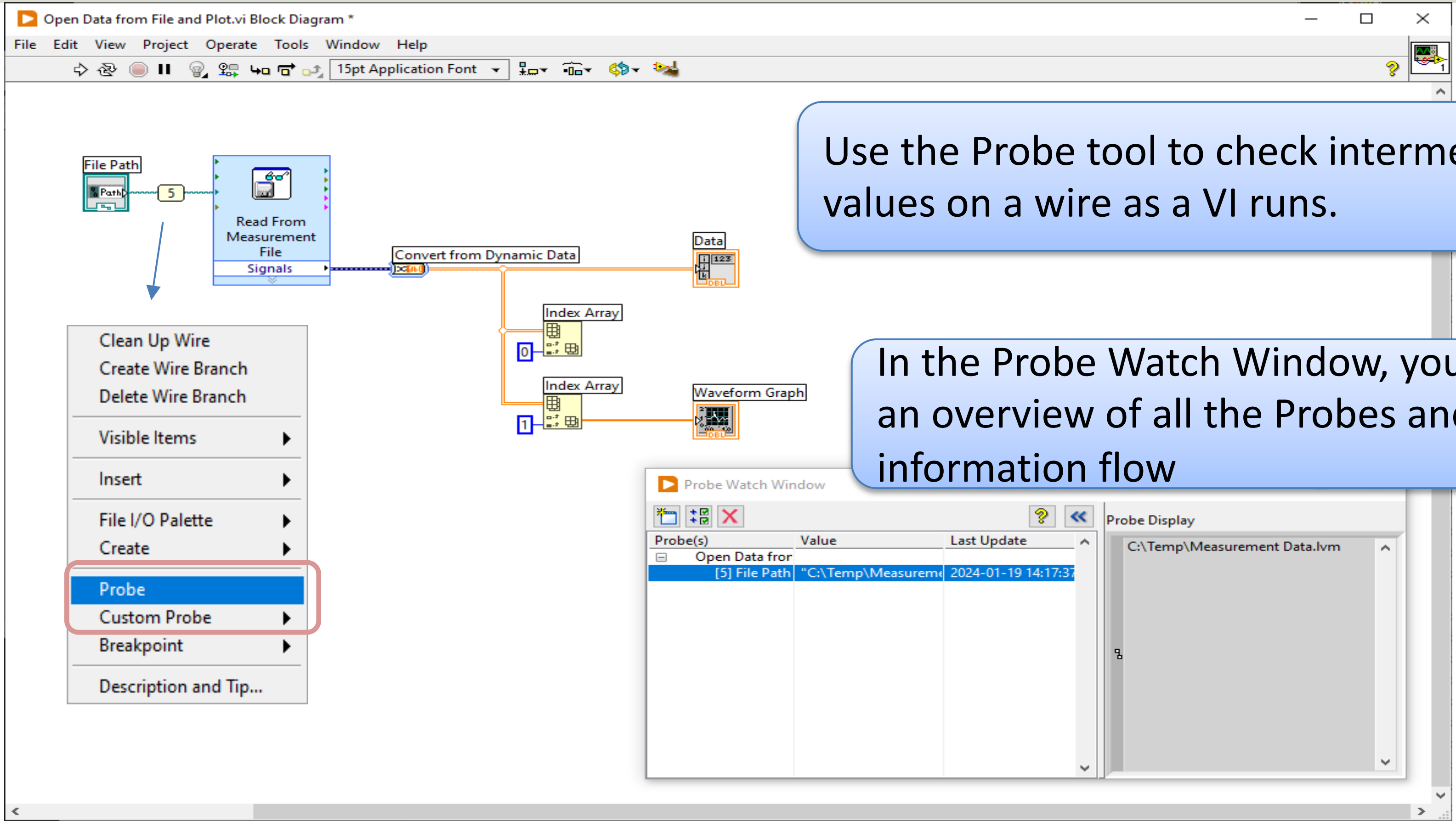
Highlight Execution


Open Data from File and Plot.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Slow
✓ Medium
Fast
Very Fast
File Path
Path


Open Data from File and Plot.vi Block Diagram
File Edit View Project Operate Tools Window Help
File Path
t Data.lvm
Read From Measurement File
Sig
Convert from Dynamic Data
Data
Index Array
0
Index Array
1
Waveform Graph

- View an animation of the execution of the block diagram by clicking the Highlight Execution button.
- Execution highlighting shows the flow of data on the block diagram from one node to another using bubbles that move along the wires.
- Note! Execution highlighting greatly reduces the speed at which the VI runs.

