Base Language - Feature #12010

crypto feature compatibility

10/15/2026 09:26 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 09:27 PM - Greg Shah

MESSAGE-DIGEST HMAC support, and ENCRYPT/DECRYPT compatibility fixes

Project: Base Language
Related: #8778 (MD5-DIGEST result), #6399 (ENCRYPT/DECRYPT IV length)

Problem

Several defects in util/SecurityOps.java affect hashing and encryption. They are grouped here because they share the same code and the same kind of testing: comparison against values OpenEdge produced.

1. MESSAGE-DIGEST does not support HMAC

Code that signs requests for common cloud APIs (for example AWS Signature Version 4) calls:

```
rKey = message-digest("HMAC-SHA-256", cData, cKey).
rKey = message-digest("HMAC-SHA-256", cData, rKey).   /* key as RAW */
```

FWD has two problems with this:

- SecurityOps.algorithm() (near line 3668) accepts only MD5, SHA-1, SHA-256 and SHA-512. Any other name raises "Invalid message digest algorithm name." That message and its error number (-1) are FWD's own, and a TODO there asks what the 4GL actually does.
- The keyed path (messageDigest(String, byte[], byte[]), near line 3628) hashes the data with the key appended. It does not compute an RFC 2104 HMAC.

So a signature computed this way fails, and every request signed with it is rejected.

2. MESSAGE-DIGEST hashes text as UTF-8

CHARACTER and LONGCHAR input is always converted to UTF-8 before hashing (near line 3695). The 4GL hashes the bytes in the session's codepage, or in the LONGCHAR's fixed codepage. Text that is not ASCII therefore produces a different digest. SHA1-DIGEST and MD5-DIGEST share the helper and should be checked too.

3. ENCRYPT and DECRYPT do not fall back to SECURITY-POLICY when the key or IV is unknown

The one- and four-argument forms use SECURITY-POLICY:SYMMETRIC-ENCRYPTION-KEY and SYMMETRIC-ENCRYPTION-IV when the argument is the unknown value. The following overloads do not:

- encrypt(BinaryData, raw key) (near line 2070) and encrypt(BinaryData, raw key, raw iv) (near line 2093);
- the matching decrypt overloads (near lines 2164 and 2185).

The three-argument forms also ignore an unknown IV. The String and Text overloads delegate to these, so they inherit the defect.

Work

1. **Establish the OpenEdge 12.8 behaviour of MESSAGE-DIGEST** with testcases:
   - the full list of algorithm names accepted, including every HMAC-* form and CRC-32 if documented;
   - what the third argument does for a plain algorithm (appended, prepended or HMAC), and for an HMAC-* algorithm;
   - the key as CHARACTER, RAW and MEMPTR;
   - the error number and message for an invalid algorithm name.
2. **Implement the HMAC algorithms** with javax.crypto.Mac. Verify them against the RFC 4231 test vectors and against OpenEdge output.
3. **Correct the keyed form of the plain algorithms** if OpenEdge differs from FWD's current "append the key" behaviour.
4. **Hash text in the right codepage:** the session codepage for CHARACTER, and the fixed codepage for LONGCHAR. Do the same in SHA1-DIGEST and MD5-DIGEST if they share the helper.
5. **Replace the FWD-specific error** for an invalid algorithm with the 4GL error.
6. **Add the SECURITY-POLICY fallback** for an unknown key and IV to the two- and three-argument ENCRYPT / DECRYPT overloads.
7. **Add compatibility tests for DECRYPT of data OpenEdge encrypted** and stored, such as a database field written by a 4GL program and read back by the converted one. Cover at least:
   - AES_OFB_128 with a key from GENERATE-PBE-KEY and an unknown IV;
   - AES_CBC_128 with a salt and PBE-KEY-ROUNDS 1000;
   - the session default algorithm.
8. Correct the gap marking for MESSAGE-DIGEST, SHA1-DIGEST, ENCRYPT and DECRYPT once these are done.