Conversion Tools - Bug #2066

brew adds unnecessary comma in method parameter list after comment

02/27/2013 01:59 PM - Eric Faulhaber

Status:	Closed		Start date:	02/28/2013
Priority:	Normal		Due date:	
Assignee:	Constantin Asofiei		% Done:	100%
Category:			Estimated time:	0.00 hour
Target version:	Conversion Support for Server Features			
billable:	No		case_num:	
vendor_id:	GCD		version:	
Description				

History

#1 - 02/27/2013 02:06 PM - Eric Faulhaber

On a whim last night, I tried to compile the partially converted customer app project, and quickly hit an "illegal start of expression" error in many places. This was due to something like:

```
void myMethod( /* comment */, arg1, ...)
```

It looks like the test in brew's next-child rules for a star comment when emitting the comma is looking ahead instead of behind in this case.

#2 - 02/28/2013 01:21 AM - Constantin Asofiei

- Status changed from New to WIP

- Start date set to 02/28/2013

- Assignee set to Constantin Asofiei

#3 - 02/28/2013 01:44 AM - Constantin Asofiei

I suggest to disable comments, run conversion in the good directories and compile again, to uncover non-comment problems.

#4 - 02/28/2013 03:01 AM - Constantin Asofiei

- File ca_upd20130228a.zip added

Fix attempt for star comments mixed with children of IF, method arguments, and others. Will test with MAJIC and some server dirs.

#5 - 02/28/2013 05:12 AM - Ovidiu Maxiniuc

I believe this was also addressed in the #1450. Unfortunately it was not finalized.

#6 - 02/28/2013 08:41 AM - Constantin Asofiei

- File ca_upd20130228b.zip added

This fixes these cases:

```
i = (/* 1 */ if /* 2 */ func0(/* 3 */ 1, /* 4 */ 1, /* 5 */ 1 /* 6 */) /* 7 */ = /* 8 */ 1 /* 9 */ then /* 10
*/ /* 222 */ /* 333 */ 1 /* 11 */ else /* 12 */ 11 /* 13 */).

assign /* 1a */ i /* 2a */ = /* 3a */ 1 /* 4a */
       /* 5a */ book.book-title /* 6a */ = /* 7a */ "2" /* 8a */
       /* 9a */ book.isbn /* 10a */ = /* 11a */ "asdasdasdasd" /* 12a */.

assign /* 1b */ i /* 2b */ = /* 3b */ 1000 /* 4b */.
```

#7 - 02/28/2013 09:09 AM - Greg Shah

The code looks fine. And that is some testcase! It made me laugh to read it. Well done.

#8 - 02/28/2013 09:10 AM - Constantin Asofiei

I still have some regressions in it, I'm testing and will re-upload.

#9 - 02/28/2013 04:15 PM - Constantin Asofiei

The conclusion for today, after trying a scoping approach to solve some nested calls issues is this, in brew phase (a few examples):
1. this.getChildAt(nextChildIndex - 1).type should be written as this.getPrevChildNonStar().type
2. nextChildIndex > 1 should be written as getNonStarIndex(this, nextChildIndex) > 1
3. nextChildIndex 1 should be written as getNonStarIndex(this, nextChildIndex) 1
4. and others

What I'm trying to say here is that it will be cleaner (and easier) to modify the next-child-rules part to act as if the star nodes are not there. As we have only a move-forward phase provided by next-child-rules section where we can act on children, and not a "child-finished-rules" section to be able to look backwards, trying to fix next-child-rules feels like chasing my own tail. It will be very ugly and in the end I will still not be sure that I will cover all cases.

#10 - 03/01/2013 06:57 AM - Constantin Asofiei

- File ca_upd20130301a.zip added

Made next-child-rules to be ignorant of star-comment nodes, I'm putting this through conversion regression testing.

#11 - 03/01/2013 07:53 AM - Constantin Asofiei

- File ca_upd20130301b.zip added

Replaced getNextSibling with ignore-star version.

#12 - 03/01/2013 09:12 AM - Greg Shah

I am fine with the changes. If they pass conversion testing, let's go with it.

#13 - 03/01/2013 09:13 AM - Constantin Asofiei

I'm running conversion now.

#14 - 03/01/2013 09:24 AM - Constantin Asofiei

I'm stopping conversion, a server folder has failed with some other comment related issue. I need to get rewrite at least the following too:

1. numberOfChildren
2. nextSibling
3. getChildAt
4. numImmediateChildren

#15 - 03/01/2013 11:41 AM - Constantin Asofiei

- File ca_upd20130301d.zip added

I'm testing this version with the server files which had comment errors, if everything is good I move to conversion regression testing.
Note that I left the next-child-rules section un-indented, to ease review.

