Base Language - Feature #2082

add conversion support for 4GL's duplicate variable "feature"

03/06/2013 08:18 AM - Constantin Asofiei

Status:	Closed	**Start date:**	03/06/2013
Priority:	Normal	**Due date:**	
Assignee:	Constantin Asofiei	**% Done:**	80%
Category:		**Estimated time:**	0.00 hour
Target version:	Conversion Support for Server Features		
billable:	No	**version:**	
vendor_id:	GCD		

Description

History

#1 - 03/06/2013 08:19 AM - Constantin Asofiei

4GL allows two vars with the same name, if this construct is encountered:

```
def new global shared var ch-1 like book.book-title. /* this is used */
def shared var ch-1 as int format "x(10)". /* this is ignored */
```

This is possible only if the first definition is from a "DEFINE NEW GLOBAL SHARED" statement (and only this). All subsequent definitions for var with the same name (using DEFINE SHARED OR DEFINED VAR or anything else), are ignored.

#2 - 03/06/2013 08:42 AM - Constantin Asofiei

- File ca_upd20130306c.zip added

- % Done changed from 0 to 80

- Status changed from WIP to Review

Attached update hides all duplicate var defs, only if the first one is DEF NEW GLOBAL SHARED. I'm putting this through conversion regression testing now.

#3 - 03/06/2013 08:58 AM - Greg Shah

The code looks good. I only have 2 questions:

1. Is there any difference in behavior when the non-global define variable is inside a nested scope (e.g. an internal procedure). In such cases, we would expect the global var to be temporarily hidden but the local var and would not think of that as a duplicate.

2. What happens in the case where the global var is created after the non-global var?

#4 - 03/06/2013 09:03 AM - Constantin Asofiei

> 1. Is there any difference in behavior when the non-global define variable is inside a nested scope (e.g. an internal procedure). In such cases, we would expect the global var to be temporarily hidden but the local var and would not think of that as a duplicate.

Yes, you are correct, when inside a procedure/function, it should hide the global var.

> 2. What happens in the case where the global var is created after the non-global var?

Compile-time error.

#5 - 03/06/2013 09:05 AM - Greg Shah

Please put a comment into the variable_definitions.rules about the order of the definitions and that we deliberately ignore that case since it fails in the 4GL with a compile error. That way, future readers will know that it was intentional.

#6 - 03/06/2013 09:10 AM - Constantin Asofiei

- File ca_upd20130306d.zip added

Added comment and fixed the proc/function case. I'm putting it back into conversion regression testing.

#7 - 03/06/2013 10:33 AM - Constantin Asofiei

The update has passed conversion regression testing. Note that even MAJIC had two cases of duplicated vars, in two files, which looked like this:

```
def new global shared var ch-1 as char. /* this is used */
def new shared var ch-1 as char. /* this must be ignored */
```

This didn't cause compile problems because one was an instance field for the class, and the other was emitted as an instance field for the Block instance associated with the external program call.

#8 - 03/06/2013 10:54 AM - Greg Shah

Check it in and distribute it.

#9 - 03/06/2013 10:59 AM - Constantin Asofiei

Committed to bzr revision 10255.

#10 - 03/06/2013 11:10 AM - Greg Shah

- Status changed from Review to Closed

#11 - 03/07/2013 05:13 AM - Constantin Asofiei

- File ca_upd20130307b.zip added

Fixed multiple calls like:

```
def new global shared var ch as char.
def new global shared var ch as char.
```

This is going through conversion regression testing.

#12 - 03/07/2013 05:49 AM - Constantin Asofiei

I need to rewrite this at the parser side, as the duped var gets ref'ed by the parser:

```
def new global shared var i as int. /* 1st def */

def shared var i as int. /* 2nd def */

procedure proc0.
   message i. /* the parser links this with the 2nd def. as 2nd def is hidden, it will no longer emit (as prom
oted) and 1st def will not be promoted, becaused is not ref'ed*/
end.
```

#13 - 03/07/2013 06:27 AM - Constantin Asofiei

- File ca_upd20130307c.zip added

So the problem was that the parser was adding tempidx to the dupped vars too. The sollution was to not register the dupped vars with the symbol dictionary, and for the dupped vars remove their tempidx during post-parse-fixups phase. This is going through conversion regression testing.

#14 - 03/07/2013 08:45 AM - Constantin Asofiei

This has regressions, one of them relaed to this, as 4GL (v9-only I think) allowed this:

```
def var ch as char label "adsadasd" as log init no.

if ch then message ch. /* ch is not char, but logical */
```

In v10, the code above does not compile.

#15 - 03/07/2013 08:59 AM - Constantin Asofiei

Fixed a problem (I was using varDict.getSymbol instead of varDict.getSymbolAtScope(name, 0) to search only in the current scope). Still working on the issue at note 14.

#16 - 03/07/2013 09:51 AM - Constantin Asofiei

MAJIC has the case at note 14 in only one file. I've checked the history of this file and this change was introduced after they switched to P2J server. So, the conclusion is that this is a bug in the 4GL sources, not another 4GL stupid feature.

I'll edit the file and put it in a separate task.

#17 - 03/07/2013 12:13 PM - Constantin Asofiei

- File ca_upd20130307d.zip added

The edited file is in task #2085.

This update has passed conversion regression testing (there are changes in two files, a var was promoted to instance field from instead of anon class instance field).

#18 - 03/07/2013 12:29 PM - Greg Shah

I'm not sure why you switched from a Set to a Map in fixups/variable_definitions.rules (did you plan to use it and then later on, find you didn't need it)?

Otherwise, I don't see a problem. Go ahead and check it in and distribute it.

#19 - 03/07/2013 12:42 PM - Constantin Asofiei

- File ca_upd20130307g.zip added

Greg Shah wrote:

> I'm not sure why you switched from a Set to a Map in fixups/variable_definitions.rules (did you plan to use it and then later on, find you didn't need it)?

Yes, I wanted to save the first node for the first var definition, which didn't help in the end. I've switched back to Set now.

Attached update was committed as revision 10263.

#20 - 11/16/2016 11:06 AM - Greg Shah

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

Files

ca_upd20130306c.zip	5.3 KB	03/06/2013	Constantin Asofiei
ca_upd20130306d.zip	5.42 KB	03/06/2013	Constantin Asofiei
ca_upd20130307b.zip	1.1 KB	03/07/2013	Constantin Asofiei
ca_upd20130307c.zip	39.9 KB	03/07/2013	Constantin Asofiei
ca_upd20130307d.zip	302 KB	03/07/2013	Constantin Asofiei
ca_upd20130307g.zip	305 KB	03/07/2013	Constantin Asofiei