Item 319
git_comments:

1. // contract with translators is that they have to understand codepoints // and they just took care of a surrogate pair
2. https://issues.apache.org/jira/browse/LANG-720

git_commits:

1. **summary:** [LANG-720] StringEscapeUtils.escapeXml(input) outputs wrong results when an input contains characters in Supplementary Planes. ALSO rewrite method to avoid modification of counter variable in for loop
   message: [LANG-720] StringEscapeUtils.escapeXml(input) outputs wrong results when an input contains characters in Supplementary Planes. ALSO rewrite method to avoid modification of counter variable in for loop git-svn-id: https://svn.apache.org/repos/asf/commons/proper/lang/trunk@1146844 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** StringEscapeUtils.escapeXml(input) outputs wrong results when an input contains characters in Supplementary Planes.
   description: Hello. I use StringEscapeUtils.escapeXml(input) to escape special characters for XML. This method outputs wrong results when input contains characters in Supplementary Planes. String str1 = "\uD842\uDFB7" + "A"; String str2 = StringEscapeUtils.escapeXml(str1); // The value of str2 must be equal to the one of str1, // because str1 does not contain characters to be escaped. // However, str2 is diffrent from str1. System.out.println(URLEncoder.encode(str1, "UTF-16BE")); //%D8%42%DF%B7A System.out.println(URLEncoder.encode(str2, "UTF-16BE")); //%D8%42%DF%B7%FF%FD The cause of this problem is that the loop to translate input character by character is wrong. In CharSequenceTranslator.translate(CharSequence input, Writer out), loop counter "i" moves from 0 to Character.codePointCount(input, 0, input.length()), but it should move from 0 to input.length().
2. **summary:** StringEscapeUtils.escapeXml(input) outputs wrong results when an input contains characters in Supplementary Planes.
   description: Hello. I use StringEscapeUtils.escapeXml(input) to escape special characters for XML. This method outputs wrong results when input contains characters in Supplementary Planes. String str1 = "\uD842\uDFB7" + "A"; String str2 = StringEscapeUtils.escapeXml(str1); // The value of str2 must be equal to the one of str1, // because str1 does not contain characters to be escaped. // However, str2 is diffrent from str1. System.out.println(URLEncoder.encode(str1, "UTF-16BE")); //%D8%42%DF%B7A System.out.println(URLEncoder.encode(str2, "UTF-16BE")); //%D8%42%DF%B7%FF%FD The cause of this problem is that the loop to translate input character by character is wrong. In CharSequenceTranslator.translate(CharSequence input, Writer out), loop counter "i" moves from 0 to Character.codePointCount(input, 0, input.length()), but it should move from 0 to input.length().
   label: test
3. **summary:** StringEscapeUtils.escapeXml(input) outputs wrong results when an input contains characters in Supplementary Planes.
   description: Hello. I use StringEscapeUtils.escapeXml(input) to escape special characters for XML. This method outputs wrong results when input contains characters in Supplementary Planes. String str1 = "\uD842\uDFB7" + "A"; String str2 = StringEscapeUtils.escapeXml(str1); // The value of str2 must be equal to the one of str1, // because str1 does not contain characters to be escaped. // However, str2 is diffrent from str1. System.out.println(URLEncoder.encode(str1, "UTF-16BE")); //%D8%42%DF%B7A

System.out.println(URLEncoder.encode(str2, "UTF-16BE")); //%D8%42%DF%B7%FF%FD The cause of this problem is that the loop to translate input character by character is wrong. In CharSequenceTranslator.translate(CharSequence input, Writer out), loop counter "i" moves from 0 to Character.codePointCount(input, 0, input.length()), but it should move from 0 to input.length().
label: code-design