Probes and Probe Watch Window



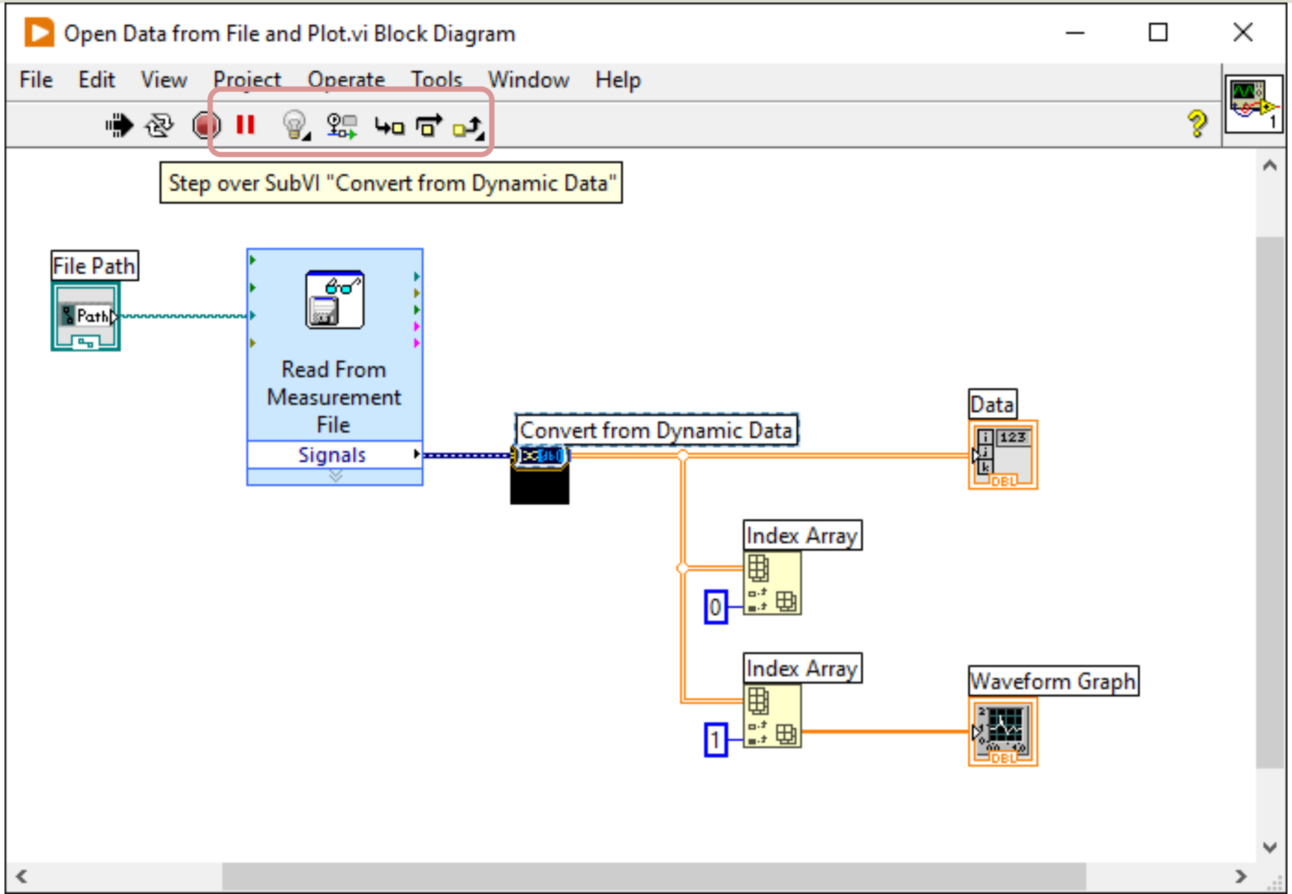
Probe(s)	Value	Last Update
Open Data fror		
[5] File Path	"C:\Temp\Measureme	2024-01-19 14:17:37

C:\Temp\Measurement Data.lvm

Use the Probe tool to check intermediate values on a wire as a VI runs.

In the Probe Watch Window, you get an overview of all the Probes and the information flow

