Base Language - Feature #4768

finish copy-lob support

07/15/2020 02:58 PM - Greg Shah

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Ovidiu Maxiniuc	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**version:**	
vendor_id:	GCD		

Description

Related issues:

Related to Base Language - Feature #2135: implement COPY-LOB language statement	**Closed**
Related to Base Language - Bug #6623: validate the memptr bytes in "copy-lob ...	**New**

History

#1 - 07/15/2020 03:01 PM - Greg Shah

The following need to be completed:

- TODOs in the code, primarily those where errors need to be handled with a more appropriate error code and message.
- Check undo behavior (if any) with test cases.
- Investigate flushing behavior (e.g., copy to a CLOB, then a FIND with a separate buffer for the same record does not pick up the change).
- Implement and fix various validations of source/target code page missing or different compared to 4GL.
- Determine correct code page for a database LOB field (currently defaults to longchar implementation for CLOB, -cpinternal for BLOB).
- Implement byte-order-marker (BOM) handling for files.
- Remove LobFile from version control (no longer used).
- Remove javadoc references to LargeObjectOps (no longer used).

#2 - 07/15/2020 03:07 PM - Greg Shah

- Related to Feature #2135: implement COPY-LOB language statement added

#3 - 11/07/2020 08:57 AM - Greg Shah

- Assignee set to Ovidiu Maxiniuc

#4 - 03/16/2021 06:46 PM - Ovidiu Maxiniuc

- Check undo behavior (if any) with test cases.

Actually the LOB fields are only allowed in NO-UNDO tables, the attempt to create an undoable table with a LOB field will result in a compile (or runtime for dynamic case) error 14435 (Large object fields can only be defined in NO-UNDO Temp-Tables).

I added the check at runtime but for static conversion I will add only a warning, to allow the conversion to continue, even this is not on par with 4GL.

A second, related issue. I know that, by default, we override the NO-UNDO / undoable attribute in recent code. I do not know how this will work with

code which already passed the validation.

#5 - 03/16/2021 07:15 PM - Eric Faulhaber

Ovidiu Maxiniuc wrote:

- Check undo behavior (if any) with test cases.

> Actually the LOB fields are only allowed in NO-UNDO tables, the attempt to create an undoable table with a LOB field will result in a compile (or runtime for dynamic case) error 14435 (Large object fields can only be defined in NO-UNDO Temp-Tables).
>
> I added the check at runtime but for static conversion I will add only a warning, to allow the conversion to continue, even this is not on par with 4GL.

This is sufficient, because we only need to convert valid 4GL code. If your warning is emitted, that will mean someone has fed invalid code into the conversion and the source code will need to be fixed.

> A second, related issue. I know that, by default, we override the NO-UNDO / undoable attribute in recent code. I do not know how this will work with code which already passed the validation.

The override is one direction only: UNDOable temp-tables can be forced to NO-UNDO, but not vice-versa. So, if a temp-table defined with a LOB field must already be NO-UNDO to be valid 4GL, we are OK.

#6 - 03/16/2021 11:51 PM - Eric Faulhaber

Ovidiu, please note we still need to ensure that UNDO on a persistent table with a LOB field works correctly.

#7 - 04/01/2021 09:14 AM - Greg Shah

Is this task complete as of 3821c rev 12222?

#8 - 04/01/2021 03:18 PM - Ovidiu Maxiniuc

- Status changed from New to WIP

- % Done changed from 0 to 90

Yes. It passed my testcases.

Then I used the xfer testcases for validation. Initially there were a lot of incorrect responses and the processing was stopping before finishing the full test set.
With rev 12222 there are only 3 kind of "UNEXPECTED_ERROR" reported:

- incorrect error number reported. This happens when the 4GL statement contains multiple invalid parameters and the order of validation is different. I tried switching around these but this did not fixed the issues for some cases. I chose the order with less errors reported. Examples: 11332 vs 11334 and 12008 vs 14500;
- a particular case of the above is 11316 vs 12008. Both of them seem generated by same cause. There is not much information on 11316. In fact, googling this error code will result in pages with error 12008;
- use of random method for generating test data. By the nature of this, the result is unpredictable. Attempting to text decode a randomized binary string to a character-type value will not always work as expected, at least because the random function will not generate the same series of numbers for 4GL and Java. And we cannot enforce this. As result, tests 76, 80, 98, 103, 119, 128 from Blob to Longchar and 60, 61, 67, 68 from Blob to Longchar [Overlay] will ... ~~randomly~~ occasionally fail.

Note:
There are some issues related to CODEPAGES. In FWD, we use Charset -s for text decoding. The problem is, these are more permissible than 4GL counterpart, meaning that they will return some String values even if the CODEPAGE used in 4GL will throw errors for same binary source. I tried to add pre-validation step but this is done only for ISO8859-1 CP for which I had enough information to understand the differences.

I updated the progress of this task to only 90% because of the issues related to CODEPAGE and the first item above.

#9 - 04/08/2021 12:12 AM - Ovidiu Maxiniuc

- % Done changed from 90 to 100

I found some edge-cases where the memptr behaves differently and adjusted the code accordingly.
The CLOB support was also improved.

Committed as revision 12257 of 3821c.

I updated the progress to 100% so that the task can be closed if the customer application has no issue with the change-set. The remaining issues related to I18N will probably be handled in related tasks (#4761).

#10 - 04/08/2021 06:53 AM - Greg Shah

- Status changed from WIP to Closed

#11 - 07/21/2022 07:22 AM - Greg Shah

- Related to Bug #6623: validate the memptr bytes in "copy-lob from memptr to clob/longchar" statement added