

Base Language - Feature #2063

Support negative numbers defined with the "-" on the right side

02/27/2013 04:31 AM - Costin Savin

Status:	Closed	Start date:	02/27/2013
Priority:	Normal	Due date:	
Assignee:	Costin Savin	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	Conversion Support for Server Features	vendor_id:	GCD
billable:	No		
Description			
Related issues:			
Related to Base Language - Bug #2065: ambiguity in date/decimal literals that...			Closed

History

#1 - 02/27/2013 04:37 AM - Costin Savin

Statements like:

```
def var i as int.  
i = 99-. /*this fails*/  
message i.
```

will fail with error:

```
Rule Type : WALK  
Source AST: [ 99- ] BLOCK/ASSIGNMENT/ASSIGN/EXPRESSION/NUM_LITERAL/ @2:5 {227633266708}  
Copy AST : [ 99- ] BLOCK/ASSIGNMENT/ASSIGN/EXPRESSION/DEC_LITERAL/ @2:5 {227633266708}  
Condition : putNote("oldtypeStr", prog.num_literal)  
Loop : false  
--- END RULE REPORT ---
```

Strangely enough this works in 4GL and the variable will be assigned the negative number -99

#2 - 02/27/2013 06:13 AM - Costin Savin

- File cs_upd20130227a.zip added

This update fixes conversion side but there is still work to do to have the correct value set.

#3 - 02/27/2013 06:44 AM - Constantin Asofiei

Update looks good, this is going through conversion regression testing now.

#4 - 02/27/2013 06:58 AM - Eric Faulhaber

- *Project changed from Liberty to Base Language*

#5 - 02/27/2013 08:02 AM - Costin Savin

- *File cs_upd20130227b.zip added*

Added proposed update which corrects the assignment.

#6 - 02/27/2013 08:18 AM - Constantin Asofiei

Please test with these cases too:

```
def var i as int.  
def var i64 as int64.  
def var d as dec.  
  
i = 9-.  
i64 = 9999999999999999-.  
d = 1.99992-.
```

Greg: if we use the "1.2-" as a dec literal in a d = 1.2- assignment, the parser converts it as a date_literal. Should we fix this now too ?

#7 - 02/27/2013 09:06 AM - Greg Shah

We must be very careful here.

```
def var d1 as date init 1.2-.  
def var d2 as date init 1.12-.  
message d1 d2.
```

Both of these are valid date literals, at last when used in the init clause. BUT they are also BOTH valid decimal literals and Progress handles this based on context. Outside of the init clause, it seems that the 4GL may limit this syntax somewhat. We can reduce the potential problems by testing the months and only triggering this "conflict" when the "months" are 1 through 12. But we will still have issues.

We already do some "morphing" of initializer types. This may be a place where we have to do that again.

But I don't want to mix all this work together with the current problem. It is very sensitive. We will work it later.

#8 - 02/27/2013 09:08 AM - Greg Shah

The coding changes look OK. Moving the sign to the left was what I was going to propose. Make sure you remove the extra `println()` in `cleanup.rules`.

#9 - 02/27/2013 09:28 AM - Costin Savin

- File `cs_upd20130227c.zip` added

Added proposed update

#10 - 02/27/2013 09:30 AM - Greg Shah

I'm OK with it. Put it through conversion testing.

#11 - 02/27/2013 11:17 AM - Constantin Asofiei

Has passed conversion regression testing (no changes in generated sources), please check in and distribute.

#12 - 02/27/2013 11:39 AM - Costin Savin

Committed to bazaar as revision number 10220.

#13 - 02/27/2013 12:12 PM - Greg Shah

- % Done changed from 0 to 100

- Status changed from WIP to Closed

#14 - 02/28/2013 04:30 PM - Constantin Asofiei

We have a case where the `999.99-` literal is converted to `new date("999.99-")` instead of decimal:

```
book.price = 999.99-.
d = 999.99-.
```

is converted to:

```
book.setPrice(new date("999.99-")); /* compile error */
d.assign(new date("999.99-"));
```

I think we should consider the lvalue (in simple assignments case), when determining the data type of the literal.

#15 - 02/28/2013 06:10 PM - Greg Shah

The lexer cannot easily change its behavior in response to feedback from the parser. This is a core limitation of ANTLR and is caused by the fact that the lexer has its own lookahead and is called in an unpredictable pattern which by its nature is often well ahead of what is being processed in the parser. By the time we know in the parser that the `DEC_LITERAL` is a top-level rvalue being assigned to a `VAR_DATE`, it is too late.

But there is a much simpler solution. We can implement bounds checking on the month and day fields, such that only a small number of overlapping cases are wrong.

#16 - 11/16/2016 11:07 AM - Greg Shah

- Target version changed from Milestone 4 to Conversion Support for Server Features

Files

cs_upd20130227a.zip	5.48 KB	02/27/2013	Costin Savin
cs_upd20130227b.zip	7.52 KB	02/27/2013	Costin Savin
cs_upd20130227c.zip	7.55 KB	02/27/2013	Costin Savin