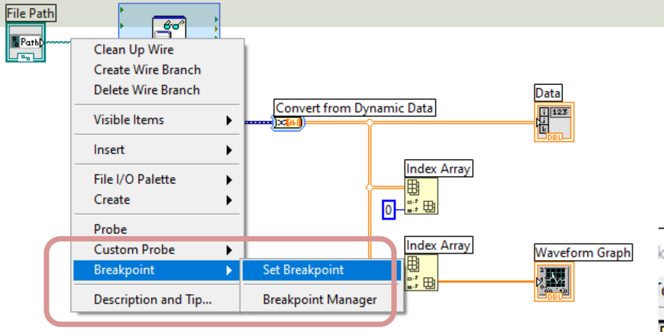
Pause Execution

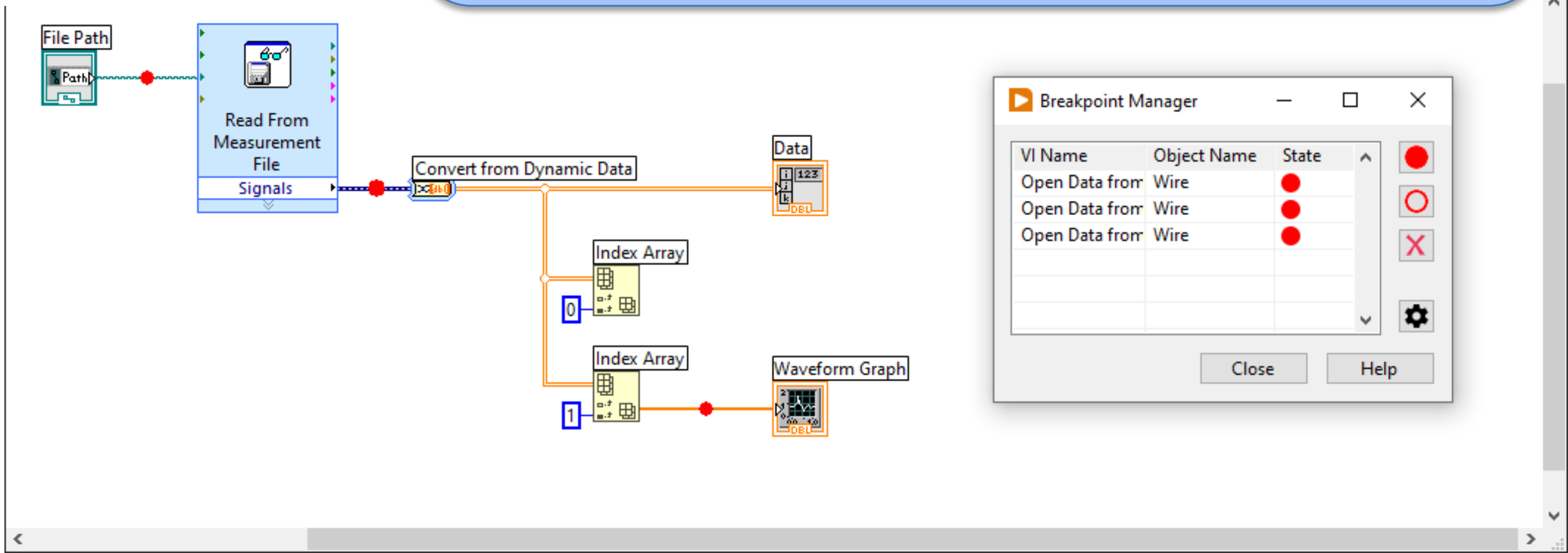


Step over SubVI "Convert from Dynamic Data"

Breakpoints

- Use the Breakpoint tool to place a Breakpoint on a VI, node, or wire on the block diagram and pause execution at that location.
- When you set a breakpoint on a wire, execution pauses after data passes through the wire.





VI Name	Object Name	State
Open Data from	Wire	●
Open Data from	Wire	●
Open Data from	Wire	●

