

Base Language - Feature #1627

support for empty EXPORT statements

10/21/2012 10:14 AM - Greg Shah

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Eugenie Lyzenko	% Done:	100%
Category:		Estimated time:	8.00 hours
Target version:	Conversion Support for Server Features	vendor_id:	GCD
billable:	No		
Description			

History

#1 - 10/31/2012 01:41 PM - Greg Shah

- Target version set to Milestone 7

#2 - 02/18/2013 09:59 AM - Greg Shah

- Estimated time changed from 4.00 to 8.00

- Assignee set to Eugenie Lyzenko

- Target version changed from Milestone 7 to Milestone 4

It is possible to have an "empty" EXPORT statement in the 4GL, as in:

```
EXPORT.
```

Normally, it would look like this (a list of expressions OR a database table follows the keyword):

```
EXPORT var1 (some-var2 + 54) var3.  
EXPORT database-table.
```

There are other options too, but these are the most common forms.

I have not tested it, but I suspect that this is also possible:

```
EXPORT STREAM my-stream.  
EXPORT DELIMITER "|".  
EXPORT NO-LOBS.
```

If all of the above "empty" export statements are accepted, then this also may be accepted:

```
EXPORT STREAM my-stream DELIMITER "|" NO-LOBS.
```

What you should do:

1. Create testcases to determine which of the empty export statements are possible.
2. Test to see if the usage of any of these statements ever changes the output file. If not, then these would be considered a NOP.
3. Based on the findings above, we would want to detect the empty export statement cases and if they are NOPs we would want to hide them in the

tree. This is probably best done during post_parse_fixups.

#3 - 02/18/2013 02:55 PM - Eugenie Lyzenko

- File export_tests.20130218a.zip added

The statements

```
EXPORT var1 (some-var2 + 54) var3.
```

```
EXPORT database-table.
```

both have run-time error in 4GL:

...

```
EXPORT cannot be used with output to the screen. (1105)
```

...

The statements has the following results:

```
EXPORT STREAM my-stream. - export_test1.p - generates 0 output file, no errors
```

```
EXPORT DELIMITER "|". - export_test2.p - generates 0 output file, no errors
```

```
EXPORT NO-LOBS. - export_test3.p - generates 0 output file, no errors
```

```
EXPORT STREAM my-stream DELIMITER "|" NO-LOBS. - export_test4.p - generates 0 output file, no errors
```

The testcases uses has been uploaded here.

#4 - 02/18/2013 04:15 PM - Greg Shah

```
EXPORT cannot be used with output to the screen. (1105)
```

When a stream is not explicitly used (e.g. EXPORT STREAM my-stream...) then you must first redirect the terminal like this:

```
OUTPUT TO my_file.txt.
```

```
EXPORT var1.
```

```
OUTPUT CLOSE.
```

Then there will not be any errors.

```
generates 0 output file
```

Do you mean they generate a 0 byte sized output file or that no output file is ever created?

#5 - 02/18/2013 04:29 PM - Eugenie Lyzenko

When a stream is not explicitly used (e.g. `EXPORT STREAM my-stream...`) then you must first redirect the terminal like this:

```
OUTPUT TO my_file.txt.
```

```
EXPORT var1.
```

```
OUTPUT CLOSE.
```

Then there will not be any errors.

This redirects output to the file and produces non-empty file with real values of the variable like printing to the file. This is not NOP code.

Do you mean they generate a 0 byte sized output file or that no output file is ever created?

New file generation with 0 bytes in size. Or overwrites the file with the same name with new one with 0 bytes inside. This means the code like this can erase the previously written data (if stream is opening not in append mode) and can be dangerous in some cases if some important data will be overwritten.

#6 - 02/18/2013 05:20 PM - Greg Shah

This redirects output to the file and produces non-empty file with real values of the variable like printing to the file. This is not NOP code.

Yes, exactly. That is a NON-empty export statement. It is the common case.

New file generation with 0 bytes in size. Or overwrites the file with the same name with new one with 0 bytes inside. This means the code like this can erase the previously written data (if stream is opening not in append mode) and can be dangerous in some cases if some important data will be overwritten.

Let me make sure I understand. In `export_test[1-3].p`, if you remove the empty `EXPORT` statement from those files, do you still get the 0 byte output file?

#7 - 02/18/2013 05:35 PM - Eugenie Lyzenko

Let me make sure I understand. In export_test[1-3].p, if you remove the empty EXPORT statement from those files, do you still get the 0 byte output file?

Yes, all these 4 cases (1-4) and appropriate 4GL statements gives the same result with and without export statement line - 0 byte output file. So the export does nothing itself in these cases.

#8 - 02/18/2013 06:39 PM - Greg Shah

Good. Then the solution is to do the following:

1. In post_parse_fixups.rules, you should add a new rule-set like this:

```
<rule-set name="fixups/hide_empty_exports" />
```

Add it near the end of the file (just after the "fixups/unspecified_undo").

2. In hide_empty_exports.rules, you must write rules to detect when we have an empty export statement. When you find one, you will hide it at the STATEMENT node. You can probably do something like this:

```
<!-- detect an empty export statement (there are no expressions and there is no
      DB record reference -->
<rule>
  relativePath("STATEMENT/KW_EXPORT") and
  !descendant(prog.export_field, 1) and
  !descendant(prog.table, 1) and
  !descendant(prog.buffer, 1) and
  !descendant(prog.temp_table, 1) and
  !descendant(prog.work_table, 1)

  <!-- hide the statement -->
  <action>copy.parent.setHidden(true)</action>
</rule>
```

#9 - 02/24/2013 01:43 PM - Eugenie Lyzenko

- File *ExportTestConverted20130224.zip* added

- File *evl_upd20130224a.zip* added

The modified *post_parse_fixups.xml* and new rule have been uploaded to eliminate empty export. Also the converted code to see the effect for samples has been uploaded.

#10 - 02/24/2013 02:33 PM - Greg Shah

The code looks good. I will run conversion testing now.

#11 - 02/24/2013 03:32 PM - Greg Shah

This has passed conversion testing. Go ahead and check it in and distribute it.

#12 - 02/24/2013 03:43 PM - Eugenie Lyzenko

The *evl_upd20130224a.zip* has been committed in bzz as 10197.

#13 - 02/24/2013 03:52 PM - Greg Shah

- Status changed from *New* to *Closed*

#14 - 02/26/2013 06:05 PM - Eric Faulhaber

- % Done changed from *0* to *100*

#15 - 11/16/2016 11:07 AM - Greg Shah

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

Files

<i>export_tests.20130218a.zip</i>	1.3 KB	02/18/2013	Eugenie Lyzenko
<i>evl_upd20130224a.zip</i>	5.29 KB	02/24/2013	Eugenie Lyzenko
<i>ExportTestConverted20130224.zip</i>	1.9 KB	02/24/2013	Eugenie Lyzenko