Strings and Property/Invoke Nodes

STRINGS

Strings are simply text that the computer registers as a bunch of ASCII coded numbers.

ASCII control characters

DEC	HEX	Simbolo ASCII	
00	00h	NULL	(carácter nulo)
01	01h	SOH	(inicio encabezado)
02	02h	STX	(inicio texto)
03	03h	ETX	(fin de texto)
04	04h	EOT	(fin transmisión)
05	05h	ENQ	(enquiry)
06	06h	ACK	(acknowledgement)
07	07h	BEL	(timbre)
08	08h	BS	(retroceso)
09	09h	HT	(tab horizontal)
10	0Ah	LF	(salto de linea)
11	0Bh	VT	(tab vertical)
12	0Ch	FF	(form feed)
13	0Dh	CR	(retorno de carro)
14	0Eh	SO	(shift Out)
15	0Fh	SI	(shift In)
16	10h	DLE	(data link escape)
17	11h	DC1	(device control 1)
18	12h	DC2	(device control 2)
19	13h	DC3	(device control 3)
20	14h	DC4	(device control 4)
21	15h	NAK	(negative acknowle.)
22	16h	SYN	(synchronous idle)
23	17h	ETB	(end of trans. block)
24	18h	CAN	(cancel)
25	19h	EM	(end of medium)
26	1Ah	SUB	(substitute)
27	1Bh	ESC	(escape)
28	1Ch	FS	(file separator)
29	1Dh	GS	(group separator)
30	1Eh	RS	(record separator)
31	1Fh	US	(unit separator)
127	20h	DEL	(delete)

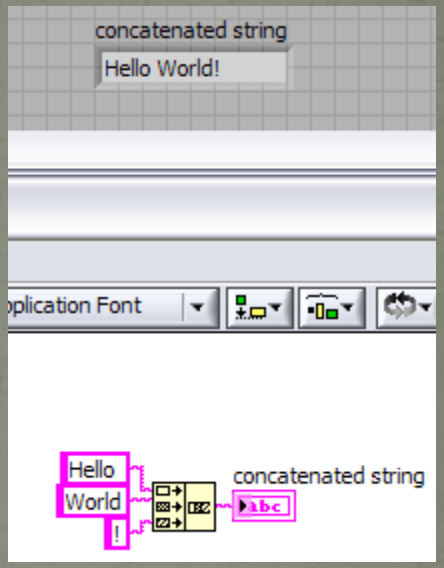
ASCII printable characters

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`
33	21h	!	65	41h	A	97	61h	a
34	22h	"	66	42h	B	98	62h	b
35	23h	#	67	43h	C	99	63h	c
36	24h	$	68	44h	D	100	64h	d
37	25h	%	69	45h	E	101	65h	e
38	26h	&	70	46h	F	102	66h	f
39	27h	'	71	47h	G	103	67h	g
40	28h	(	72	48h	H	104	68h	h
41	29h	)	73	49h	I	105	69h	i
42	2Ah	*	74	4Ah	J	106	6Ah	j
43	2Bh	+	75	4Bh	K	107	6Bh	k
44	2Ch	,	76	4Ch	L	108	6Ch	l
45	2Dh	-	77	4Dh	M	109	6Dh	m
46	2Eh	.	78	4Eh	N	110	6Eh	n
47	2Fh	/	79	4Fh	O	111	6Fh	o
48	30h	0	80	50h	P	112	70h	p
49	31h	1	81	51h	Q	113	71h	q
50	32h	2	82	52h	R	114	72h	r
51	33h	3	83	53h	S	115	73h	s
52	34h	4	84	54h	T	116	74h	t
53	35h	5	85	55h	U	117	75h	u
54	36h	6	86	56h	V	118	76h	v
55	37h	7	87	57h	W	119	77h	w
56	38h	8	88	58h	X	120	78h	x
57	39h	9	89	59h	Y	121	79h	y
58	3Ah	:	90	5Ah	Z	122	7Ah	z
59	3Bh	;	91	5Bh	[	123	7Bh	{
60	3Ch	<	92	5Ch	\	124	7Ch	|
61	3Dh	=	93	5Dh	]	125	7Dh	}
62	3Eh	>	94	5Eh	^	126	7Eh	~
63	3Fh	?	95	5Fh	_			

theASCIIcode.com.ar

Concatenating Strings

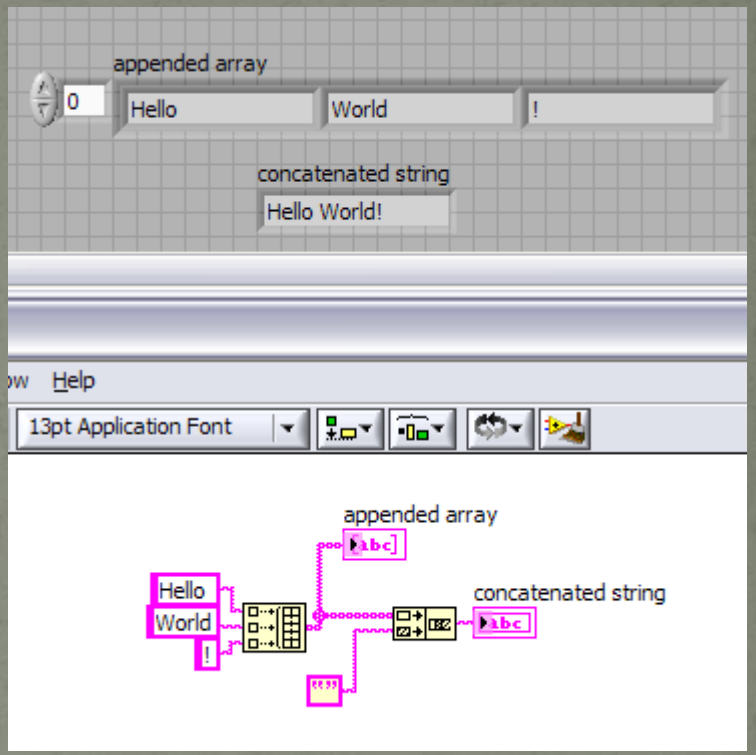
Concatenating Strings is probably the most used string function. It allows you to combine several strings into one string.



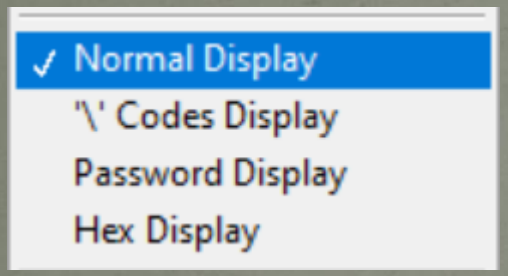
Concatenating Strings

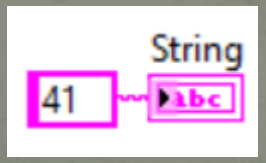
Here we concatenated an array of strings to form one string.

The Concatenate String vi requires 2 inputs, so I placed an empty string as the 2nd input.



String Display





If you right-click on a string constant, you will see the following options.

Normal is regular display.

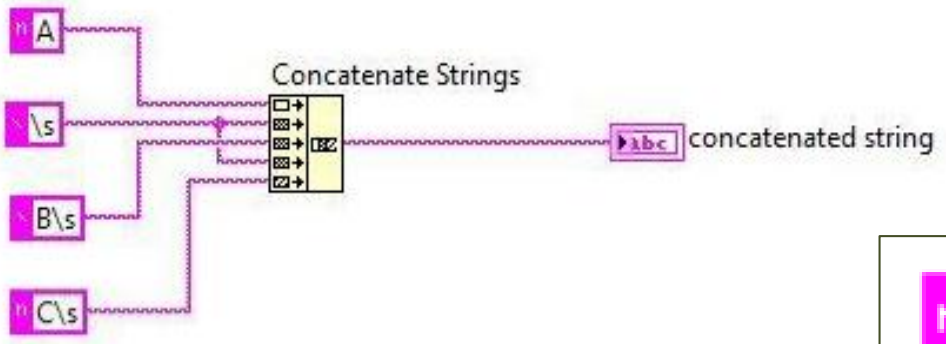
'\' Codes is C syntax where you can use \n for new line or \s for space.

