

Improved Dolt

(work by Denis Kudriashov)

- No AST transformation**
- No pretty printing**
- No method header**

- Shows the power of First Class Variables**

Pharo10

The screenshot displays the Pharo10 IDE interface. On the left, a code editor window shows the following code:

```
1 | temp |  
2  
3   temp := 2.  
4  
5 temp halt
```

In the center, a stack browser window titled "Halt" shows the current stack frame:

Class	Method
UndefinedObject	DoItIn:
OpalCompiler	[receiver withArgs: (context ifNil: [#()] ifNotI
	on:do:
	evaluate

Below the stack browser, the current object is identified as "an UndefinedObject (nil)". A variable table is also visible:

Type	Variable
implicit	self
temp. var	temp
implicit	stackTop
implicit	thisContext
implicit	Exception

On the right, a debugger window titled "Halt" shows the execution context:

```
1 DoItIn: ThisContext  
2  
3 ^ (ThisContext readVariableNamed: 'temp') halt
```

Pharol1

The image shows a screenshot of the Pharol1 debugger interface. It features a 'Playground' window on the left with the following code:

```
1 | temp |  
2 |  
3 |   temp := 2.  
4 |  
5 | temp halt
```

Below the code is a status bar showing 'Line: 1:1'. To the right of the code is a 'Stack' window with a toolbar containing 'Proceed', 'Into', 'Over', 'Through', and 'Run to'. The stack contains the following elements:

- Class
- UndefinedObject
- OCReceiverDoltSemanticScope (OCReceiverDoltSemanticScope)
- OpalCompiler
- SpCodePresenter

Overlaid on the right is another 'Halt' window, also with a 'Stack' window. This stack contains:

- Class
- UndefinedObject
- OCContextualDoltSemanticScope (OCContextualDoltSemanticScope evaluateDolt:)
- OpalCompiler
- SpCodePresenter
- FullBlockClosure (BlockClosure)

The 'UndefinedObject' in this stack is highlighted. Below the stack is a toolbar with 'Proceed', 'Into', 'Over', 'Through', 'Run to', 'Restart', 'Return', 'Where is?', 'Create', and 'Advanced Step'. At the bottom of this window, a code editor shows:

```
1 | temp halt
```