

JNI04-J. Do not assume that Java strings are null-terminated

(THIS CODING RULE OR GUIDELINE IS UNDER CONSTRUCTION)

According to [\[JNI Tips\]](#), section "UTF-8 and UTF-16 Strings", Java uses UTF-16 strings that are not null-terminated. UTF-16 strings may contain `\u0000` in the middle of the string, so it is necessary to know the length of the string when working on Java strings in native code.

JNI does provide methods that work with Modified UTF-8 (see [\[API 2013\]](#), Interface `DataInput`, section "Modified UTF-8"). The advantage of working with Modified UTF-8 is that it encodes `\u0000` as `0xc0 0x80` instead of `0x00`. This allows the use of C-style null-terminated strings that can be handled by C standard library string functions. However, arbitrary UTF-8 data cannot be expected to work correctly in JNI. Data passed to the `NewStringUTF()` function must be in Modified UTF-8 format. Character data read from a file or stream cannot be passed to the `NewStringUTF()` function without being filtered to convert the high-ASCII characters to Modified UTF-8. In other words, character data must be normalized to Modified UTF-8 before being passed to the `NewStringUTF()` function. (For more information about string normalization see [IDS01-J. Normalize strings before validating them](#). Note, however, that that rule is mainly about UTF-16 normalization whereas what is of concern here is Modified UTF-8 normalization.)

Noncompliant Code Example

This noncompliant code example shows an example where the wrong type of character encoding is used with erroneous results.

Compliant Solution

In this compliant solution ...

Risk Assessment

If character data is not normalized before being passed to the `NewStringUTF()` function then erroneous results may be obtained.

Rule	Severity	Likelihood	Remediation Cost	Priority	Level
JNI04-J	Low	Probable	Medium	P4	L3

Automated Detection

It may be possible to automatically detect whether character data from untrusted sources has been normalized before being passed to the `NewStringUTF()` function.

Bibliography

JNI Tips	UTF-8 and UTF-16 Strings
API 2013	Modified UTF-8

