

User Interface - Bug #4130

FWD truncates decimal part instead of rounding when displaying

07/04/2019 12:54 AM - Alexei Kaigorodov

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:	Alexei Kaigorodov	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			
Related issues:			
Related to User Interface - Bug #3691: literal \$ character in format appears ...			New

History

#1 - 07/04/2019 01:10 AM - Alexei Kaigorodov

- File test.p added

the article [<https://knowledgebase.progress.com/articles/Article/000030560>] reads:

FORMAT rounds the displayed value when there are more digits stored to the right of the decimal point than the format can show. This behavior is not configurable.

But in case the format has no place for decimal fraction at all, FWD truncates the number instead of rounding.

Thus, the following program:

```
display string(DECIMAL("99.876"), ">>>.99") skip.  
shows " 99.88" both on 4gl and FWD.
```

But

```
display string(DECIMAL("-99.876"), "->>>.") skip.  
shows "-100" on 4gl and "-99" on FWD.
```

#2 - 07/04/2019 11:27 AM - Alexei Kaigorodov

- Assignee set to Alexei Kaigorodov

#3 - 07/05/2019 01:23 AM - Alexei Kaigorodov

So I have a short (1 line) 4gl test program:

```
display string(DECIMAL("-99.876"), "->>>.") skip.
```

I converted it to a short (43 lines) Java program named com.goldencode.testcases.format_string.b4130.Test and would like to run it against the FWD runtime.

The only way I know is to start server, and then to start client driver with argument pointing to the source test program.

But I would like to run it faster, as a junit test, using some simple wrapper:

```
com.goldencode.Launcher.launch(com.goldencode.testcases.format_string.b4130.Test.class);
```

Is it possible?

#4 - 07/05/2019 04:55 AM - Alexei Kaigorodov

made a fix in `secure/code/p2j_repo/p2j/active/4130a/`
Now the test `uast/format_string/abl/makeCheck.p` shows full compatibility with 4gl.

Please review the fix.

#5 - 07/05/2019 06:32 AM - Greg Shah

But I would like to run it faster, as a junit test, using some simple wrapper:

```
com.goldencode.Launcher.launch(com.goldencode.testcases.format_string.b4130.Test.class);
```

Is it possible?

Not at this time.

#6 - 07/05/2019 08:33 AM - Greg Shah

Code Review Task Branch 4130a Revision 11322

1. Please explain the logic behind your change. On the surface it is not clear why it is correct. For example, if the decimal point is at the end of the format string (`decPt == (len - 1)`), what is right and why should expected not be 0?
2. Running the ChUI regression testing can be used as a reality check.
3. Each file that is changed needs a history entry.

#7 - 07/06/2019 10:10 PM - Alexei Kaigorodov

Greg Shah wrote:

2. Running the ChUI regression testing can be used as a reality check.

How to run regression testing?

#8 - 07/08/2019 12:06 AM - Alexei Kaigorodov

Greg Shah wrote:

Code Review Task Branch 4130a Revision 11322

1. Please explain the logic behind your change. On the surface it is not clear why it is correct. For example, if the decimal point is at the end of the format string (`decPt == (len - 1)`), what is right and why should expected not be 0?

because expected is the number of digits after decimal point in the raw number. It must never be changed. It was set to 0 only to force the condition in the subsequent check if (`size < expected`) to return false in case there are no digits right to the decimal point in the formatted string. The idea was that rounding has effect only if we need to display digits after the point, which appear to be wrong in our case of `string(DECIMAL("-99.876"), "->>>.")`. Even if the idea were right, it should have been done without the change of the expected variable.

#9 - 07/08/2019 12:50 AM - Alexei Kaigorodov

Greg Shah wrote:

3. Each file that is changed needs a history entry.

Done.

#10 - 07/16/2019 09:05 AM - Greg Shah

- Related to Bug #3691: literal \$ character in format appears on the right side of the fill-in added

#11 - 07/30/2021 05:43 PM - Roger Borrello

- Start date deleted (07/04/2019)

I looked over these `uast/format_string` tests, and got them running on 3821c_12737. There are errors being reported on FWD:

```
rfb@rfb:~/testcases/uast/format_string/bin$ cat ../generated/check_dec_frame.err
decFrameFormatter(abc->, >>>, >>9, 0.12)='abc 0'; but 'abc0' expected.
decFrameFormatter(abc->, >>>, >>9, 1.23)='abc 1'; but 'abc1' expected.
decFrameFormatter(abc->, >>>, >>9, 23.45)='abc 23'; but 'abc23' expected.
decFrameFormatter(abc->, >>>, >>9, 99.876)='abc 100'; but 'abc100' expected.
decFrameFormatter(-zzz., 99.876)='99.'; but '100.' expected.
decFrameFormatter(-***., 99.876)='*99.'; but '100.' expected.
decFrameFormatter(-999., 99.876)='099.'; but '100.' expected.
decFrameFormatter(->>>., 99.876)='99.'; but '100.' expected.
decFrameFormatter(abc->, >>>.99, 0.12)='abc .12'; but 'abc.12' expected.
decFrameFormatter(abc->, >>>.99, 1.23)='abc 1.23'; but 'abc1.23' expected.
decFrameFormatter(abc->, >>>.99, 23.45)='abc 23.45'; but 'abc23.45' expected.
decFrameFormatter(abc->, >>>.99, 99.876)='abc 99.88'; but 'abc99.88' expected.
rfb@rfb:~/testcases/uast/format_string/bin$ cat ../generated/check_dec_string.err
```

```
decStringFormatter(->>>, 1.23)=' 1'; but ' 1' expected.
decStringFormatter(->>>, 23.45)=' 23'; but ' 23' expected.
decStringFormatter(->>>, >>>, >>>, 1.23)=' 1'; but ' 1' expected.
decStringFormatter(->>>, >>>, >>>, 23.45)=' 23'; but ' 23' expected.
decStringFormatter(->>>, >>>, >>>, 99.876)=' 100'; but ' 100' expected.
decStringFormatter(+>>>>>>, 0.12)='+'; but ' +' expected.
decStringFormatter(+>>>>>>, 1.23)='+1'; but ' +1' expected.
decStringFormatter(+>>>>>>, 23.45)='+23'; but ' +23' expected.
decStringFormatter(+>>>>>>, 99.876)='+100'; but ' +100' expected.
decStringFormatter(>>>), 1.23)=' 1 ' ; but ' 1 ' expected.
decStringFormatter(>>>), 23.45)=' 23 ' ; but ' 23 ' expected.
decStringFormatter($>>>>), 0.12)='$ ' ; but ' $ ' expected.
```

I'm not sure of the difference between the string and frame versions of the tests, but it looks like there is a problem with the > formatting.

How was this task left off? It could be related to the handling of the > in general.

#12 - 08/02/2021 05:39 AM - Greg Shah

Work was paused and 4130a is an active branch with pending changes that have now diverged greatly from trunk and 3821c.

Files

test.p	92 Bytes	07/04/2019	Alexei Kaigorodov
--------	----------	------------	-------------------