Password shows asterisks.

Hex is the ASCII code for the string. For instance this code snippet produces the letter A.

CLAD Question

Q33: What will be the value of the **concatenated string** indicator after the VI completes execution?



A

A B C\s concatenated string

B

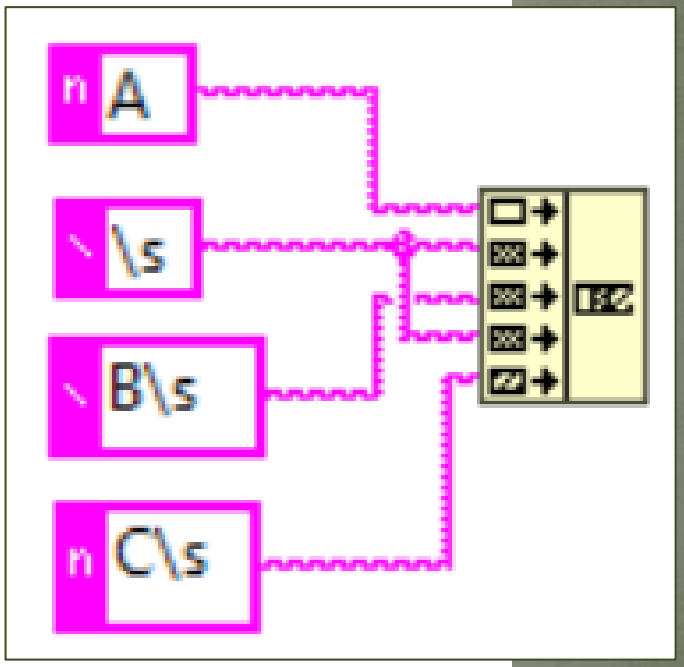
ABC concatenated string

C

A\sB\s\sC\s concatenated string

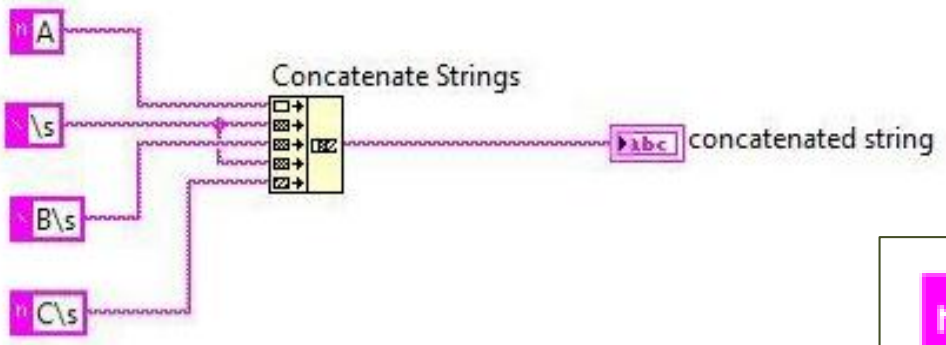
D

A B C concatenated string



CLAD Question

Q33: What will be the value of the **concatenated string** indicator after the VI completes execution?



A

A B C\s concatenated string

B

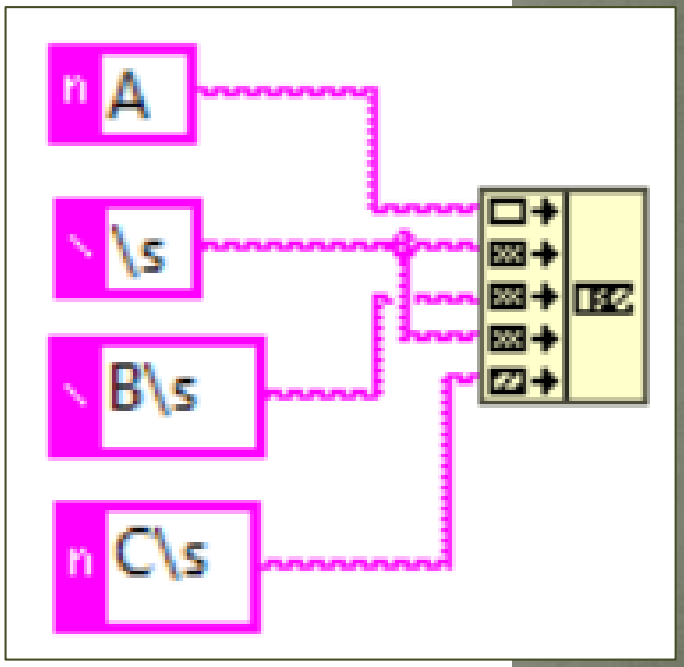
ABC concatenated string

C

A\sB\s\sC\s concatenated string

D

A B C concatenated string



CLAD Question

Which of the following display options are available for strings on the Front Panel?

a. '\' Codes
b. Password
c. Hex
d. All of the above

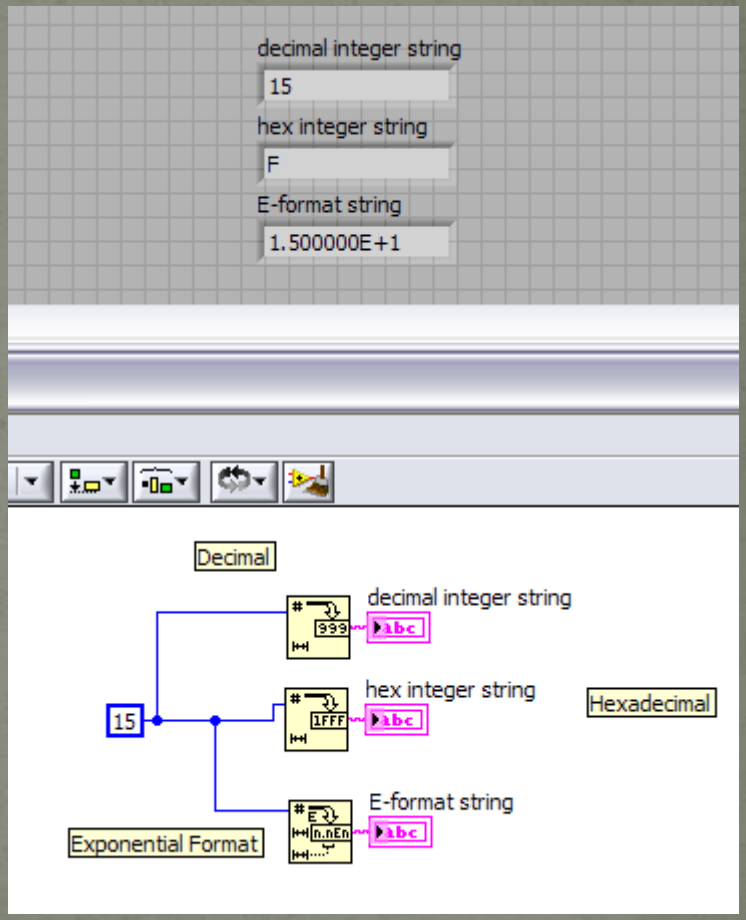
CLAD Question

Which of the following display options are available for strings on the Front Panel?

