

# Integer to String example

Block editor, A = B.ToString()[3] block

## Expression

A: \$String7

Name	\$String7
Assign Default	Constant
Content Type	String
Default Value	

Content Type

B: (Number const)

Int	7
-----	---

Int

C: (null)

Generate expression

Result: \$String7 = Expression: B.ToString()

Test

Comment

OK Cancel

# String to Integer example

Block editor, A = Int32.Parse(B)[2] block

## Expression

A

Stest

Name	Stest
Assign Default	Constant
Content Type	Number
Default Value	0

Content Type

B

(String const)

String	5
--------	---

String

C

(null)

Generate expression

Result	Expression
Stest	= Int32.Parse(B)

Test

Comment

OK Cancel

# Increase Point Values

Block editor, A = Point.Add(B, new Size(C))[6] block

## Expression

A

\$FinalPoint

Name	\$FinalPoint
Assign Default	Constant
Content Type	Point
Default Value	{X=0,Y=0}
X	0
Y	0

Content Type

B

\$FinalPoint

Name	\$FinalPoint
Assign Default	Constant
Content Type	Point
Default Value	{X=0,Y=0}
X	0
Y	0

Content Type

C

\$PointAdder

Name	\$PointAdder
Assign Default	Constant
Content Type	Point
Default Value	{X=10,Y=10}
X	10
Y	10

Content Type

Generate expression

Result	Expression
\$FinalPoint	= Point.Add(B, new Size(C))

--->> Testing...  
Result = {X=10,Y=10}  
Test is OK.  
Completed in 172ms

Test

Comment

OK

Cancel

# Random Number Less Than 1000

Block editor, Expression[2] block

**Expression**

A: \$Number (new)

Name	<b>\$Number</b>
Assign Default	<b>Constant</b>
Content Type	<b>Number</b>
Default Value	<b>0</b>

Content Type

B: (null)

C: (null)

Generate expression

Result: \$Number      Expression: = new Random().Next(1000)

---->> Testing...  
Result = 631  
Test is OK.  
Completed in 157ms

Test

Comment

OK      Cancel

# Example of Splitting a String In Countdown Expression Block

Block editor, A = B.Split(',') [C][2] block

**Expression**

A: \$Substring

Name	\$Substring
Assign Default	Constant
Content Type	String
Default Value	

Content Type

B: (String const)

String	one, two, three, four
--------	-----------------------

String

C: \$Index

Name	\$Index
Assign Default	Constant
Content Type	Number
Default Value	0

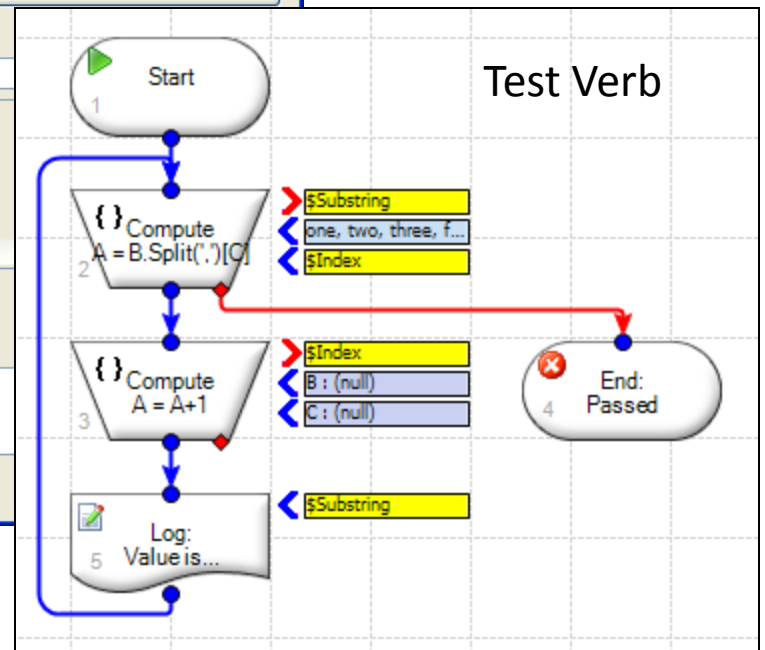
Content Type

Generate expression

Result: \$Substring = B.Split(',') [C]

Expression: B.Split(',') [C]

Comment:



Here is C# code that will substitute an underscore for a space.

```
string initial = "test 1";  
Console.Out.WriteLine( initial.Replace(' ', '_'));
```

Resulting output is: “test\_1”

Here is an assignment command that could be used in a Countdown TestCase

Assignment:

```
A = B.Replace( ' ', '_')
```



### Expression

**A**

\$ResultString

Name	\$ResultString
Assign Default	Constant
Content Type	String
Default Value	

Content Type

**B**

(String const)

String "Spaces"

String

**C**

(null)

String

Generate expression

Result      Expression

\$ResultString      =      B.Replace(' ', '\_')

Test

Comment

OK      Cancel

TestDesigner - [ChangeCharacter - Execute mode]

File Edit View Test Case Editor Debug Window Help

Repository Explorer

- Administrator@JPOMPLUND610
  - TestSets
  - TestCases
    - PPC5.0GoCalendarTh
    - Start
    - on Pocket PC 2
    - \*ChangeCharacter
  - TestVerbs
  - Navigation Maps
  - Platforms
  - Virtual Devices
  - Devices
  - Global Variables
  - Data Sources
  - External Assemblies

100%

```

    graph TD
      Start([1 Start]) --> Compute[/2 Compute A = B.Re.../]
      Compute --> End([3 End: Passed])
  
```

Start Page ChangeCharacter

Log Pane

ChangeCharacter

Filter By: Debug 35 items, 0 pics (0 Mb)

Time	Message
09.06.29 19:30:25.166	Expression: \$ResultString = B.Replace('.', '_') == "S_p_a_c_e_s"
09.06.29 19:30:25.166	End Block: True
09.06.29 19:30:25.166	ChangeCharacter run completed. Exit at End: 3. Status = "Passed"
09.06.29 19:33:31.755	Start ChangeCharacter
09.06.29 19:33:31.755	Start
09.06.29 19:33:32.177	Expression: \$ResultString = B.Replace('.', '_') == "S_p_a_c_e_s"
09.06.29 19:33:32.177	End Block: True
09.06.29 19:33:32.177	ChangeCharacter run completed. Exit at End: 3. Status = "Passed"