4. **summary:** StringEscapeUtils.escapeXml(input) outputs wrong results when an input contains characters in Supplementary Planes.
description: Hello. I use StringEscapeUtils.escapeXml(input) to escape special characters for XML. This method outputs wrong results when input contains characters in Supplementary Planes. String str1 = "\uD842\uDFB7" + "A"; String str2 = StringEscapeUtils.escapeXml(str1); // The value of str2 must be equal to the one of str1, // because str1 does not contain characters to be escaped. // However, str2 is diffrent from str1. System.out.println(URLEncoder.encode(str1, "UTF-16BE")); //%D8%42%DF%B7A System.out.println(URLEncoder.encode(str2, "UTF-16BE")); //%D8%42%DF%B7%FF%FD The cause of this problem is that the loop to translate input character by character is wrong. In CharSequenceTranslator.translate(CharSequence input, Writer out), loop counter "i" moves from 0 to Character.codePointCount(input, 0, input.length()), but it should move from 0 to input.length().
5. **summary:** StringEscapeUtils.escapeXml(input) outputs wrong results when an input contains characters in Supplementary Planes.
description: Hello. I use StringEscapeUtils.escapeXml(input) to escape special characters for XML. This method outputs wrong results when input contains characters in Supplementary Planes. String str1 = "\uD842\uDFB7" + "A"; String str2 = StringEscapeUtils.escapeXml(str1); // The value of str2 must be equal to the one of str1, // because str1 does not contain characters to be escaped. // However, str2 is diffrent from str1. System.out.println(URLEncoder.encode(str1, "UTF-16BE")); //%D8%42%DF%B7A System.out.println(URLEncoder.encode(str2, "UTF-16BE")); //%D8%42%DF%B7%FF%FD The cause of this problem is that the loop to translate input character by character is wrong. In CharSequenceTranslator.translate(CharSequence input, Writer out), loop counter "i" moves from 0 to Character.codePointCount(input, 0, input.length()), but it should move from 0 to input.length().
6. **summary:** StringEscapeUtils.escapeXml(input) outputs wrong results when an input contains characters in Supplementary Planes.
description: Hello. I use StringEscapeUtils.escapeXml(input) to escape special characters for XML. This method outputs wrong results when input contains characters in Supplementary Planes. String str1 = "\uD842\uDFB7" + "A"; String str2 = StringEscapeUtils.escapeXml(str1); // The value of str2 must be equal to the one of str1, // because str1 does not contain characters to be escaped. // However, str2 is diffrent from str1. System.out.println(URLEncoder.encode(str1, "UTF-16BE")); //%D8%42%DF%B7A System.out.println(URLEncoder.encode(str2, "UTF-16BE")); //%D8%42%DF%B7%FF%FD The cause of this problem is that the loop to translate input character by character is wrong. In CharSequenceTranslator.translate(CharSequence input, Writer out), loop counter "i" moves from 0 to Character.codePointCount(input, 0, input.length()), but it should move from 0 to input.length().
7. **summary:** StringEscapeUtils.escapeXml(input) outputs wrong results when an input contains characters in Supplementary Planes.
description: Hello. I use StringEscapeUtils.escapeXml(input) to escape special characters for XML. This method outputs wrong results when input contains characters in Supplementary Planes. String str1 = "\uD842\uDFB7" + "A"; String str2 = StringEscapeUtils.escapeXml(str1); // The value of str2 must be equal to the one of str1, // because str1 does not contain characters to be escaped. // However, str2 is diffrent from str1. System.out.println(URLEncoder.encode(str1, "UTF-16BE")); //%D8%42%DF%B7A System.out.println(URLEncoder.encode(str2, "UTF-16BE")); //%D8%42%DF%B7%FF%FD The cause of this problem is that the loop to translate input character by character is wrong. In CharSequenceTranslator.translate(CharSequence input, Writer out), loop counter "i" moves from 0 to Character.codePointCount(input, 0, input.length()), but it should move from 0 to input.length().
8. **summary:** StringEscapeUtils.escapeXml(input) outputs wrong results when an input contains characters in Supplementary Planes.
description: Hello. I use StringEscapeUtils.escapeXml(input) to escape special characters for XML. This method outputs wrong results when input contains characters in Supplementary Planes. String str1 = "\uD842\uDFB7" + "A"; String str2 = StringEscapeUtils.escapeXml(str1); // The value of str2 must be equal to the one of str1, // because str1 does not contain characters to be escaped. // However, str2 is diffrent from str1. System.out.println(URLEncoder.encode(str1, "UTF-16BE")); //%D8%42%DF%B7A System.out.println(URLEncoder.encode(str2, "UTF-16BE")); //%D8%42%DF%B7%FF%FD The cause of this problem is that the loop to translate input character by character is wrong. In

CharSequenceTranslator.translate(CharSequence input, Writer out), loop counter "i" moves from 0 to Character.codePointCount(input, 0, input.length()), but it should move from 0 to input.length().

jira_issues_comments:

1. Patch for org/apache/commons/lang3/text/translate/CharSequenceTranslator.java.
2. **body:** The patch does not break any unit test with the latest from SVN but it is missing a unit test. Perhaps we should hold off since we are in the middle of a VOTE.
   label: test
3. **body:** I was also going to ask for a unit test, but wanted to improve my understanding of the situation anyway, so adapted the posted problem code. Even though we are currently voting on the release of 3.0.0 from RC4 I don't see why we can't fix this in trunk; the RC tag is already cut. I have used the concept of the patch to rewrite the entire method in question, primarily to avoid the modification of a counter variable within a for loop. Committed revision 1146844.
   label: code-design
4. OK, thanks for the redo. I think we should cut another RC to pick this up.
5. was going to punt to the dev list ;) I just used a sports metaphor. :|
6. Note that we'll release this in 3.0.1. 3.0 will go out with this as a known issue and 3.0.1 will follow (August).