Database - Bug #11893

CONTAINS NOT operator

09/19/2026 10:11 AM - Greg Shah

Status:	Rejected	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**reviewer:**	
vendor_id:	GCD	**production:**	No
case_num:		**env_name:**	
version_reported:		**topics:**	
version_resolved:			

Description

Related issues:

Related to Database - Feature #5219: implement native full-text search for wo...	**Test**

History

#1 - 09/19/2026 10:14 AM - Greg Shah

- Related to Feature #5219: implement native full-text search for word index and CONTAINS support added

#2 - 09/19/2026 10:15 AM - Greg Shah

CONTAINS treats the NOT operator as OR

Summary

In a CONTAINS search expression, FWD maps ! and ^ to a logical OR. If the 4GL treats ! as NOT — which the operator's name and conventional use both suggest — then every CONTAINS expression using it returns the wrong rows, silently and with no error.

The machinery for NOT is present and complete; it is simply never reached, because no lexer path produces it.

This needs an OpenEdge reference run to confirm before it is treated as a defect. The write-up below separates what was observed in the FWD source from what has not been established.

What the Code Does

src/com/goldencode/p2j/util/LogicalExpressionConverter.java:983-988, in mapLogicalCharacter:

```
case '&amp;':
   return Op.AND;
case '|':
case '!':
case '^':
   return Op.OR;
```

All three of |, ! and ^ collapse to Op.OR.

NOT Exists but Is Unreachable

Op.NOT is defined and the normal form conversion handles it — pushNots() and distribute() at :281-380 implement the usual CNF transformation including negation. But Op.NOT is only ever **consumed**, never **produced**: every occurrence in the file is inside toNormalForm / pushNots, and no lexer path emits it.

The SQL generator assumes as much. src/com/goldencode/p2j/persist/orm/FqlToSqlConverter.java:3683 asserts:

> CNF for a valid CONTAINS argument does not contain negations

So the pipeline is internally consistent — it is consistent around the assumption that CONTAINS expressions never negate.

Why This Matters

If ! is NOT in the 4GL, then:

- a & !b is compiled as a OR b — it will match records that contain b, which is the precise opposite of what was asked for
- there is no error, no warning and no log entry; the query simply returns the wrong rows
- the failure is data-dependent, so it will not show up in a smoke test and may not show up in casual use

A wrong-answer bug with no diagnostic is worse than a crash, and this one is in a query predicate, so it can quietly widen a result set.

There is a second-order effect too: (!word) would be parsed as an OR with a missing left operand and raise the spurious 4GL syntax error 2876 ("missing word before a '&' '|' or ')'").

Exposure

It is a latent defect for any project that uses negation in a CONTAINS expression.

What Has Not Been Established

The central question is unanswered: does OpenEdge treat ! as NOT in a CONTAINS search expression?

No OpenEdge reference was available while investigating, and nothing in the FWD tree documents the intended semantics of ! here. The FWD code is self-consistent, so it cannot be used to answer the question — it would look exactly the same whether the mapping is correct or wrong.

Also unestablished:

- what ^ means in the 4GL, if anything — it may be a synonym for NOT, a synonym for OR, or not an operator at all
- whether OpenEdge accepts a leading negation such as !word, and what it returns

Suggested Next Step

Run a small reference case against a real OpenEdge install — the fwd-openedge-reference workflow covers this — on a word-indexed field with known contents:

```
FOR EACH t WHERE t.f CONTAINS "alpha &amp;amp; !beta": ... END.
FOR EACH t WHERE t.f CONTAINS "alpha | beta":  ... END.
FOR EACH t WHERE t.f CONTAINS "alpha ^ beta":  ... END.
```

Three rows are enough to settle it: one containing only alpha, one containing only beta, one containing both. Which rows come back for the first query distinguishes NOT from OR unambiguously.

If ! is NOT, the fix is to emit Op.NOT from the lexer for ! and let the existing pushNots() / distribute() handle it — the CNF machinery is already there — and then remove the assumption at FqlToSqlConverter.java:3683 and generate the negated predicate.

How It Was Found

While assessing the CONTAINS gap marking (K3 in #11888-24). The marking question is written up separately in that task; that document recommends raising the level, which is unaffected by this defect because that project does not use the operator.

Sources

- src/com/goldencode/p2j/util/LogicalExpressionConverter.java, FWD branch 11747a
- src/com/goldencode/p2j/persist/orm/FqlToSqlConverter.java
- #11888-24 K3 — the marking assessment this came out of

#4 - 09/21/2026 09:03 AM - Stefanel Pezamosca

Verified on OpenEdge against FWD: ! and ^ are exact synonyms of | (OR), so FWD's mapping is correct and there is no wrong-answer bug.

71 search expressions against 15 data rows, run three ways — an OpenEdge temp-table, a real OpenEdge database table, and FWD's Contains.evaluate row-matching path — compared on row sets and 4GL error codes: **0 mismatches**.

The decisive cases:

```
expression              reading
alpha | beta            OR
alpha ! beta            OR
alpha ^ beta            OR
alpha & beta ! gamma    (a & b) | g
alpha ! beta & gamma    a | (b & g)
```

```
alpha ! beta          same rows as alpha | beta, including the row holding BOTH
                      words -> rules out NOT and XOR
alpha & beta ! gamma  (a & b) | g -- ! carries |'s precedence as well
alpha & !beta         error 2876 in OpenEdge and in FWD
!beta   (!beta)       error 2876 in both -- the 2876 the report calls spurious is in fact correct
```

There is no negation syntax at all: NOT, AND and OR are ordinary searchable words, -beta is equivalent to beta, and ~ is not an escape. Op.NOT is unreachable by design, not by omission.
I recommend closing this as not a defect.

#5 - 09/21/2026 09:21 AM - Ovidiu Maxiniuc

From my PoV, this is a hallucination of the AI.

I did write tests for exactly this issue. Indeed, as programmers, we are very familiar with ! to mean negation. Even if it looks like the ¦ (Broken Bar, \u00A6) which is in some CPs used as replacement of | (standard Pipe). It is very peculiar that 4GL uses ! as OR, and it is more also peculiar it uses the ^ as well (as it usually means EXCLUSIVE OR, with a different logic table). This is very confusing and unusual. Even the fact that all these 3 operators (!, | and &) for same goal.

The good part: the SQL side does support the negation and FWD can easily modified to accept such operator. So we could add an extension to standard 4GL here. The problem is to pick another character to represent the operation, since the usual ! is already taken. Maybe ¬ (not sign, rarely found on a keyboard) or ~?

#6 - 09/21/2026 07:46 PM - Greg Shah

We can set this to rejected, as long as we have checked in all the related testcases into our CONTAINS suite.

#7 - 09/22/2026 04:32 AM - Stefanel Pezamosca

Greg Shah wrote:

> We can set this to rejected, as long as we have checked in all the related testcases into our CONTAINS suite.

While working on 5219a I have generated a lot of testcases including these ones. So, it's already covered.

#8 - 09/22/2026 04:35 AM - Greg Shah

- % Done changed from 0 to 100

- Status changed from New to Rejected

Happy to reject this!