Base Language - Bug #11894

SECURITY-POLICY:GET-CLIENT returns the live client-principal instead of a copy

09/19/2026 10:23 AM - Greg Shah

Status:	Review	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Eduard Soltan	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**reviewer:**	Constantin Asofiei
vendor_id:	GCD	**production:**	No
case_num:		**env_name:**	
version_reported:		**topics:**	
version_resolved:			

Description

History

#1 - 09/19/2026 10:24 AM - Greg Shah

SECURITY-POLICY:GET-CLIENT returns the live client-principal instead of a copy

Summary

SECURITY-POLICY:GET-CLIENT() is documented, by the 4GL and by FWD's own javadoc, to return **a copy** of the sealed client-principal. FWD returns the stored object itself. A caller which modifies what it gets back modifies the session's security identity.

The mirror-image problem exists on the way in: SET-CLIENT stores the caller's handle directly rather than copying it, so the caller keeps a reference to the object the session is now using.

Severity

Functional, silent, and security-adjacent. There is no error and no warning; the wrong result is simply that two things which should be independent are the same object.

The practical exposure depends on whether application code modifies a client-principal after retrieving it. A sealed client-principal is intended to be immutable to the application, so a well behaved program will not notice — but nothing in FWD enforces that, and the 4GL contract exists precisely so that it does not have to.

Evidence

src/com/goldencode/p2j/util/SecurityPolicyManager.java:380-392 — the javadoc states the contract and the code does not honour it:

```
/**
 * Implementation of SECURITY-POLICY:GET-CLIENT method. Returns the handle to a
 * copy of the sealed client-principal object that represents the user identity
 * for the ABL session. If no identity has been established for the session
 * using the SECURITY-POLICY:SET-CLIENT( ) method, this method returns the
 * Unknown value (?).
 *
 * @return The handle to a copy of the sealed client-principal object.
 */
public static handle getClient()
{
   return work.get().client;
}
```

work.get().client is the context-local work area field. No copy is made.

The corresponding store, SecurityPolicyManager.java:556, in the SET_CLIENT path:

```
wa.client = h;
```

h is the handle the caller passed in. It is retained directly.

What Correct Behaviour Looks Like

Per the 4GL, GET-CLIENT hands back a copy so that the caller cannot reach into the session identity, and SET-CLIENT takes a copy so that the caller cannot mutate it afterwards. Either half alone leaves a hole:

- Fixing only GET-CLIENT still lets whoever called SET-CLIENT retain a live reference.
- Fixing only SET-CLIENT still hands a live reference to every later GET-CLIENT caller.

Suggested Fix

Copy the client-principal on both boundaries. The copy has to preserve the seal and the validation state, so it should go through whatever the client-principal implementation already provides for duplication rather than a field-by-field clone written at this call site.

Worth checking while fixing:

- whether any FWD-internal caller of getClient() **relies** on getting the live object — a copy would change behaviour for those, and they would need to read the work area field directly instead
- whether SecurityOps.setDbClient, called just above the assignment at :554, should receive the original or the copy

How It Was Found

While resolving conflicting gap markings for SECURITY-POLICY:GET-CLIENT (K6 in #11888-24). Constantin Asofiei flagged the copy semantics in #11888-317977:

> security-manager:get-client - there is a bug (SecurityPolicyManager$WorkArea.client gets assigned or returned directly, and not a copy), but otherwise is FULL

This report records the bug itself. The marking question it came from is written up separately in #11888-24 K6.

Not Established

- Whether OpenEdge's copy is deep or shallow, and what exactly "a copy of the sealed client-principal" includes. That determines what the fix has to duplicate.
- Whether any of the project code path modifies a retrieved client-principal. The gap analysis records security-manager:get-client usage but the report does not say what callers do with the result.

Sources

- src/com/goldencode/p2j/util/SecurityPolicyManager.java, FWD branch 11747a
- #11888-317977 — Constantin Asofiei
- #11888-24 K6 — the gap marking side of the same finding

#3 - 09/21/2026 02:16 AM - Eduard Soltan

- Assignee set to Eduard Soltan

- Status changed from New to WIP

#4 - 09/21/2026 05:29 AM - Eduard Soltan

- % Done changed from 0 to 100

- Status changed from WIP to Review

- reviewer Constantin Asofiei added

GET-CLIENT returned the stored client-principal instead of a copy, and SET-CLIENT stored the caller's handle instead of a copy. Fixed on branch **11894a rev. 16765** in src/com/goldencode/p2j/util/SecurityPolicyManager.java - both boundaries now copy via ClientPrincipal.clone().

Not hypothetical: DELETE OBJECT on a retrieved client-principal wiped the session identity, because ClientPrincipal.delete() clears every attribute. FWD's own TestGetClient.cls does exactly that, so the bug was reachable with no project code involved.

- two GET-CLIENT calls return distinct objects, neither one being the one passed to SET-CLIENT method.
- the copy carries USER-ID, DOMAIN-NAME, SESSION-ID, LOGIN-STATE and the properties map
- deleting either side leaves the other fully functional, properties included - so the copy is **deep**.