

Conversion Tools - Bug #11837

preprocessor expands a TAB byte to spaces where OE emits the byte

09/06/2026 01:32 PM - Greg Shah

Status:	Review	Start date:	
Priority:	Normal	Due date:	
Assignee:	Paula Păstrăguș	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:			
billable:	No	reviewer:	Greg Shah
vendor_id:	GCD	production:	No
case_num:		env_name:	
version_reported:		topics:	4GL Preprocessor
version_resolved:			
Description			
Related issues:			
Related to Conversion Tools - Bug #10099: Tilde consumes the octals/follow-up...			Internal Test
Related to Conversion Tools - Bug #6308: post string literal alternative codi...			Internal Test

History

#1 - 09/06/2026 01:33 PM - Greg Shah

OE : txt = "text"~~~t<TAB> . FWD: txt = "text"~~~t~~~~~ .
OE : c = "A<TAB> B" . FWD: c = "A~~~~B" .

Failing tests: alternative_coding_quirk_06 (pre-existing, currently mis-attributable to [#10099](#)), define_tilde_t_tab, define_octal_011_tab.

#2 - 09/06/2026 01:33 PM - Greg Shah

- Related to Bug #10099: Tilde consumes the octals/follow-up characters, when in OE this doesn't happen added

#3 - 09/06/2026 01:33 PM - Greg Shah

- Related to Bug #6308: post string literal alternative coding quirk in 4GL preprocessor added

#4 - 09/08/2026 03:54 AM - Paula Păstrăguș

- Status changed from New to WIP

I'm looking into this one.

#5 - 09/08/2026 07:13 AM - Paula Păstrăguș

I've created task branch 11837a.

#6 - 09/08/2026 09:26 AM - Paula Păstrăguș

The issue here is that OE expands a literal tab typed in the source (stop 8, absolute column) but emits the raw byte for a tab produced by an escape (~t, ~011). Currently, FWD expands both types of tabs in TextLexer.tab() because it evaluates the character without knowing its provenance.

Deleting the two lines in tab() that rewrite the buffer appears to fix the escaped tabs and reports "0 regressions". However, this is a false positive: our current test suite has no input containing a literal tab. Implementing this change would silently break FWD's ability to match OE byte-for-byte on literal source-tab cases.

To properly cover this gap and prevent future false positives, I will also add the 5 new TAB test cases I've generated to the test suite. This ensures both literal and escaped tabs are correctly validated.

I am currently looking for a proper fix for this.

Greg, please let me know if I have the green light to upload my tab tests to the testcases project as a new test set named **tab_expansion_test_set** into the preprocessor test_plan.xml?

- tab_from_escape.p

```
&scoped-define x "A~tB"
define variable c as character no-undo.
c = {&x}.
message c.
```

- tab_in_code.p

```
define variable c as character no-undo.
c = "a".
c = "b".
c = "c".
message c.
```

- tab_in_comment.p

```
/* A B */
define variable c as character no-undo.
message c.
```

- tab_in_define.p

```
&scoped-define x "A B"
define variable c as character no-undo.
c = {&x}.
message c.
```

- tab_in_string.p

```
define variable c as character no-undo.
c = "1 |".
c = "12 |".
c = "1234567 |".
c = "12345678 |".
message c.
```

#7 - 09/08/2026 09:31 AM - Greg Shah

Yes, you can add your preprocessor tests. Also, please make sure to update the wiki page.

#8 - 09/08/2026 09:55 AM - Paula Păstrăguș

I committed the tests to the testcase project as rev 1909 and updated the current status table on the Preprocessor Testcases wiki to reflect the new test suite.

#9 - 09/08/2026 10:34 AM - Paula Păstrăguș

Greg, implementing this fix requires modifications to ClearStream.java as well as the text.g and braces.g grammars. Should I follow the same process we used for [#11835](#): commit it here for review first, and then port the revision once it is approved?

#10 - 09/08/2026 10:58 AM - Greg Shah

Yes

#11 - 09/09/2026 05:22 AM - Paula Păstrăguș

- Status changed from WIP to Review
- Assignee set to Paula Păstrăguș
- % Done changed from 0 to 100
- reviewer Greg Shah added

I committed the fix as rev **16746**:

- ClearStream now queues the origin of each emitted tab (source vs produced).
- Grammars (text.g, braces.g) pop this queue to skip expansion for produced tabs.
- Tabs inside expanded macro text are explicitly re-marked as produced, since literal source tabs are already replaced by spaces during the first pass. Refs: [#11837](#).

Greg, please review.

#12 - 09/09/2026 05:43 AM - Paula Păstrăguș

Testing was completed on branch **11747a**. The fix successfully resolves the two original cases (define_tilde_t_tab and define_octal_011_tab), as well as the tab_from_escape test. As a result, the entire tab_expansion_test_set is now passing.

- With the fix: **117 PASSED, 24 FAILED**
- Without the fix: **114 PASSED, 27 FAILED**

#13 - 09/09/2026 05:57 AM - Paula Păstrăguș

- File alt.png added

As for alternative_coding_quirk_06, the TAB differences were resolved by the changes on this branch, but it is still failing on a unicode escape (~u0042B in OE against ~uu0042 in FWD, line 69) and on a trailing backslash that FWD splits onto its own line, making the file 73 lines against a 72 line baseline.



alt.png	6.01 KB	09/09/2026	Paula Păstrăguș
---------	---------	------------	-----------------