a. '\' Codes
b. Password
c. Hex
d. All of the above

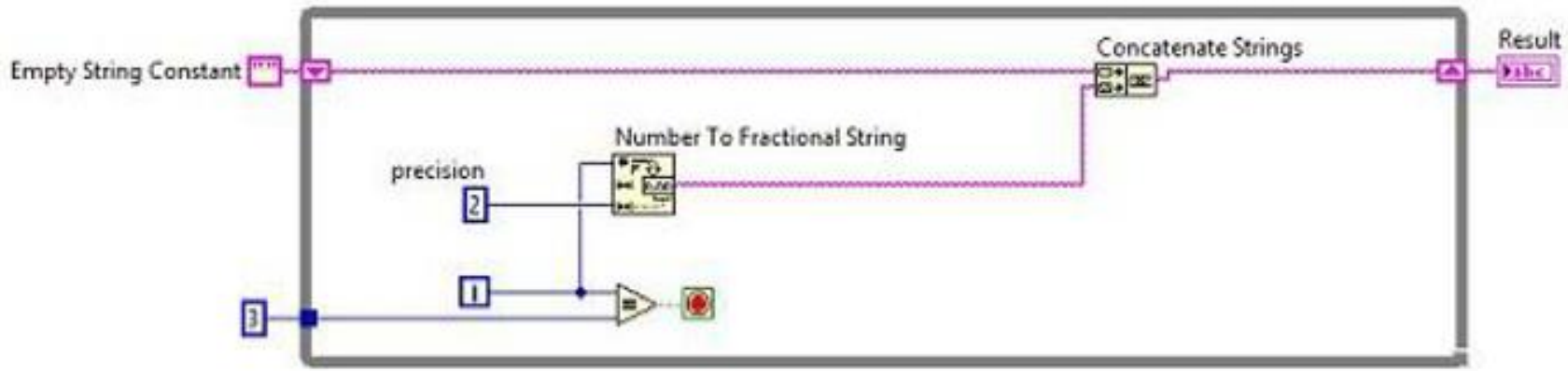
String to Number conversion (and vice versa)

This example shows how you can convert to Decimal, Hexadecimal, or Exponential format.



CLAD Question

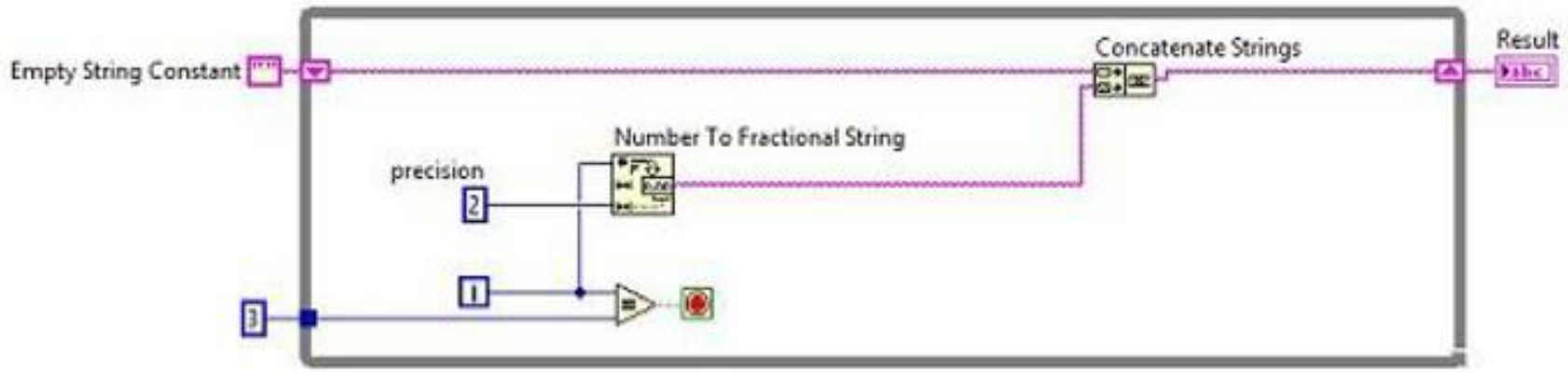
Q35: What string is displayed in the **Result** indicator after the VI completes execution?



A	123
B	0123
C	1.00 2.00 3.00
D	0.001.002.003.00

CLAD Question

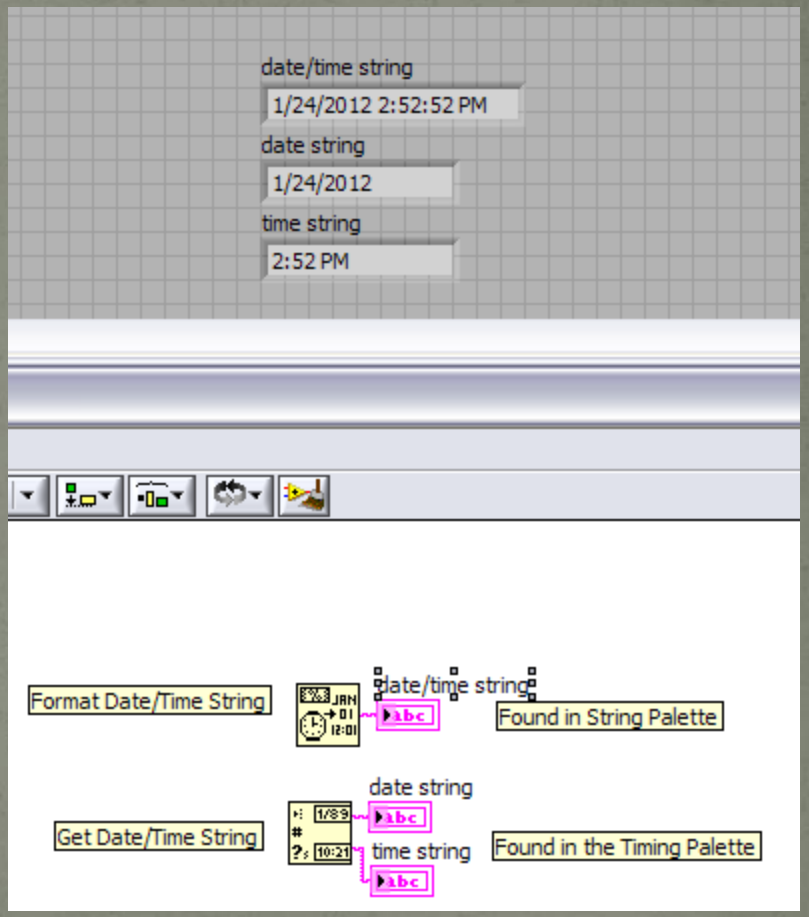
Q35: What string is displayed in the **Result** indicator after the VI completes execution?



A	123
B	0123
C	1.00 2.00 3.00
D	0.001.002.003.00

Date & Time String

You can grab the date and time from with the following VIs. This is important when collecting data so that you have this record.

