

CoCoA-5 - Bug #726

TopLevel cannot "import" a package variable

06 Jun 2015 22:34 - John Abbott

Status:	Closed	Start date:	06 Jun 2015
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	enhancing/improving	Estimated time:	7.00 hours
Target version:	CoCoA-5.1.2 summer 2015	Spent time:	1.95 hour
Description			
I would like to have a user controllable "verbosity level" in a package. How should this be done?			
I had thought of having a package "global" variable \$.verbosity but then how can this be tested inside package functions?			
TopLevel verbosity; does not work (perhaps because the variable is not exported?)			
TopLevel \$.verbosity; gives a syntax error at \$ saying it was expecting an identifier.			
TopLevel \$PkgName.verbosity; gives the same syntax error.			
Related issues:			
Related to CoCoA-5 - Support #890: ImportByRef and ImportByValue behave in an...		Closed	15 Jun 2016

History

#1 - 06 Jun 2015 22:39 - John Abbott

I think a "global variable" is a reasonable design for this task.

There is a "philosophical" question: is a qualified package variable effectively a top-level variable? It can certainly be used like one in some contexts (e.g. a top-level assignment)

I imagine the parser will have to be modified; if the interpreter has been implemented cleanly, maybe no changes will be needed there :-)

#2 - 27 Jul 2015 11:55 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

JAA thinks the relevant code is Parser.C:837 -- probably TT_IDENTIFIER should be changed into something else. JAA will investigate.

#3 - 27 Jul 2015 12:16 - John Abbott

The problem can be worked around by following this example:

```
package $jaa;

export SetFlag;
export ClearFlag;
export GetFlag;

define SetFlag()
  TopLevel jaa_flag;
  jaa_flag := true;
enddefine; -- SetFlag

define ClearFlag()
  TopLevel jaa_flag;
```

```

jaa_flag := false;
enddefine; -- ClearFlag

define GetFlag()
  TopLevel jaa_flag;
  if not (IsDefined(jaa_flag)) then ClearFlag() endif;
  return jaa_flag;
enddefine; -- GetFlag

endpackage

```

This uses a TopLevel variable (called `jaa_flag`) instead of a package variable; pretty ugly!

#4 - 27 Jul 2015 13:47 - John Abbott

There is apparently no need to "import" package variables; it suffices just to use the fully qualified name, for instance

```

define IncrCounter()
  $MyPkg.counter := $MyPkg.counter + 1;
enddefine;

```

This is a sufficient solution for the problem I have in mind (`ArithGroup.cpkg5`). Is it the right solution?

The documentation for `TopLevel` and `ImportByValue/ImportByRef` should mention this.

#5 - 27 Jul 2015 14:46 - John Abbott

- % Done changed from 10 to 50

JAA now thinks that current design is OK: namely that package variables are always accessible using their fully qualified names. The only time the name may be used not fully qualified is at "top level" inside its native package (e.g. for setting the initial value).

Since package variables are actually globals, their use should not be encouraged by too much convenience.

I have added two small comments to the CoCoA-5 manual entries for `TopLevel` and `ImportByRef/ImportByValue`; these comment simply state that a package variable is always accessible via its fully qualified name (and that `TopLevel` and `ImportXYZ` do not work with package variables).

#6 - 27 Jul 2015 15:06 - John Abbott

- *Status changed from In Progress to Closed*
- *Assignee set to John Abbott*
- *Target version changed from CoCoA-5.?.? to CoCoA-5.1.2 summer 2015*
- *% Done changed from 50 to 100*

Answer: package variables are accessed via the fully qualified name.

#7 - 15 Jun 2016 14:07 - John Abbott

- *Related to Support #890: ImportByRef and ImportByValue behave in an unexpected manner (i.e. fail when I think they should succeed) added*