User Interface - Feature #12032

implement the ITERATION-CHANGED event

10/20/2026 01:40 PM - Greg Shah

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

Description

History

#1 - 10/20/2026 01:40 PM - Greg Shah

Implement the ITERATION-CHANGED event

Project: User Interface

Problem

ITERATION-CHANGED is an old browse event which the 4GL still supports, and which VALUE-CHANGED replaced in the documentation. Applications written against older releases use it heavily to keep a detail pane in step with the focused row of a browse, in two forms:

```
ON ITERATION-CHANGED OF br_main DO:
   RUN refresh_detail.
END.

OPEN QUERY q_main FOR EACH ...
APPLY "ITERATION-CHANGED" TO br_main.
```

FWD converts both forms but does not know the event at runtime:

- It is not in the Keyboard event name table, so Keyboard.eventCode() returns -1.
- EventList.eventCode returns null, so the trigger is never registered.
- LogicalTerminal.apply(handle, String) logs "Could not recognize event" and returns.
- The browse never fires it.

So both the triggers and the explicit APPLY statements are silent no-ops, and every detail pane driven this way stops refreshing. The gap marking records it as Full / None.

Work

1. Establish on OpenEdge exactly when ITERATION-CHANGED fires on a browse:
   - when the user moves the focused row with the keyboard or the mouse;
   - after OPEN QUERY, REPOSITION, SELECT-ROW, SELECT-FOCUSED-ROW, REFRESH and INSERT-ROW;
   - on single- and multiple-selection browses, and on an empty browse;
   - its order relative to VALUE-CHANGED, ROW-ENTRY, ROW-LEAVE and ROW-DISPLAY, and whether both fire for the same change.
2. Add the event to the event tables so that ON ITERATION-CHANGED registers a trigger and APPLY "ITERATION-CHANGED" runs it.
3. Fire it from the browse at the points established above. Do not simply alias it to VALUE-CHANGED unless OpenEdge shows the two are identical in every case.
4. Check that it works for static and dynamic browses, and for a trigger defined with PERSISTENT RUN.
5. Update the gap marking.

Testcases

Each with an OpenEdge baseline that records which events fire, how many times and in what order:

- moving through a browse with the arrow keys, page keys and the mouse;
- the programmatic operations listed above;
- APPLY "ITERATION-CHANGED" to a browse with and without a trigger;
- single- and multiple-selection browses.