https://www.halvorsen.blog

LabVIEW Fundamentals

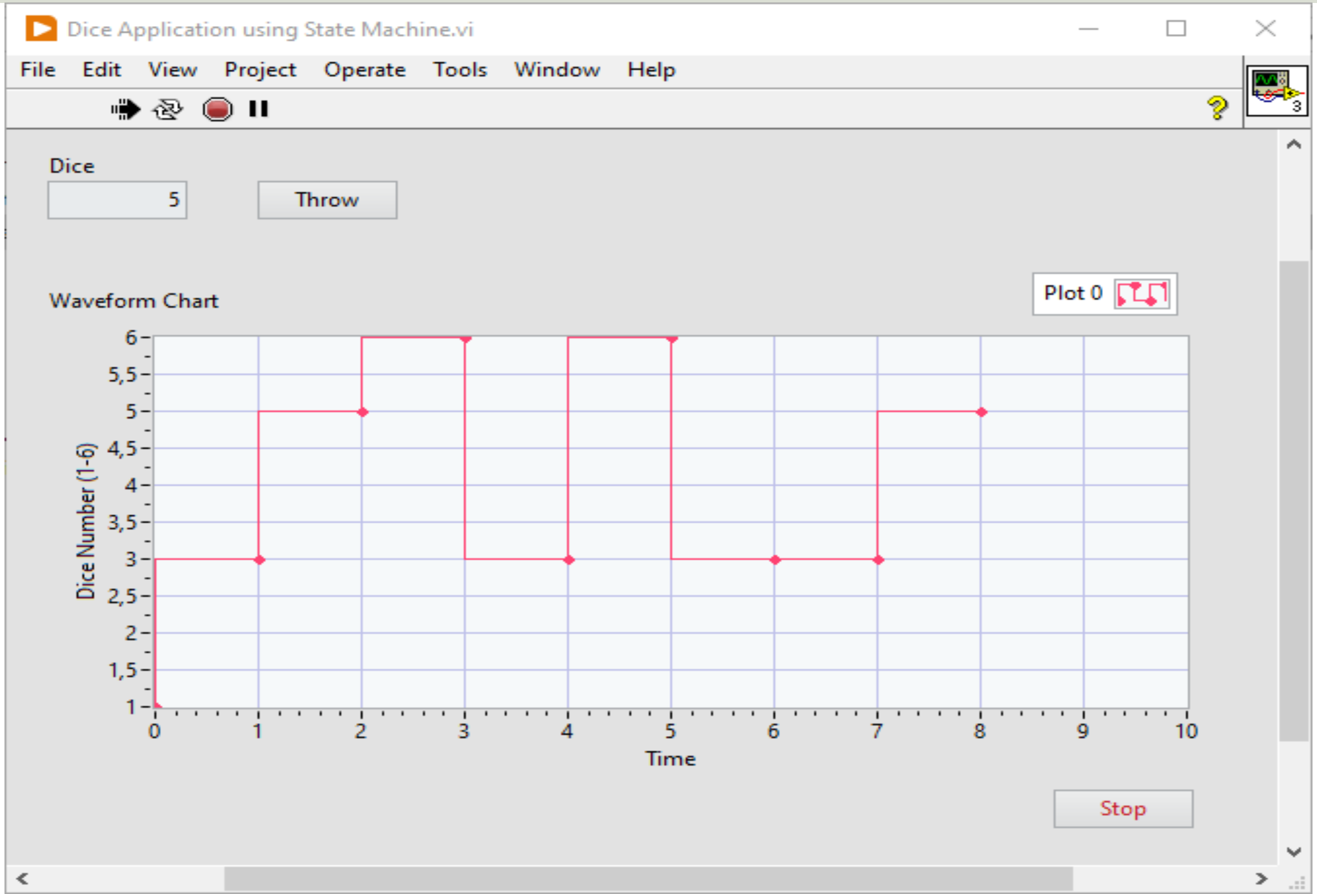
State Machine in LabVIEW

Hans-Petter Halvorsen

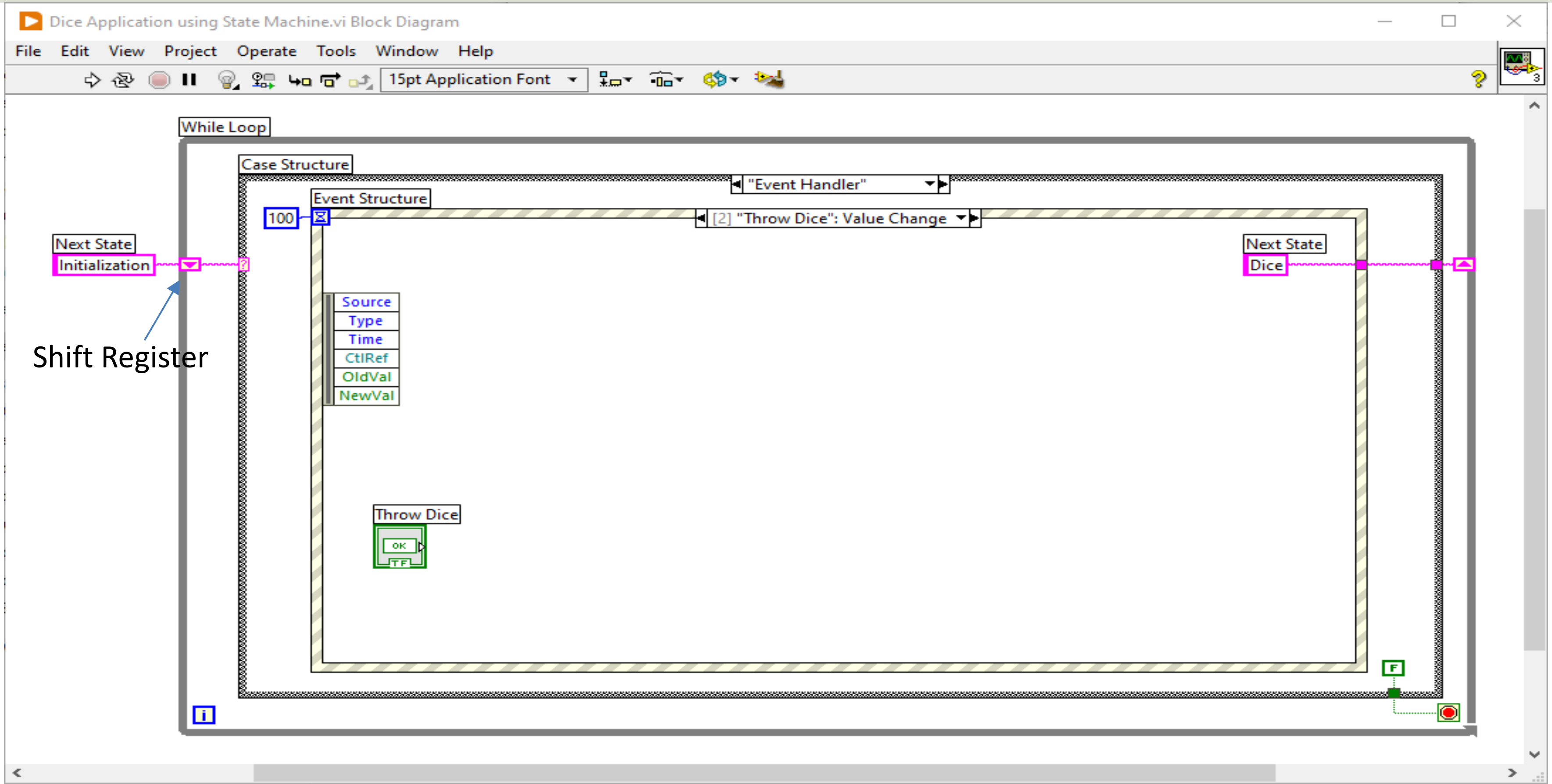
State Machine in LabVIEW

- We will create a basic Application in LabVIEW where we will use the “State Machine” principle
- Using this basic principle, you can easily create larger applications in LabVIEW
- It is also easy to extend the Application with new features