#16 - 03/01/2013 11:53 AM - Greg Shah

Well done. It looks good. Painful, though. I have to think about this problem a bit to see if there is a way to help this in the future in TRPL.

#17 - 03/01/2013 01:40 PM - Constantin Asofiei

OK, the server's compilable folders do not seem to report any comment-related errors. I'm putting this through conversion testing (comments disabled).

#18 - 03/01/2013 02:48 PM - Constantin Asofiei

Conversion has finished, compiled fully, but there are some regressions in terms of:

1. new-lines no longer emitted in some cases (i.e. between inner classes)
2. additional ';' character emitted on an empty line or at the end of a "//" comment

I'm running another conversion with comments enabled, to check if it compiles. I will fix the two issues before releasing the update.

#19 - 03/01/2013 03:40 PM - Constantin Asofiei

MAJIC has compiled with comments enabled (and AFAIK this was not possible before this update).

#20 - 03/01/2013 03:42 PM - Greg Shah

WOW! That is very cool. Great job!

#21 - 03/05/2013 03:46 PM - Constantin Asofiei

- File ca_upd20130305c.zip added

This update passes conversion regression testing with comments off (no changes in generated sources), and MAJIC does compile with comments on. Next I will leave a conversion of all server folders over night, with comments enabled, and compare the build logs with the comments off versions - there should be no differences.

#22 - 03/06/2013 05:18 AM - Constantin Asofiei

The server folders had no errors related to comments.

#23 - 03/06/2013 07:13 AM - Constantin Asofiei

Committed to bzr revision 10251.

#24 - 03/06/2013 08:26 AM - Greg Shah

- % Done changed from 0 to 100

- Status changed from WIP to Closed

#25 - 03/16/2013 04:40 PM - Constantin Asofiei

There is a problem when the comment is emitted after the peer:

```
def var i as int.
case i:
when 0 then /* v0 */
  assign i = i + 0.
when 1 then /* v1 */
  assign i = i + 0.
end.
```

converts to:

```
            switch ((i).intValue())
            {
              case 0:
              {
                i.assign(plus(i, 0));
                break;
              }
 /* v0 */             case 1:
              {
                i.assign(plus(i, 0));
                break;
              }
 /* v1 */;
            }
```

Note the extra semicolon emitted after the last comment. This is caused by a change in brew.xml, but if I remove that change, other cases get regressed.

At this time, I'm not sure if we really need to emit the comment after the code where it was in the legacy code... can you explain a little why you chose this approach in comments.rules?

#26 - 03/19/2013 04:06 AM - Constantin Asofiei

wrong task

#27 - 03/19/2013 04:12 AM - Constantin Asofiei

wrong task

#28 - 03/19/2013 04:22 AM - Constantin Asofiei

wrong task

#29 - 03/22/2013 04:09 PM - Constantin Asofiei

- File ca_upd20130322i.zip added

This is a quirk for the SWITCH statement comment case. At this time, I think convert/comments.rules might be incomplete (idea is, why not skip intermediate artifical nodes with no line numbers, when searching for peers?), but if the quirk passes conversion regression testing and the server folders accept it, we should go with it for now.

#30 - 03/22/2013 04:22 PM - Greg Shah

I agree. The approach you have taken is fine for now.

I also agree that we will need to make additional changes to the comments processing. At a minimum, the way we associate comments with surrounding code needs improvement.

The idea of bypassing artificial nodes sounds good, but I have to look carefully at the impact before I can really say.

Anyway, if this passes testing, you can release it.

#31 - 03/23/2013 10:04 AM - Constantin Asofiei

ca_upd20130322i.zip was committed to bzr revision 10320. MAJIC conversion regression testing has passed (with comments disabled and enabled), and also the server folders seem to have accepted it (no comment related errors in logs).

#32 - 11/16/2016 10:58 AM - Greg Shah

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

Files

ca_upd20130228a.zip	7.63 KB	02/28/2013		Constantin Asofiei
ca_upd20130228b.zip	7.79 KB	02/28/2013		Constantin Asofiei
ca_upd20130301a.zip	7.99 KB	03/01/2013		Constantin Asofiei
ca_upd20130301b.zip	7.97 KB	03/01/2013		Constantin Asofiei
ca_upd20130301d.zip	8.24 KB	03/01/2013		Constantin Asofiei
ca_upd20130305c.zip	8.36 KB	03/05/2013		Constantin Asofiei
ca_upd20130322i.zip	8.49 KB	03/22/2013		Constantin Asofiei