Base Language - Bug #2151

problems honoring errors raised when NO-ERROR clause is present

05/15/2013 12:01 PM - Constantin Asofiei

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#1 - 05/15/2013 12:06 PM - Constantin Asofiei

Ovidiu found a problem which is related to how DISPLAY manages errors during expression evaluation. In a case like:

```
def var i as int.
function j return datetime:
  message i.
  i = i + 1.
  return 2050-12-31t10:30:21.555.
end.

function k return datetime-tz:
  return 2050-12-31t10:30:21.555+10:00.
end.

display "datetime vs. datetime-tz:" 
        dynamic-function("j") lt dynamic-function("k") /* this generates error, if NO-ERROR clause is present 
*/
        dynamic-function("j") eq dynamic-function("k") /* this generates error, if NO-ERROR clause is present 
*/
        no-error.

def var l1 as log.
def var l2 as log.
assign l1 = dynamic-function("j") lt dynamic-function("k") /* this generates error, if NO-ERROR clause is pres
ent */
       l2 = dynamic-function("j") eq dynamic-function("k") /* this generates error, if NO-ERROR clause is pres
ent */
       no-error.
```

the following happens:
- in the DISPLAY, even if the first dynamic-function("j") eq dynamic-function("k") generates error, the second dynamic-function("j") eq dynamic-function("k") is executed too. If we use ErrorManager.recordOrShowError to generate the error, when building the THIS-PROCEDURE handle for the second dynamic-fuction("j") call in the DISPLAY case, the handle will not initialize as ErrorManager.isPendingError() is true, and the handle's ctor doesn't force the value to be assigned.
- in the second case, in 4GL the error is generated on the first l1 = dynamic-function("j") lt dynamic-function("k") assignment, and the second l2 = ... assignment will not be executed.

This hurts us because when building a handle instance for an existing WrappedResource instance, we check for the ErrorManager.isPendingError() flag. But, considering that for handles there is no INIT clause when defining the variable, we can assume that the handle(something) c'tors will always force the value to be set, and not check the ErrorManager.isPendingError() flag (i.e. they will be intended only for P2J internal usage).

All current usage of the handle c'tors (with 1 parameter) seem to accept this solution, except ArrayAssigner.assign{Multi|Single}(..., WrappedResource). In this case, we really need to rely on the ErrorManager.isPendingError() flag, because we need the assignment to fail if there is a pending error, for ASSIGN ... NO-ERROR cases.

But, there is a bigger problem with our ASSIGN ... NO-ERROR implementation, as we let the code execute even if an error is raised (but silently ignored) by a previous line. Although, currently I'm not sure if this is only related to the DYNAMIC-FUNCTION specific case above, or can be duplicated with some other code which generates an error.