

## Base Language - Bug #2105

### BlockManager does not return some datatypes correctly

03/29/2013 08:55 AM - Ovidiu Maxiniuc

<b>Status:</b>	New	<b>Start date:</b>	03/29/2013
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	8.00 hours
<b>Target version:</b>			
<b>billable:</b>	No	<b>case_num:</b>	
<b>vendor_id:</b>	GCD	<b>version:</b>	
<b>Description</b>			
<b>Related issues:</b>			
Related to Base Language - Feature #1920: implement persistent procedures			<b>Closed</b>

### History

#### #1 - 03/29/2013 09:04 AM - Ovidiu Maxiniuc

Consider the following 4GL code:

```
FUNCTION get-long64 RETURNS INT64:  
    RETURN 1000.  
END FUNCTION.  
DEFINE VARIABLE i64 AS INT64.  
i64 = DYNAMIC-FUNCTION("get-long64").
```

Even though the code converts and compile OK, at runtime will crash with more exception ending with:

```
Caused by: java.lang.ArrayStoreException: com.goldencode.p2j.util.integer  
    at com.goldencode.p2j.util.BlockManager.functionBlock(BlockManager.java:7460)  
    at com.goldencode.p2j.util.BlockManager.function(BlockManager.java:274)  
    at com.goldencode.p2j.util.BlockManager.function(BlockManager.java:254)  
    at com.goldencode.testcases.util.X8int64bf.getLong64(X8int64bf.java:213)  
    ...
```

This is because the returnNormal from converted function asserts that the result (1000) is an integer value and retVal[i].duplicate() fails to be assigned to ret[i] which is int64 (the return type of the function).

## #2 - 03/29/2013 09:19 AM - Ovidiu Maxiniuc

A first attempt to fix would be to have only one `returnNormal` static function with a BDT parameter and the value wrapped at conversion time. However, this approach is not correct, as in 4GL, the result of calling a function statically differs from the result obtained by calling it dynamically. Here is a sample:

```
def var d1 as date.  
def var d2 as datetime.  
d1 = today.  
d2 = today.  
  
function f1 returns datetime. return d1. end.  
function f2 returns datetime. return d2. end.  
  
message dynamic-function("f1") f1().  
message dynamic-function("f2") f2().
```

where the output is something like:

```
29/03/13 29/03/2013 00:00:00.000  
29/03/2013 00:00:00.000 29/03/2013 00:00:00.000
```

The conclusion is that the dynamic call returns the actual object (which may not be of the exact return type of the function), while the static call always returns a value of the expected type. In P2J terms, this means:

- when calling statically, the result MUST be wrapped into the function's return datatype
- when calling dynamically, the result is the exact object' datatype that was used in the return statement.

## #3 - 04/12/2013 05:51 AM - Constantin Asofiei

- Estimated time set to 8.00

## #4 - 04/12/2013 06:04 AM - Constantin Asofiei

- Parent task deleted (#1920)