

Conversion Tools - Bug #11835

-keeptildes keeps escape text for macro-expanded content, where OE emits only the byte

09/06/2026 01:25 PM - Greg Shah

Status:	Review	Start date:	
Priority:	Normal	Due date:	
Assignee:	Paula Păstrăguș	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		reviewer:	Greg Shah
billable:	No	production:	No
vendor_id:	GCD	env_name:	
case_num:		topics:	4GL Preprocessor
version_reported:			
version_resolved:			
Description			
Related issues:			
Related to Conversion Tools - Bug #10099: Tilde consumes the octals/follow-up...			Internal Test
Related to Conversion Tools - Bug #10237: keepTildes is keeping tilde when pr...			New

History

#1 - 09/06/2026 01:27 PM - Greg Shah

-keeptildes exists to emulate the [Progress Tilde Bug](#) — OE leaves tildes in the saved listing that it has already removed from the stream it compiles. That emulation is correct for escapes written directly in the source. It is **wrong for content that arrived by macro expansion**: OE resolves the escape during expansion and the listing carries only the resulting byte, with no tilde text to preserve. FWD keeps the text and appends the byte in both cases.

	OE listing	FWD -keeptildes
define_tilde_t_tab	c = "A<TAB> B".	c = "A~t <TAB>B".
define_octal_011_tab	c = "A<TAB> B".	c = "A~011 <TAB>B".
define_tilde_n_terminates	c = "A".	c = "A~n".
define_double_tilde_n	c = "AAA.~nBBB.".	c = "AAA.~~nBBB.".

The .preproc (no-keeptildes) output for all of these is **byte-identical to the OE baseline**, which confirms the defect is confined to the keeptildes path and that converted output is unaffected.

The wiki's [Known Behavioral Deviations](#) list has carried this as "No test asserts this: which behavior is correct is undecided, since -keeptildes exists to emulate an OE bug in the first place." **It is now decided by measurement** — OE has no tilde bug to emulate for expanded content — and it is now asserted by tests.

1. Testcases

Currently failing on this and nothing else:
define_double_tilde_n, define_tilde_n_terminates, define_octal_012_terminates, define_tilde_ctrl_escapes, keeptildes_escaped_braces_in_define, and the pre-existing tildes_in_preproc_defines.

Probably also printable_ascii_characters (def frame a~041. in OE versus def frame a~041!. in FWD — the same "text plus byte" shape); worth confirming when this is fixed.

2. Do not resolve this by changing what the tests compare

Comparing this class against .preproc instead of .keeptildes.preproc makes them pass, and it is tempting because the justification sounds principled. It is the wrong move: it would leave the defect in place and remove the only tests that detect it. Measured across the whole plan, 95 tests pass against .keeptildes.preproc and 15 pass **only** against it, so the keeptildes comparison is right for the suite; the five that pass only against .preproc are precisely the ones this issue is about.

#2 - 09/06/2026 01:28 PM - Greg Shah

- *topics 4GL Preprocessor added*

#3 - 09/06/2026 01:29 PM - Greg Shah

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

#5 - 09/08/2026 04:57 AM - Paula Păstrăguș

- *Status changed from New to WIP*

I'm looking into this one.

#6 - 09/08/2026 06:13 AM - Paula Păstrăguș

I've created task branch 11835a.

#7 - 09/08/2026 06:54 AM - Paula Păstrăguș

- *Related to Bug #10237: keepTildes is keeping tilde when preprocessor directive contains tilde and curly brace added*

#8 - 09/08/2026 06:59 AM - Paula Păstrăguș

- *Status changed from WIP to Review*

- *% Done changed from 0 to 100*

- *reviewer Greg Shah added*

The fix was committed as rev **16746**.

Greg, please review.

#9 - 09/08/2026 07:08 AM - Paula Păstrăguș

Testing was completed on branch **11747a**. Applying the fix successfully resolved the 6 specific tests mentioned in the initial note.

- With the fix: **110 PASSED, 26 FAILED**
- Without the fix: **104 PASSED, 32 FAILED**

The defect was at define-scan time. ClearStream.read() consulted keepTildes unconditionally, baking the escape text directly into the stored symbol value so every later expansion incorrectly reproduced it.

I resolved it by routing the eight keepTildes evaluations in read() through a new helper method that explicitly ignores define values:

```
return keepTildes && env != null && !env.isInDefine();
```

#10 - 09/08/2026 08:19 AM - Greg Shah

I'll review it here, but the change probably should be in 11747a where we have other preprocessor changes.

#11 - 09/08/2026 08:23 AM - Paula Păstrăguș

I'll keep that in mind. Please let me know if you would like me to port the revision once it gets the green light. (reviewed and properly tested, if the preprocessor suite is not enough to validate this).

#12 - 09/08/2026 08:26 AM - Greg Shah

Code Review Task Branch 11835a Revision 16746

I'm good with the change. Please apply the change to 11747a.

#13 - 09/08/2026 08:27 AM - Greg Shah

- Assignee set to Paula Păstrăguș

11747a is going to be heavily tested across many applications. Putting this change there will reduce overall testing and avoid merge conflicts.