Base Language - Feature #12013

X-DOCUMENT:LOAD encoding and error handling compatibility

10/15/2026 09:33 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 09:33 PM - Greg Shah

X-DOCUMENT:LOAD encoding and error handling compatibility

Project: Base Language

Problem

XDocumentImpl.load() (xml/XDocumentImpl.java, near line 754) supports the FILE, MEMPTR, LONGCHAR and URL source types, but its own comments record that it is not complete:

- the module header calls load() "pretty badly broken";
- a TODO near line 756 notes that the error handling is not shared with the other X-DOCUMENT methods;
- a TODO near line 940 notes that "the I18N implications" are "probably WRONG!"

The concrete defect is in the LONGCHAR path. It converts the content to bytes with the JVM's **platform default** character set, then decodes those bytes with the encoding the XML declaration names. When the platform default differs from the declared encoding, or from the LONGCHAR's own codepage, non-ASCII characters are damaged.

Work

1. **LONGCHAR source.** Parse from the characters directly, without the byte round trip. Where the 4GL needs bytes, encode in the LONGCHAR's fixed codepage. Establish with testcases what OpenEdge does when the LONGCHAR's codepage and the XML declaration disagree, and match it.
2. **MEMPTR and FILE sources.** Detect the encoding the way the 4GL does: byte order mark, then the XML declaration, then the default. Check a document with no declaration in a non-UTF-8 codepage.
3. **Errors.** Use one error path for an invalid source type, an unreadable source, and a document that is not well formed, matching the 4GL's message numbers and text. Cover the behaviour with and without NO-ERROR, and the state of the X-DOCUMENT after a failed load.
4. **Validation.** Check the validate argument and SCHEMA-LOCATION / ADD-SCHEMA-LOCATION interaction while in this code.
5. Correct the gap marking once done.

Testcases

Run against OpenEdge for the baseline:

- UTF-8, ISO8859-1 and windows-1252 documents with non-ASCII content, loaded from FILE, MEMPTR and LONGCHAR;
- a LONGCHAR fixed to UTF-8 holding a document that declares ISO8859-1, and the reverse;
- documents with and without a byte order mark and a declaration;
- malformed documents and missing files, with and without NO-ERROR.