State Machine Example

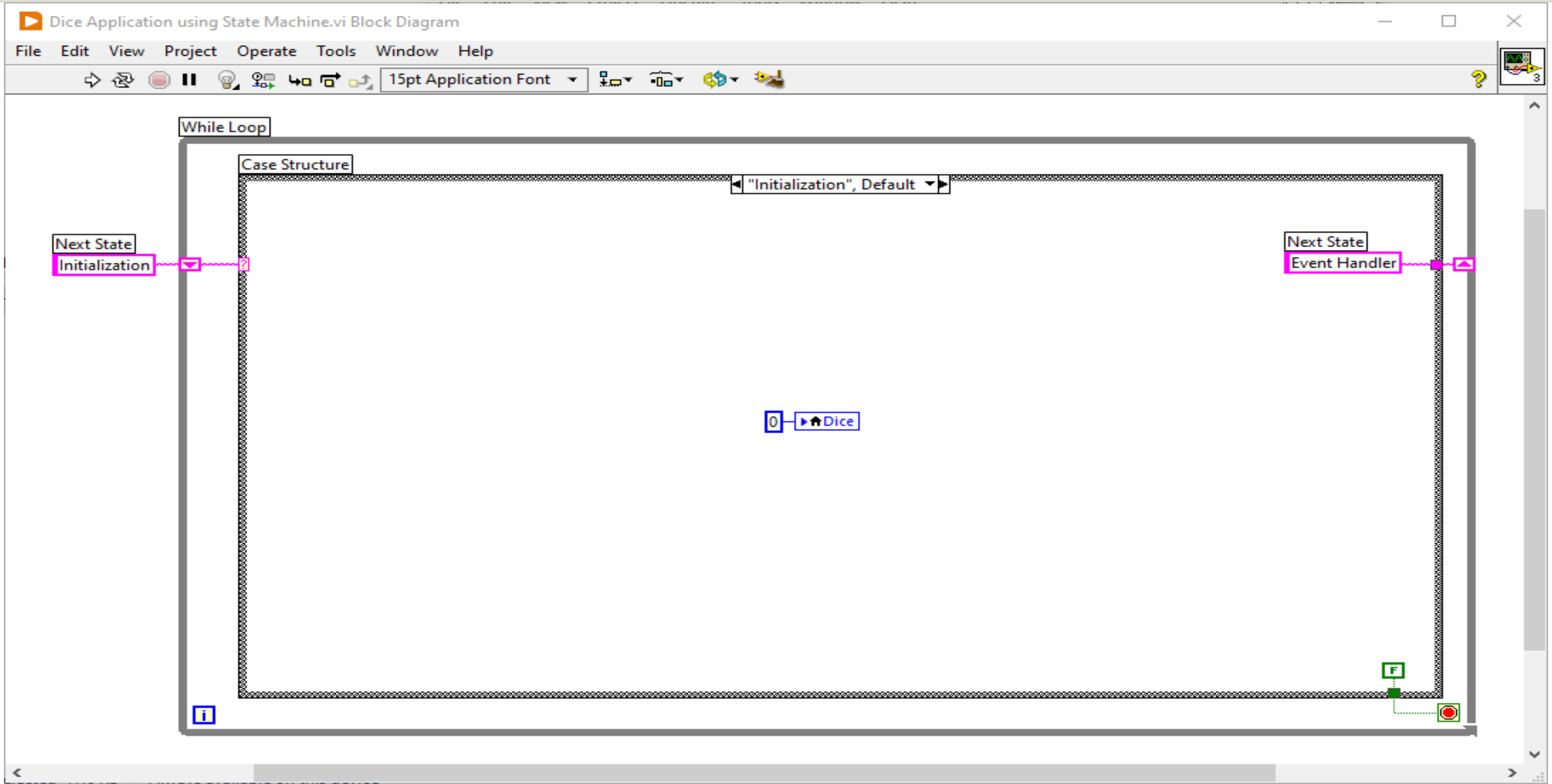


State Machine Example



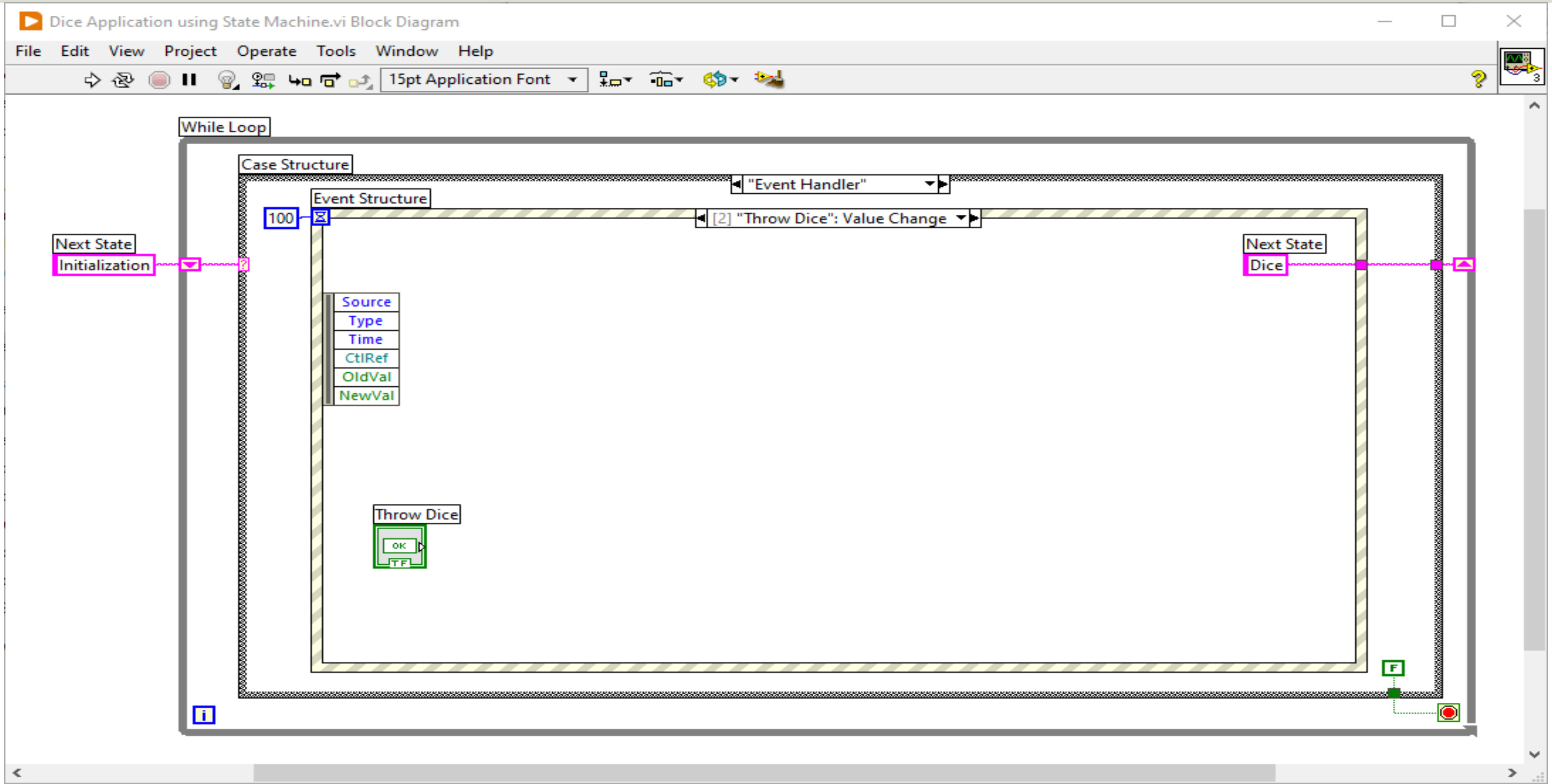
Shift Register

Block Diagram 1

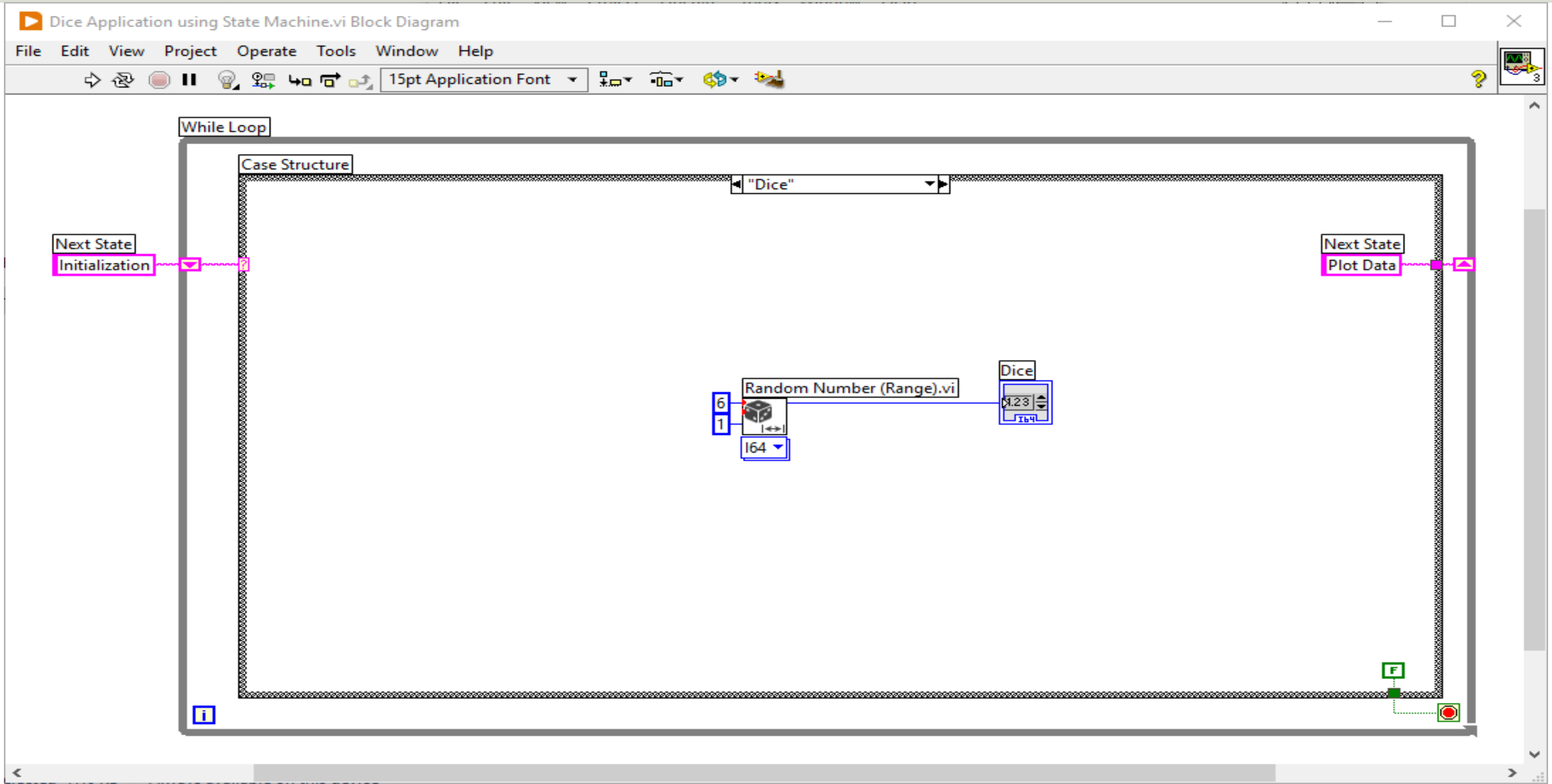


Dice Application using State Machine.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
While Loop
Case Structure
"Initialization", Default
Next State
Initialization
Next State
Event Handler
0
Dice
F
i