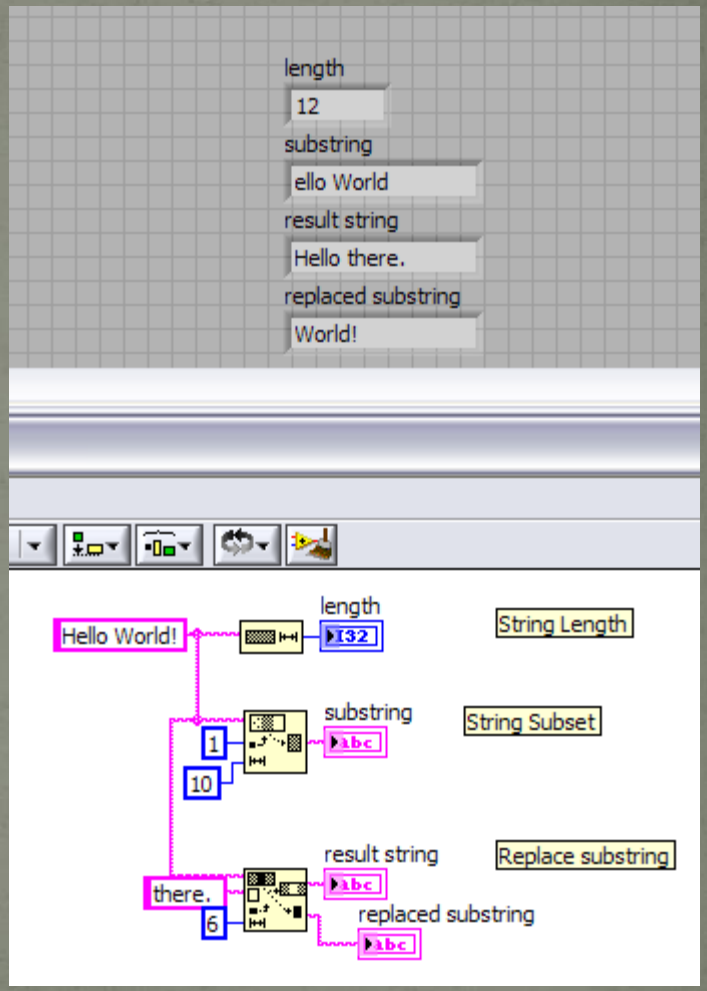


Other String Commands

You can get the string length.

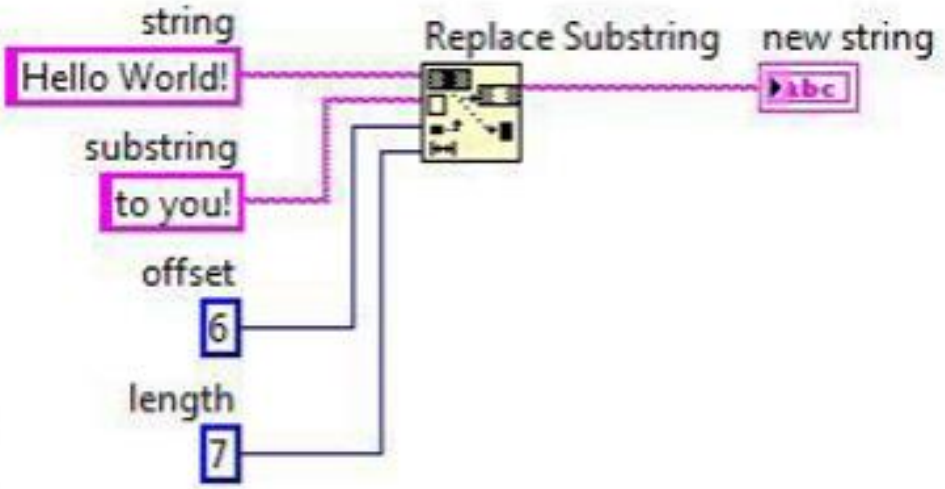
You can take part of the string (substring).

You can replace part of the string.



CLAD Question

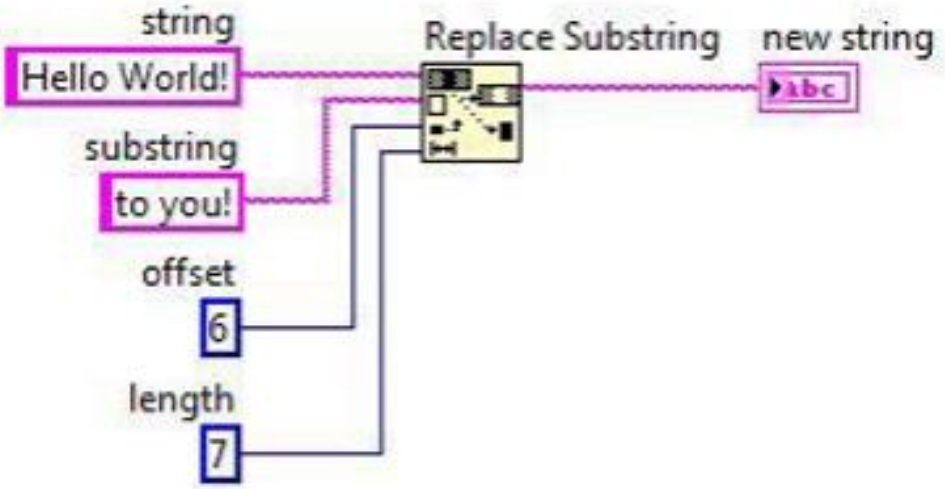
What is the result in **new string** after the following code has executed?



a. Hello to you!
b. Hello Wto you!
c. Hello to you!!
d. Helloto you!

CLAD Question

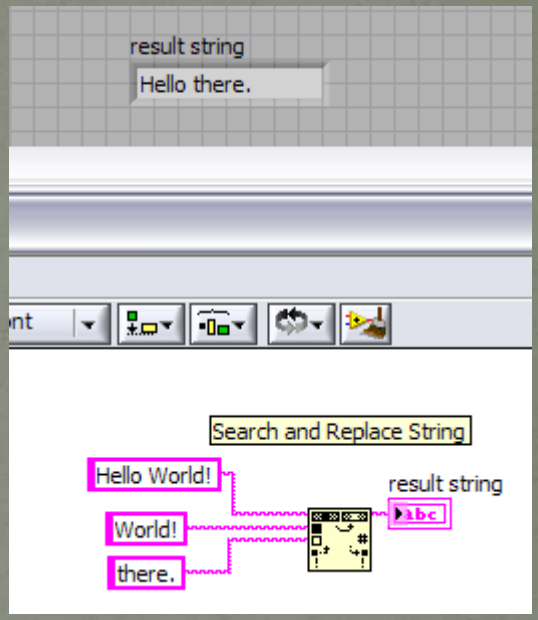
What is the result in **new string** after the following code has executed?



a. Hello to you!
b. Hello Wto you!
c. Hello to you!!
d. Helloto you!

Other String Commands

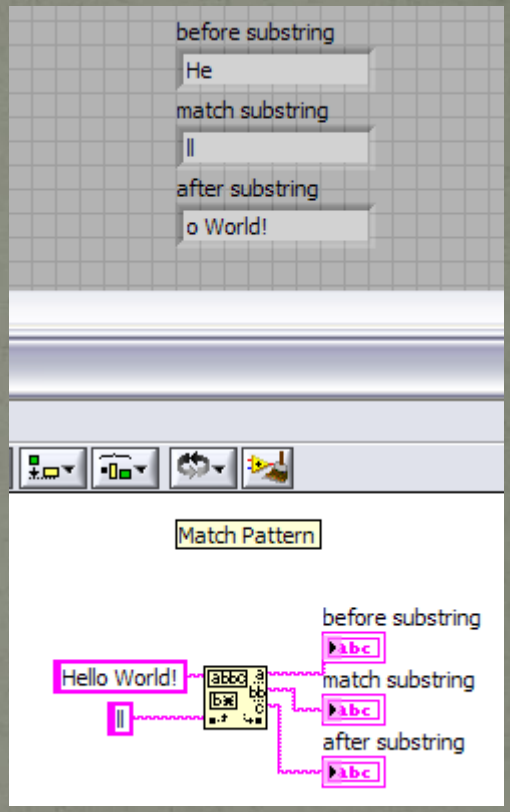
Search and Replace string searches for a particular string and replaces with another string.



