

## Bugs - Bug #1454

### progress lexer fails to parse comments without space before

07/20/2012 11:17 AM - Ovidiu Maxiniuc

<b>Status:</b>	New	<b>Start date:</b>	07/20/2012
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>case_num:</b>	
<b>billable:</b>	No	<b>version:</b>	
<b>vendor_id:</b>	GCD		
<b>Description</b>			

### History

#### #1 - 07/20/2012 11:26 AM - Ovidiu Maxiniuc

antlr fails to parse correctly progress code if the inline comment is not separated by a space.

```
define var b as char init "b".
b = b/*this will not be parsed correctly*/.
b = b./*same here*/
```

The lexer will split into <FILENAME> b/ respectively b./, <MULTIPLY> \* and so on.

Normally, the antlr parser cannot handle this stream of tokens and will prompt you with unexpected token: @b/ message and following stack:

```
at com.goldencode.p2j.uast.ProgressParser.primary_expr(ProgressParser.java:36154)
at com.goldencode.p2j.uast.ProgressParser.chained_object_members(ProgressParser.java:50586)
at com.goldencode.p2j.uast.ProgressParser.un_type(ProgressParser.java:50544)
at com.goldencode.p2j.uast.ProgressParser.prod_expr(ProgressParser.java:50425)
at com.goldencode.p2j.uast.ProgressParser.sum_expr(ProgressParser.java:34668)
at com.goldencode.p2j.uast.ProgressParser.compare_expr(ProgressParser.java:50031)
at com.goldencode.p2j.uast.ProgressParser.log_not_expr(ProgressParser.java:49907)
at com.goldencode.p2j.uast.ProgressParser.log_and_expr(ProgressParser.java:49848)
at com.goldencode.p2j.uast.ProgressParser.expr(ProgressParser.java:15440)
at com.goldencode.p2j.uast.ProgressParser.assignment(ProgressParser.java:5539)
at com.goldencode.p2j.uast.ProgressParser.single_block(ProgressParser.java:4276)
at com.goldencode.p2j.uast.ProgressParser.block(ProgressParser.java:4035)
at com.goldencode.p2j.uast.ProgressParser.external_proc(ProgressParser.java:3961)
at com.goldencode.p2j.uast.AstGenerator.parse(AstGenerator.java:1395)
```

The problem seems to be with variable tokenizer. Converting the following code will be successful:

```
define var b as char init "b".
b = "a"/*ok here*/.
b = "a"./*same here*/
```

Also, adding or removing some spaces after the comment do not change the conversion result.

**#2 - 07/20/2012 11:28 AM - Ovidiu Maxiniuc**

Note from GES:

Progress 4GL lexing and parsing is quite complex. Their design included many ambiguities and conflicts that are resolved by using some undocumented rules that are implemented deep in the Progress compiler. The lexing and parsing of the "." DOT character is especially troublesome. Progress uses the DOT for many different purposes. They require the DOT to be followed by whitespace in some cases to allow their lexer/parser to resolve the ambiguity (I am guessing here at the intention since I have never actually seen their lexing/parsing code, but I know that it helped me out when I was writing the code <g>).

However, in this specific case, I think this may be a bug. Until we have access to a 4GL dev environment, I can't test this specific case. But based on my review of the `com/goldencode/p2j/uast/progress.g`, I don't see that we intended this behavior.

**#3 - 07/24/2012 09:33 AM - Ovidiu Maxiniuc**

After investigating the `src/com/goldencode/p2j/uast/progress.g` source I realized that when defining SYMBOL (line 28945) the rules are too greedy to "eat" more chars.

Anyway, I don't know how the real Progress works here so, in order to keep the same syntax and semantics nothing should be altered until the test on a true P4GL is done.

**#4 - 07/24/2012 10:11 AM - Greg Shah**

Can you narrow down the part of the SYMBOL rule that is the problem? If so, please provide a snippet in PRE tags here. Then you can stop working this (since we don't have a real 4GL system on which to test).

**#5 - 11/15/2016 04:47 PM - Greg Shah**

- Assignee deleted (Greg Shah)