Base Language - Bug #2511

timestamp instance oddity when assigning to a datetime

02/03/2015 12:11 PM - Ovidiu Maxiniuc

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

Description

Related issues:		
Related to Base Language - Bug #2499: DST portion of the timezone offset is a...	**WIP**	**01/23/2015**

History

#1 - 02/03/2015 12:45 PM - Ovidiu Maxiniuc

Consider the following P4GL code fragment executed on windev01, configured with timezone UTC+02:00 (only need to be other then UTC+06:00):

```
DEF VAR dt1 AS DATETIME.
DEF VAR dt2 AS DATETIME.
DEF VAR dtz AS DATETIME-TZ.

SESSION:TIMEZONE = +6 * 60.

dt1 = NOW. /* assign it directly */
dtz = NOW.
dt2 = dtz. /* assign it indirectly, via dtz */
DISPLAY
      "dt1" dt1 SKIP
      "dt2" dt2 SKIP
      "dtz" dtz WITH FRAME f1 NO-LABELS.
```

Both dt1 and dt2 were assigned with the current timestamp. The difference is that the first one is assigned directly using the NOW function, for the latter the assignment is indirect, using dtz as intermediary.
Normally, one would expect that since the SESSION:TIMEZONE is constant during these variable handlings, the output for dt1 and dt2 to be the same. However, the result is slightly different than one expects (at least I):

```
dt1 03/02/2015 23:14:27.417
dt2 03/02/2015 19:14:27.417
dtz 03/02/2015 23:14:27.417+06:00
```

The NOW instance gets back in time (+06:00 - +2:00 = 4h), ignoring the SESSION:TIMEZONE.

This wouldn't be such an issue. We could use the machine default timezone to do the calculations when doing dtz to dt assignments. But looking into the generated code, we can see something like this:

```
    dt1.assign(datetimetz.now());
    dtz.assign(datetimetz.now());
    dt2.assign(dtz);
```

The same method assign is used 3 times, the important calls being the ones with datetime object called on.
They will have the same execution path, so for P2J POV, it cannot distinct when to keep the offset configured in SESSION:TIMEZONE (1st case) and when to drop to the default one (3rd assignment).

#2 - 03/23/2016 05:19 PM - Greg Shah

- Target version changed from Milestone 12 to Milestone 16

#3 - 11/16/2016 12:23 PM - Greg Shah

- Target version changed from Milestone 16 to Cleanup and Stabilization for GUI