Other String Commands

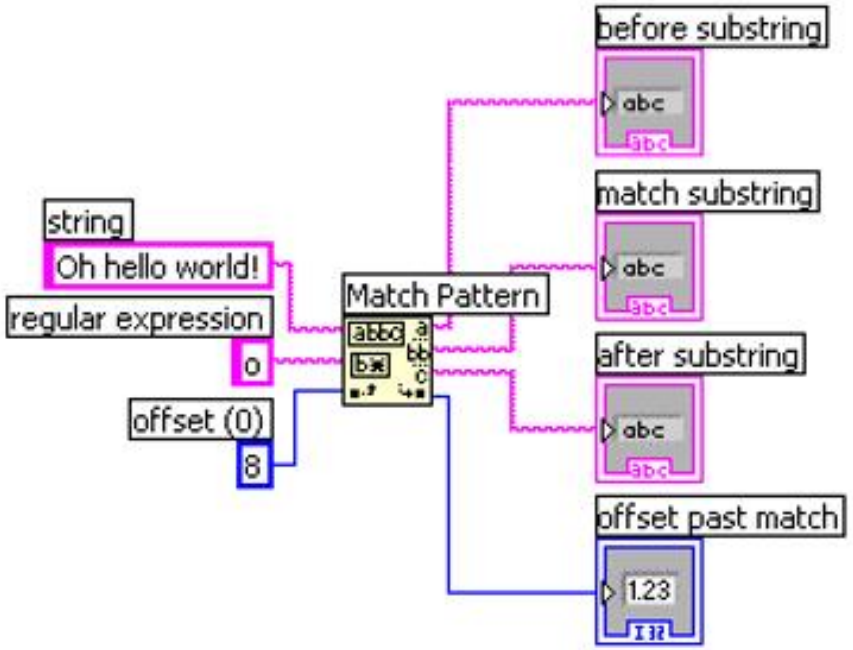
Match Pattern matches a string or part of a string and splits the result into 3 categories (before the string, the matched string, and after the string).

In this case, we're searching for "ll".



CLAD Question

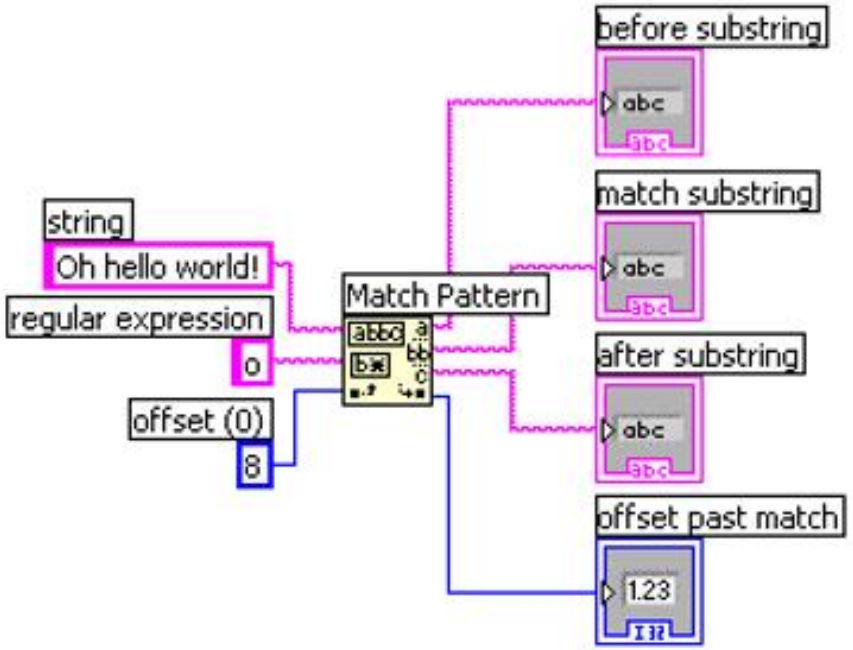
What is the value in **after substring** upon completion of the following code?



a. world!
b. rld!
c. h hello world!
d. <blank>

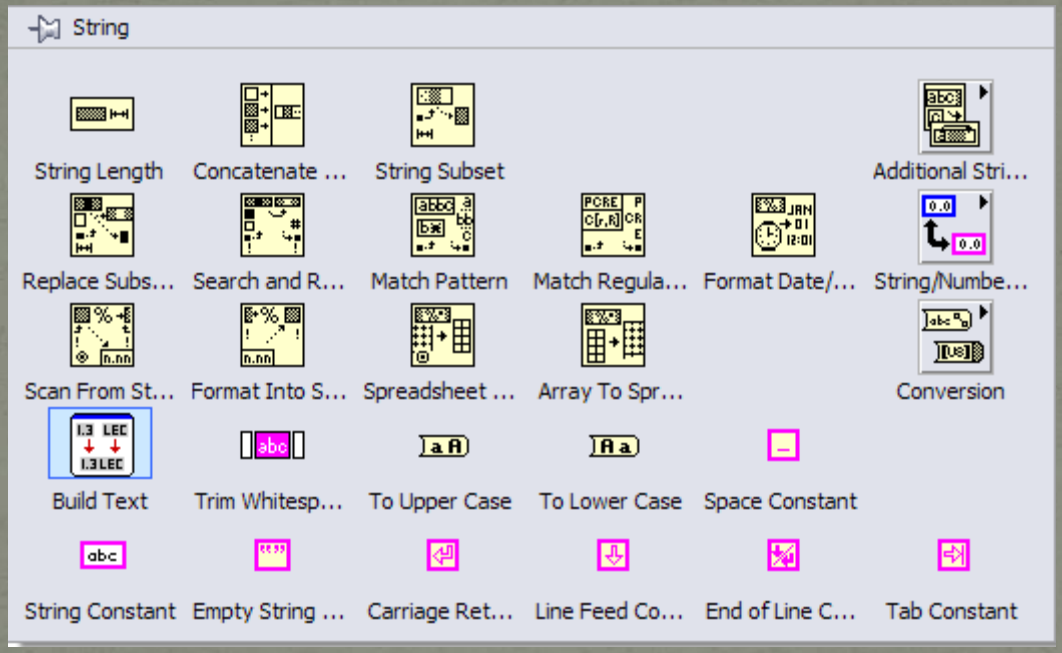
CLAD Question

What is the value in **after substring** upon completion of the following code?