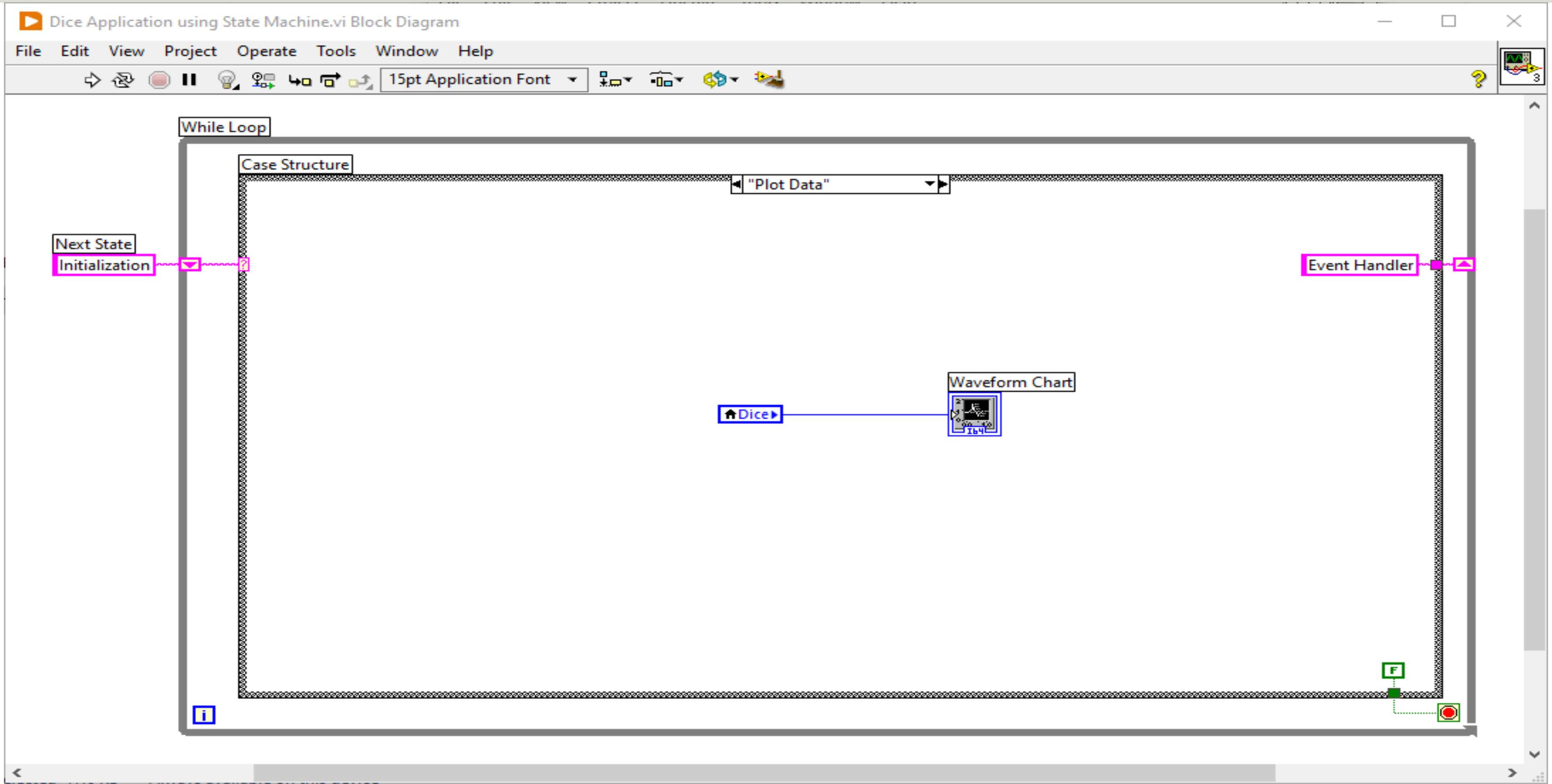
Block Diagram 2



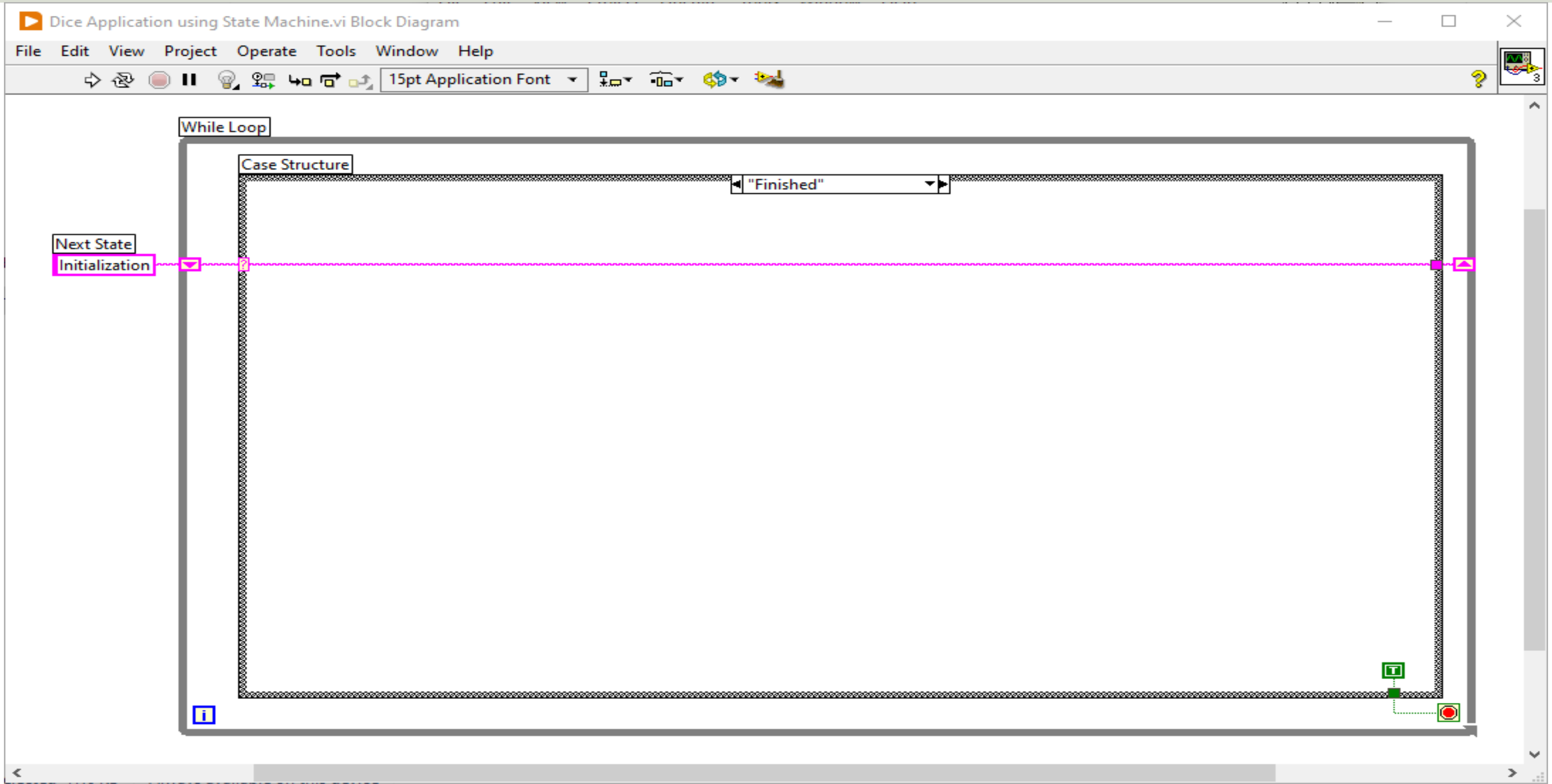
Block Diagram 3



Block Diagram 4



Block Diagram 5


Dice Application using State Machine.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
While Loop
Case Structure
"Finished"
Next State
Initialization

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog