User Interface - Bug #12123

Spreadsheet widget issues Found While Writing the Topic 15 Demo

11/05/2026 12:37 PM - Sergey Ivanovskiy

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**reviewer:**	Hynek Cihlar
vendor_id:	GCD	**production:**	No
case_num:		**env_name:**	
version_reported:		**topics:**	
version_resolved:			

Description

History

#1 - 11/05/2026 12:38 PM - Sergey Ivanovskiy

- File spreadsheet_demo_issues.p.ext-hints added

- File spreadsheet_demo_issues.p added

- File spreadsheet_demo.p.ext-hints added

- File spreadsheet_demo.p added

SPREADSHEET Widget: Issues Found While Writing the Topic 15 Demo

Background

These issues showed up while writing the spreadsheet demo for the FWD User Meeting (#12087, topic 15): abl/spreadsheet_demo/spreadsheet_demo.p in the testcases project, which drives the SPREADSHEET widget through the unchanged Formula One API (chCtrlFrame:F1Book, mapped by spreadsheet_demo.p.ext-hints). The two issues found in the signature demo (conversion) are in signature_demo_issues.textile.

Tested on trunk rev 16821, web client, Chrome. The demo works around each issue; the workaround is given per issue. Line numbers refer to trunk rev 16821.

Every issue was then checked on OpenEdge 11.6 (32-bit prowin32) with the real Formula One OCX (VCF132.OCX, sheet loaded from the metacom wrx/wssheet.wrx), using the test program abl/spreadsheet_demo/spreadsheet_demo_issues.p (see #OpenEdge Check). The same program ran on FWD trunk (web client, see #Trunk Check). All seven issues are FWD defects. Issue 7 was found during the OpenEdge check.

Summary

#	Issue	Where in FWD	OpenEdge result	Workaround in the demo
1	Style calls apply the top-left cell's style to the whole range	SheetControllerImpl.setPattern, setFont (and the number format)	FWD defect: each cell keeps its format	Format only ranges whose cells share one style
2	SetAlignment never applies	SpreadsheetWidget.setAlignment, line 3142	FWD defect: right aligned	None (alignment left out)
3	EndEdit with Cancel = 1 damages the cell	SpreadsheetWidget onStopEditing, lines 3628-3651	FWD defect: the cell and the sums never change	No cancel: the old value is written back from SelChange
4	A changed EndEdit edit string is ignored	SpreadsheetWidget onStopEditing (output res[0] unused)	FWD defect: the cell gets the changed string	Same as 3
5	TopRow / LeftCol read	SheetControllerImpl.setTop	FWD defect (values);	SetActiveCell(1, 1) at the

	wrong values; SetActiveCell scrolls the cell to the top left	Row, setLeftColumn, setActiveCell	scrolling not reproducible on OE	end of the build
6	In a sheet event, reads do not see writes made earlier in the same handler	implicit batch (#11388)	FWD defect: the read sees the write	Do not read after writing in the event
7	SelChange is not raised for SetSelection / SetActiveCell from 4GL	sheet event dispatch	FWD defect: OpenEdge raises it for every call	None needed in the demo

1. Style Calls Apply the Top-Left Cell's Style to the Whole Range

Observed: SetFont, SetPattern and NumberFormat on a selected range take the style of the range's top-left cell, change one attribute and set the result on every cell of the range. Other attributes of the other cells are lost. In the demo, a bold font set on F4:G10 (totals and shares) gave the share column the #,##0 format of column F, so the shares 0.251 showed as 0 and 1. The highlight of a row (SetPattern on A:G) copied the format of column A to the numbers.

Where: ext/sheet/.../web/SheetControllerImpl.java: setPattern (line 4481) builds the new style from r.getCellStyle() of the selection range and sets it with r.setCellStyle(newStyle); setFont (line 5610) does the same with r.getCellStyle() and builder(style). Keikai returns the top-left cell's style for Range.getCellStyle(). The number format behaved the same way in the demo.

Reproduce:

```
{&F1}:NumberRC(1, 1) = 1234.
{&F1}:NumberRC(1, 2) = 0.25.
{&F1}:SetSelection(1, 1, 1, 1).
{&F1}:NumberFormat = "#,##0".
{&F1}:SetSelection(1, 2, 1, 2).
{&F1}:NumberFormat = "0.0%".
/* bold blue font on both cells */
{&F1}:SetSelection(1, 1, 1, 2).
{&F1}:SetFont("Arial", 10, TRUE, FALSE, FALSE, FALSE, RGB-VALUE(0, 0, 255), FALSE, FALSE).
```

FWD: B1 shows 0: FormattedTextRC [0], NumberFormat [#,##0] (A1's format).

OpenEdge: A1 1,234 and B1 25.0%, both bold blue. B1 keeps NumberFormat [0.0%]; only the font changes.

Workaround: format ranges whose cells all share one style (the demo's styleRange, styleDataRow).

2. SetAlignment Never Applies

Observed: SetAlignment(4, FALSE, 2, 0) (right) and SetAlignment(7, ...) (center across cells) change nothing.

Where: src/com/goldencode/p2j/ui/SpreadsheetWidget.java line 3142:

```
public void setAlignment(NumberType hAlign, logical wordWrap, NumberType vAlign, NumberType orientation)
{
   if (!BaseDataType.isAllKnown(hAlign, wordWrap, vAlign))
   {
      invokeWidgetCommand(SpreadsheetCommands.SET_ALIGN, ...
```

The condition is inverted: the command is sent only when an argument is unknown (and then intValue() of the unknown value is passed). With known arguments nothing happens. The fix is to drop the !.

Reproduce:

```
{&F1}:TextRC(1, 1) = "right".
{&F1}:SetSelection(1, 1, 1, 1).
{&F1}:SetAlignment(4, FALSE, 2, 0).
```

FWD: the text stays left aligned. **OpenEdge:** right aligned.

3. EndEdit with Cancel = 1 Damages the Cell

Observed: a numeric cell (33500, format #,##0) referenced by SUM formulas; the user types abc and Enter; the EndEdit handler sets p-Cancel = 1.

- While the editor is open again, the totals already exclude the cell: the edit was committed before the event.
- After Escape the cell shows 33500 left aligned: it holds the old value as **text**, so it stays out of the sums and the number format does not apply.

Where: SpreadsheetWidget.java onStopEditing, lines 3628-3651: on cancel, the client gets SET_CELL_VALUE with this.editingValue (the editor's start string) and START_EDIT. The cell is restored from a string, not from its old value and type, and only after Keikai has committed the edit.

Reproduce: sheet.F1Book.EndEdit that sets p-Cancel = 1; the number 33500 in B4, SUM(B4) in C4; type abc in B4, Enter, then Escape.

FWD: inside EndEdit the cell already holds the edit: B4 TypeRC 2, TextRC [abc], C4 0. The editor stays open with abc and C4 shows 0. After Escape, B4 is text [33500] (TypeRC 2, left aligned) and C4 stays 0.

OpenEdge: inside EndEdit the cell still holds the old number: B4 TypeRC 1, 33500; C4 33500. The editor stays open with abc, and C4 keeps 33,500. After Escape, B4 is the number 33,500 with its format (TypeRC 1), and C4 is 33,500. The edit is never applied.

The restore goes to the cell that has the focus when it runs (SheetControllerImpl.setCellValue writes to getFocusRange()), not to the edited cell. Enter moves the focus down, so depending on timing the old value lands in the cell below. In an earlier run it went to B4; in both runs of 2026-11-02 it went to B5: B4 kept the text [abc], B5 became the text [33500] and the editor reopened on B5.

Workaround: the demo does not cancel. EndEdit notes the rejected cell, and SelChange (which follows when the edit is committed) writes the old entry back with EntryRC.

4. A Changed EndEdit Edit String Is Ignored

Observed: EndEdit assigns p-EditString (for example the cell's previous content); the cell still gets what the user typed.

Where: SpreadsheetWidget.java onStopEditing: the completion consumer reads only res[1] (Cancel); the output edit string res[0] is never used.

Reproduce: EndEdit sets p-EditString = "999" for B5 (33500, SUM(B5) in C5); type abc in B5, Enter.

FWD: B5 is the text [abc] and C5 is 0. **OpenEdge:** B5 is the number 999 (format #,##0) and C5 is 999: the value the handler leaves in the edit string goes into the cell.

Workaround: as for 3.

5. TopRow / LeftCol Read Wrong Values; SetActiveCell Scrolls the Cell to the Top Left

Observed in the demo: after building a sheet (rows 1-10, columns A-G, all visible in the widget), SetActiveCell(4, 2) showed the sheet scrolled to B4 at the top left; rows 1-3 and column A were hidden. TopRow = 1 and LeftCol = 1 assigned after EndBatch did not scroll back. SetActiveCell(1, 1) restores the view.

Where: SheetControllerImpl.java: setTopRow (line 2059) and setLeftColumn return when spreadsheetCtrl.getVisibleArea() is null, and otherwise set a visible rectangle that the later focus handling can override; setActiveCell (line 4595) sets the client focus and selection, which the client then scrolls into the top-left position. Not analysed further.

Reproduce (demo):

```
{&F1}:StartBatch().
/* fill A1:G10 */
{&F1}:SetActiveCell(4, 2).
{&F1}:EndBatch().
{&F1}:TopRow  = 1.
{&F1}:LeftCol = 1.
```

Test program (no batch, since OpenEdge has none): SetActiveCell(4, 2), then read TopRow and LeftCol; then TopRow = 1, LeftCol = 1 and read them again.

Read	FWD	OpenEdge
after SetActiveCell(4, 2)	TopRow 0, LeftCol 0	TopRow 1, LeftCol 1
after TopRow = 1, LeftCol = 1	TopRow 1, LeftCol 0	TopRow 1, LeftCol 1
at the end, after user input	TopRow 1, LeftCol 1	TopRow 1, LeftCol 1

On both, A1 stays visible at the top left in this program, so the scrolling of the demo needs the batch to show; that part cannot be compared with OpenEdge. The values are an FWD defect: the getters return 0 until the client has reported its visible area, and LeftCol ignores the assignment.

6. In a Sheet Event, Reads Do Not See Writes Made Earlier in the Same Handler

Observed: in sheet.F1Book.SelChange, EntryRC(7, 3) = "33500" followed by NumberRC(10, 6) (a SUM over column C) returns the total from before the write. Recalc() between the two does not help. The sheet shows the right total afterwards. In a button trigger with explicit StartBatch() / EndBatch(), a read after EndBatch() is correct.

Where: likely the implicit batch (#11388): the first write in the event starts a batch that is applied when the handler returns, and a read in between does not end it.

Reproduce: B7 10, SUM(B7) in C7; in SelChange, when B8 becomes active, EntryRC(7, 2) = "20", then NumberRC(7, 3), then Recalc() and NumberRC(7, 3) again. The same write and read outside any event (B9, C9) is the control.

FWD: C7 reads 10 both times; the control reads 20. **OpenEdge:** C7 reads 20 both times; the control reads 20. Formula One is synchronous.

Workaround: the demo does not read after writing in the event (the totals after an undone edit are those from before the edit).

7. SelChange Is Not Raised for SetSelection / SetActiveCell from 4GL

Found during the OpenEdge check.

OpenEdge: every SetSelection and SetActiveCell called from 4GL raises SelChange at once, inside the call (the log shows SelChange B4, SelChange B5 while the sheet is built, and one SelChange per cell read by the report). A user edit accepted with Enter does not move the active cell.

FWD: SelChange is raised only for user actions. On the other hand, after the cancelled edit of issue 3 it is raised twice for B4, and after an accepted edit Enter moves the active cell down (SelChange B6 after the edit of B5).

Code that relies on SelChange for its own selection changes (or that does not expect it) behaves differently. The Enter movement may depend on a Formula One setting stored in wssheet.wrx; not analysed further.

OpenEdge Check

Test program: abl/spreadsheet_demo/spreadsheet_demo_issues.p (testcases project), with spreadsheet_demo_issues.p.ext-hints. One sheet, one case per issue; every value goes to spreadsheet_demo_issues.out in the working directory, so the FWD and OpenEdge runs compare line by line. Cases 1, 2, 5 and 6b run at start; cases 3, 4 and 6 are interactive (the header lists the keys), then **Report** writes the final state.

OpenEdge run: on the Windows2012 VM through remote-oe. The wrapper defines &GLOBAL-DEFINE OE-F1 yes and includes the test; OE-F1 names the control frame sheet and loads wrx/wssheet.wrx. The 32-bit install only runs r-code, so the wrapper is compiled with the 64-bit prowin.exe (COMPILE ... SAVE) and run with C:\Progress\OE116_32\bin\prowin32.exe.

FWD run: trunk, web client (headless Chrome through Playwright), same keys; see #Trunk Check.

Note for test authors: SetSelection (and reading NumberFormat, which needs it) inside EndEdit crashes prowin32 with Formula One ("OpenEdge Graphical Client has stopped working"). The EndEdit handler of the test only reads cells with TypeRC, TextRC and NumberRC. FWD does not crash; there is nothing to emulate.

Trunk Check

Build: trunk rev 16821 (/home/sbi/projects/p2j/build/lib, built 2026-10-31). The checkout is at rev 16826, but revisions 16822-16826 do not touch ext/sheet, SpreadsheetWidget or the sheet events (terminal reader, LOB, web services, pattern engine), so the build stands for the current trunk.

Run: testcases project, 2026-11-02. The trunk files (p2j.jar, fwdspi.jar, fwdaopltw.jar, fwd_sheet.war, libp2j.so) were copied into deploy/lib for the run and the previously deployed files put back afterwards; no symlink changed. This is enough for the web client too: the spawned client loads its jars from the server's deploy/lib (the spawner's own p2j.jar under ~/opt/testcases only starts the client process). Web client in headless Chrome, two runs. The first ran the keys from the program header; because issue 3 left the focus on B5, its Down put case 4 on B6. The second run clicked B5 and B8 instead of using Down.

#	Trunk result
1	B1 after SetFont A1:B1: FormattedTextRC [0], NumberFormat [#,##0] (A1's format); bold blue
2	A2 right stays left aligned
3	In EndEdit: B4 TypeRC 2 [abc], C4 0. After the cancel: B4 text [abc], B5 text [33500] (restore in the cell below, see issue 3), C4 0
4	B5 text [abc], C5 0: the edit string 999 is ignored
5	After SetActiveCell(4, 2): TopRow 0, LeftCol 0; after TopRow = 1, LeftCol = 1: TopRow 1, LeftCol 0; at the end 1, 1
6	In SelChange C7 reads 10, also after Recalc(); outside events (6b) C9 reads 20
7	No SelChange for the SetSelection / SetActiveCell calls of the build; SelChange B5 raised four times after the cancel of issue 3

#2 - 11/05/2026 12:40 PM - Sergey Ivanovskiy

- Subject changed from Issues Found While Writing the Topic 15 Demo to Spreadsheet widget issues Found While Writing the Topic 15 Demo

Files

spreadsheet_demo.p.ext-hints	234 Bytes	11/05/2026	Sergey Ivanovskiy
spreadsheet_demo.p	14.7 KB	11/05/2026	Sergey Ivanovskiy
spreadsheet_demo_issues.p.ext-hints	234 Bytes	11/05/2026	Sergey Ivanovskiy
spreadsheet_demo_issues.p	13.7 KB	11/05/2026	Sergey Ivanovskiy