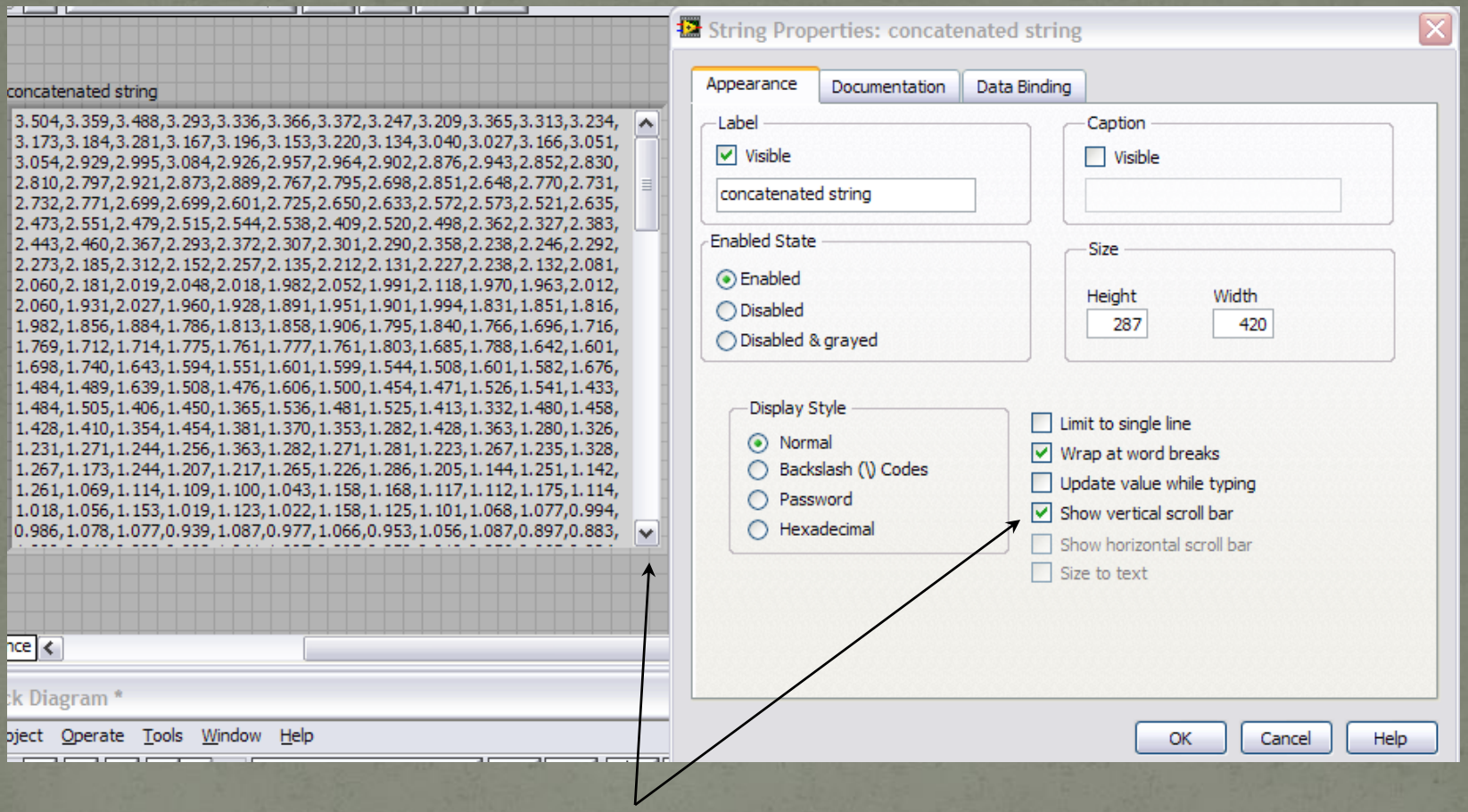


a. world!
b. rld!
c. h hello world!
d. <blank>

String Palette



Vertical Scroll Bar



If you are trying to display a string that is too big for a String Indicator, then you should add Show Vertical Scroll Bar found in properties.

PROPERTY NODES

Property Nodes allows you to programmatically control the properties of a front panel object.

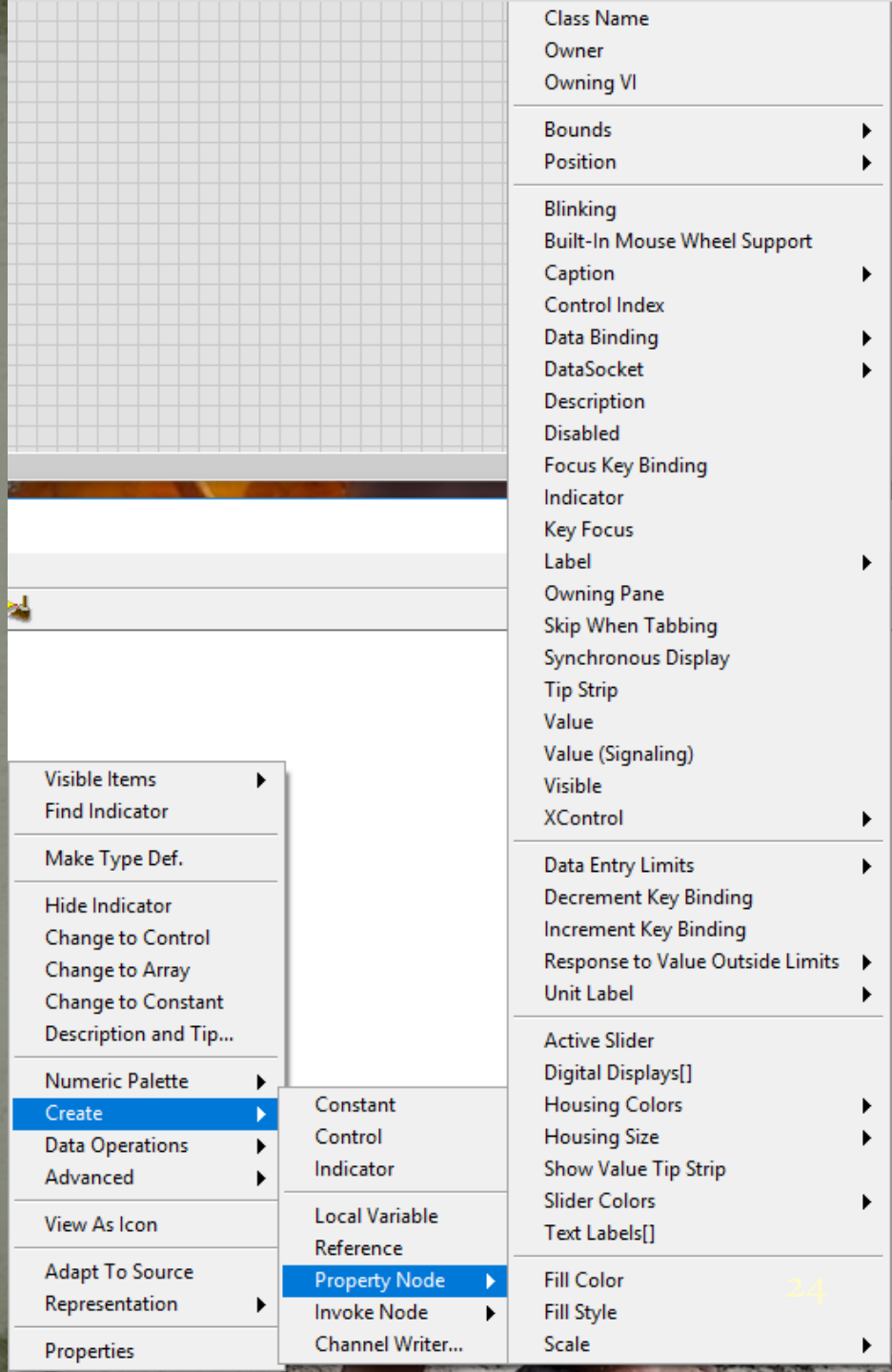
- Properties that can be changed . . .
- Color
- Visibility
- Make items blink
- Scale (max & min)

And lots of other things!

Creating Property Nodes

To create a Property Node, right-click on the object > Create > Property Node

Notice all the options that are available!

