Base Language - Feature #12011

implement EXIT-CODE as the process exit status

10/15/2026 09:30 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 - 10/15/2026 09:30 PM - Greg Shah

SESSION:EXIT-CODE must become the exit status of the client process

Project: Base Language

Problem

In OpenEdge, a batch session can assign SESSION:EXIT-CODE and then QUIT. The value becomes the exit status of the _progres / prowin process. A script, a cron job or a scheduler uses it to tell success from failure.

FWD accepts the assignment and stores the value, but it never reaches the operating system:

- SessionUtils.setExitCode(long) and getExitCode() (util/SessionUtils.java:1500-1519) keep the value in the context's work area, around line 2470.
- No code reads it when the session ends. The client process always exits with the status of a normal shutdown, so every failure reported this way looks like success to its caller.
- Conversion is also incomplete: the setter has no NumberType overload. It works for an integer literal but not for every numeric expression.

The value lives in the server-side context, where the converted business logic runs. The OS process whose status matters is the FWD client, or the process the spawner launched. So the value has to travel from the server to the client when the session ends.

Work

1. **Conversion.** Add the NumberType overload of the setter, and make sure every numeric form converts: INTEGER, INT64, DECIMAL and expressions.
2. **Transport.** When the session ends through QUIT, the normal end of the startup procedure, or the end of a batch client, send the stored exit code to the client as part of the shutdown. Check every client type that can run a batch session: the batch/ChUI client, a spawned batch process, and the headless client used by schedulers.
3. **Process exit.** In the client, exit the JVM with the received value. Also check any native launcher or script that wraps the JVM, so it passes the status through rather than replacing it with its own.
4. **OpenEdge semantics.** Establish them with testcases and match them:
   - the status when EXIT-CODE was never assigned;
   - the status after an unhandled ERROR or STOP ends the session, with and without an assigned EXIT-CODE;
   - an unknown value, a negative value and a value above 255, including any truncation the operating system applies;
   - QUIT versus the normal end of the startup procedure;
   - an assignment followed by further processing that fails before the session ends.
5. **Interactive clients.** Decide what a GUI or web client does with the value. In OpenEdge it is the exit status of prowin. A web client has no process status to set.
6. Update the gap marking for the attribute once both halves are done.

Testcases

- A batch program that sets 0, 2 and 255 and then QUIT@s, run from a shell that checks $? (and ERRORLEVEL@ on Windows).

- The unknown, negative and out-of-range cases listed above.
- A session ended by an unhandled error, with and without an assigned value.