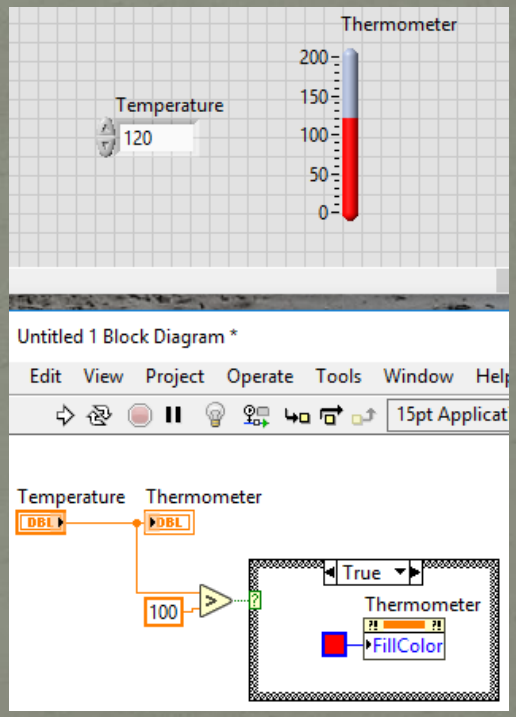


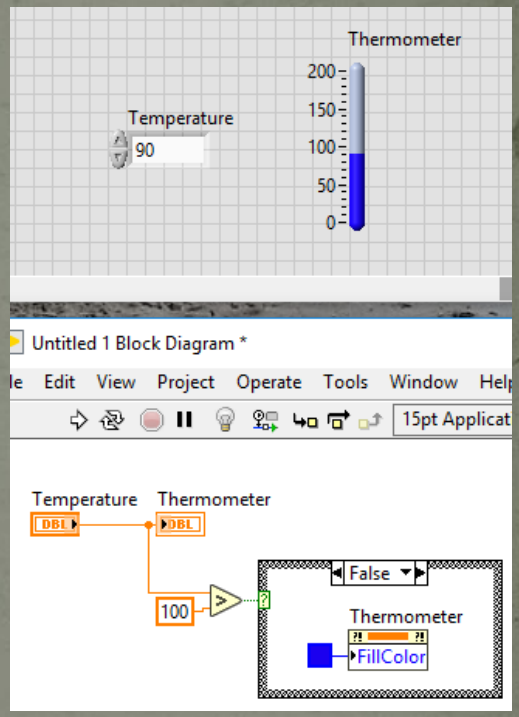
Property Nodes

Property Nodes make your LabVIEW program more powerful.

One example I read online was that you could be monitoring a temperature sensor that links to a Thermometer on your VI.

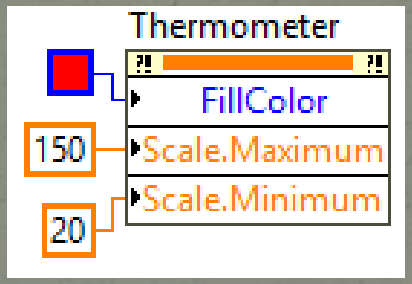
If the temperature is cool, the thermometer could be blue and when the temperature is warm, the thermometer could be red *(although physicists know that red is really colder than blue)*.



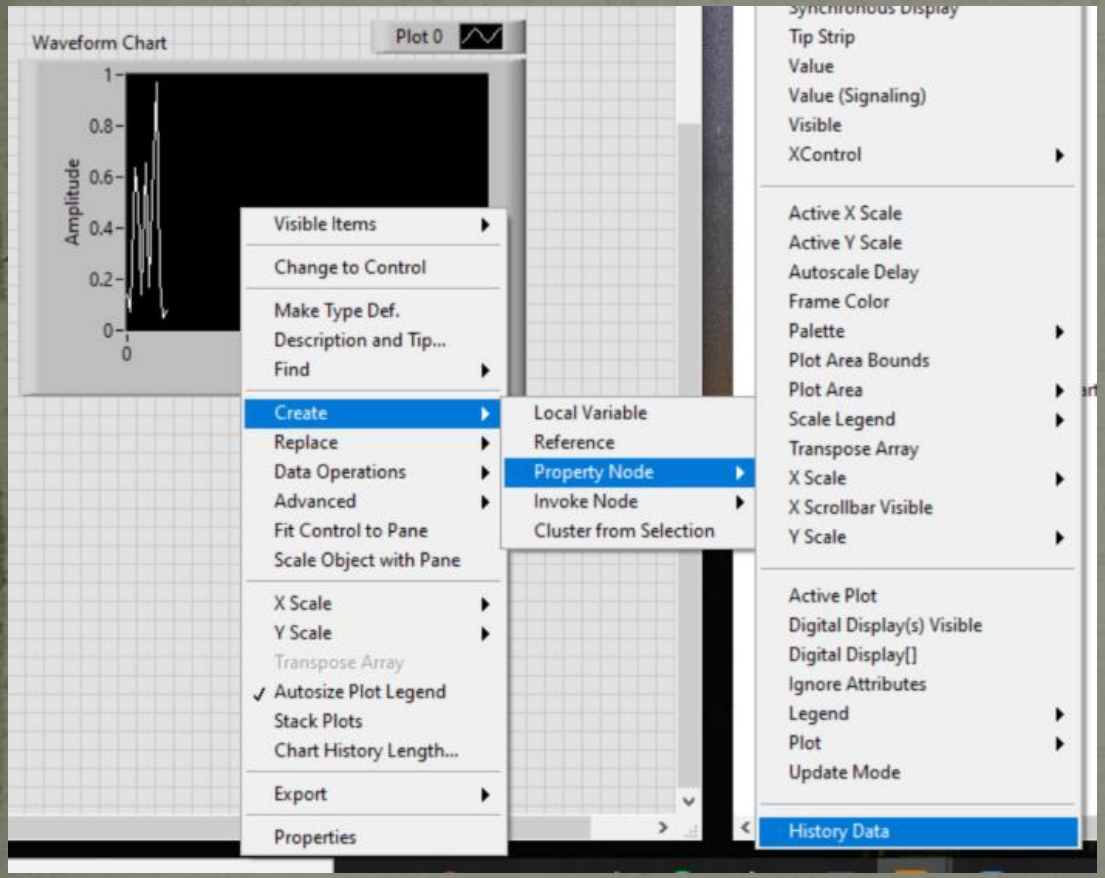


Multiple Properties

You can modify multiple properties by dragging down the property nodes. Note that the order at which the different properties are modified are in order from top to bottom.



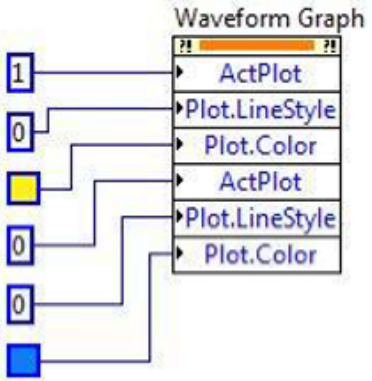
Property Node to Clear Chart



A property node used often to clear a chart is "History Data". Feed in an empty array into the property node and it will clear the chart.

Clad question

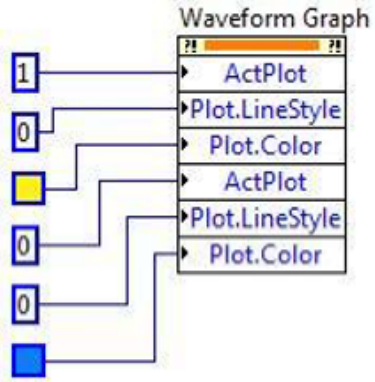
20. Which plot will change color first?



a. Plot 1 because properties are executed top-down
b. Plot 0 because properties are implemented in numeric order starting at zero
c. Both plots will be updated simultaneously due to the multithreading of properties
d. It cannot be determined because LabVIEW performs operations in dataflow format

Clad question

20. Which plot will change color first?



a. Plot 1 because properties are executed top-down
b. Plot 0 because properties are implemented in numeric order starting at zero
c. Both plots will be updated simultaneously due to the multithreading of properties
d. It cannot be determined because LabVIEW performs operations in dataflow format

You can change multiple properties of an item with one property node if you drag the property node downward.

If you are changing multiple properties, then it executes in a top-down fashion.

Clad question

Which of the following apply to Property Nodes? **(More than one answer may apply.)**

a. Property Nodes allow attributes of front panel objects to be programmatically manipulated.
b. Property Nodes can be used to update the values contained in a front panel object.
c. More than one Property Node can be used for a single front panel object.
d. Property Nodes can be used to programmatically generate a Value Change event.

Clad question

Which of the following apply to Property Nodes? **(More than one answer may apply.)**

a. Property Nodes allow attributes of front panel objects to be programmatically manipulated.

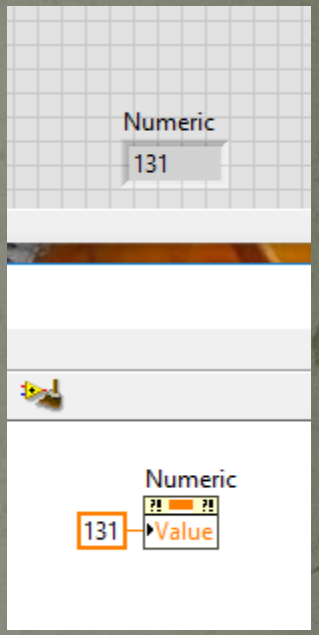
b. Property Nodes can be used to update the values contained in a front panel object

c. More than one Property Node can be used for a single front panel object.

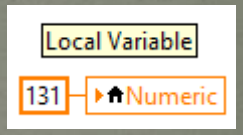
d. Property Nodes can be used to programmatically generate a Value Change event.

Property Nodes (value change) and Local Variables

Sometimes you need to change an indicator at multiple places in the code. If this output is a numeric, string, etc., then you won't be able to connected two inputs to one output. However, you can change it multiple places using either a property node (value change) or a local variable.

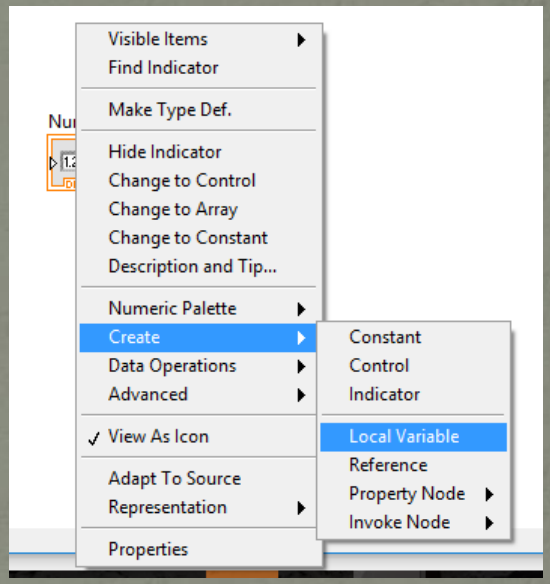


Local variables can only change the value of the item (not any other properties).

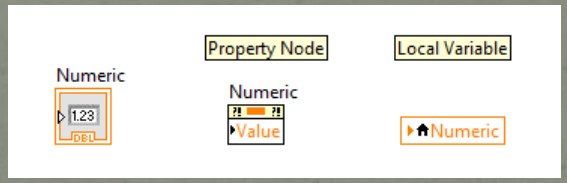


Local Variables

To create a local variable, right-click on the icon > Create > Local Variable



Property Node (Value change) Vs. Local Variable



Which is better . . . using property node (value) or using a local variable?

Evidently, local variables came first.

One difference between the two is that property nodes have an error in and error out to it, which can help with flow control. Also, local variables create a copy in memory whereas property nodes call the value by reference. Therefore, local variables will use more memory. However, local variables are supposed to be faster, so it's a speed vs. memory decision.

With the programs that we write, you probably won't see much difference between using either item.

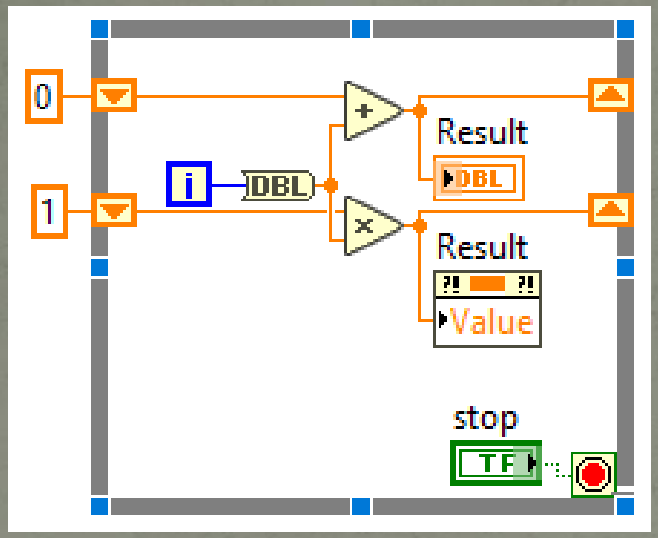
Flow control

Typically, LabVIEW prefers that programmers not use local variables or property nodes (value) because you can have scenarios where the same variable is getting written to in two different places at the same time. They suggest trying to use wires whenever possible.

However, I find that with good programming practice, local variables and property nodes are really convenient and sometimes the only option. I tend to use them often.

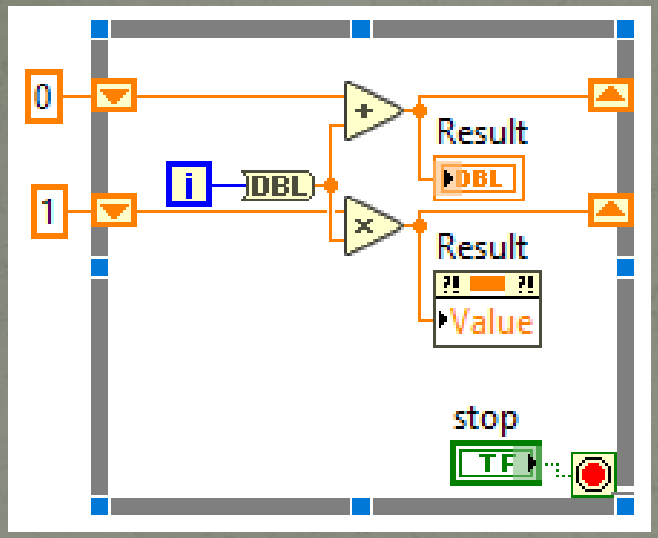
Flow control

What is wrong with the following code?



Flow control

What is wrong with the following code?



Result is written to in two different places concurrently. BAD PRACTICE!

Clad question

Q38: Which of the following apply to Property Nodes?

Property Nodes:

A	allow attributes of files on disk to be programmatically manipulated
B	can be used to update the value of a front panel control or indicator
C	return an error if you attempt to read a property before it has been written
D	can be used to invoke methods on a control

Clad question

Q38: Which of the following apply to Property Nodes?

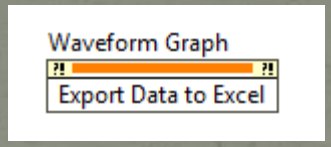
Property Nodes:

A	allow attributes of files on disk to be programmatically manipulated
B	can be used to update the value of a front panel control or indicator
C	return an error if you attempt to read a property before it has been written
D	can be used to invoke methods on a control

INVOKE NODES

Invoke Nodes are similar to property nodes, but instead of changing a value or another property of an item, invoke nodes executes some action.

For example, invoke nodes can reinitialize values to default, export a graph's image, or export data to Excel.



This Invoke Node exports data from a graph to Excel and automatically opens Excel

Clad question

Q39: Which do you use to initialize all front panel objects to their default values?

A	Application Reference
B	Invoke Node
C	User interface event
D	User event

Clad question

Q39: Which do you use to initialize all front panel objects to their default values?

A	Application Reference
B	Invoke Node
C	User interface event
D	User event