Add non-ASCII file names to zip in Java

Asked 16 years, 3 months ago Modified 12 years, 6 months ago Viewed 24k times

19

What is the best way to add **non-ASCII** file names to a **zip file** using **Java**, in such a way that the files can be properly read in both **Windows** and **Linux?**

Here is one attempt, adapted from https://truezip.dev.java.net/tutorial-6.html#Example, which works in Windows Vista but fails in Ubuntu Hardy. In Hardy the file name is shown as abc-ЖДФ.txt in file-roller.

```
import java.io.IOException;
import java.io.PrintStream;

import de.schlichtherle.io.File;
import de.schlichtherle.io.FileOutputStream;

public class Main {

    public static void main(final String[] args) throws IOException {

        try {
            PrintStream ps = new PrintStream(new FileOutputStream(
                    "outer.zip/abc-ääö.txt"));
            try {
                ps.println("The characters ääö works here though.");
            } finally {
                ps.close();
            }
        } finally {
            File.umount();
        }
    }
}
```

Unlike java.util.zip, truezip allows specifying zip file encoding. Here's another sample, this time explicitly specifiying the encoding. Neither IBM437, UTF-8 nor ISO-8859-1 works in Linux. IBM437 works in Windows.

```
import java.io.IOException;

import de.schlichtherle.io.FileOutputStream;
import de.schlichtherle.util.zip.ZipEntry;
import de.schlichtherle.util.zip.ZipOutputStream;

public class Main {

    public static void main(final String[] args) throws IOException {

        for (String encoding : new String[] { "IBM437", "UTF-8", "ISO-8859-1"
}) {
            ZipOutputStream zipOutput = new ZipOutputStream(
```

```
                    new FileOutputStream(encoding + "-example.zip"), encoding);
            ZipEntry entry = new ZipEntry("abc-ääö.txt");
            zipOutput.putNextEntry(entry);
            zipOutput.closeEntry();
            zipOutput.close();
        }
    }
}
```

java encoding zip

Share

Improve this question

Follow

edited Sep 20, 2008 at 0:35

asked Sep 19, 2008 at 23:25

Micke

2,406 ● 5 ● 20 ● 18

truezip with UTF-8 worked for me on windows 7 and mac os x 10.6.x. It still doesn't work in Linux? – Anton Kuzmin Nov 23, 2009 at 17:37

1 There was a longstanding bug - 9 years in existence - in JDK prior to v7 which prevented correct handling of filenames with names that could not be encoded with IBM CP437. bugs.sun.com/bugdatabase/view_bug.do?bug%5Fid=4244499 It has apparently been fixed in JDK7. blogs.oracle.com/xuemingshen/entry/non_utf_8_encoding_in Therefore one solution seems to be, use JDK7 and the new constructors for ZipInputStream, ZipOutputStream, and ZipFile. – Cheeso Jun 15, 2012 at 16:53

7 Answers

Sorted by: Highest score (default)

10

The encoding for the File-Entries in ZIP is originally specified as IBM Code Page 437. Many characters used in other languages are impossible to use that way.

The PKWARE-specification refers to the problem and adds a bit. But that is a later addition (from 2007, thanks to Cheeso for clearing that up, see comments). If that bit is set, the filename-entry have to be encoded in UTF-8. This extension is described in 'APPENDIX D - Language Encoding (EFS)', that is at the end of the linked document.

For Java it is a known bug, to get into trouble with non-ASCII-characters. See bug #4244499 and the high number of related bugs.

My colleague used as workaround URL-Encoding for the filenames before storing them into the ZIP and decoding after reading them. If you control both, storing and reading, that may be a workaround.

EDIT: At the bug someone suggests using the ZipOutputStream from Apache Ant as workaround. This implementation allows the specification of an encoding.

Share

Improve this answer

Follow

edited Nov 10, 2023 at 19:48

Community Bot

1 • 1

answered Oct 14, 2008 at 11:38

Mnementh

51.3k • 48 • 151 • 202

"Seem to be historically specified as IBM CP437" is a little loose. The PKWare spec says that filenames use IBM437 for the encoding, period. In 2007 PKWare added a standard way to use UTF-8. There are tools that use neither, but that is outside the spec! – Cheeso Mar 28, 2009 at 8:22

1 you wrote "if that bit is set, all filename entries have to be encoded in UTF-8". This is not correct. The use of UTF-8 or IBM437 is per-entry, not per-archive. A spec-compliant zip file can contain some entries with names encoded in UTF-8 and others encoded in IBM437. – Cheeso May 19, 2009 at 15:17

to add confusion, a german install of windows will unpack archives with cp850. fun times. – user3850 Jun 1, 2010 at 16:19

8

In Zip files, according to the spec owned by PKWare, the encoding of file names and file comments is IBM437. In 2007 PKWare extended the spec to also allow UTF-8. This says nothing about the encoding of the files contained within the zip. Only the encoding of the filenames.

I think all tools and libraries (Java and non Java) support IBM437 (which is a superset of ASCII), and fewer tools and libraries support UTF-8. Some tools and libs support other code pages. For example if you zip something using WinRar on a computer running in Shanghai, you will get the Big5 code page. This is not "allowed" by the zip spec but it happens anyway.

The DotNetZip library for .NET does Unicode, but of course that doesn't help you if you are using Java!

Using the Java built-in support for ZIP, you will always get IBM437. If you want an archive with something other than IBM437, then use a third party library, or create a JAR.

Share

Improve this answer

Follow

edited Nov 1, 2010 at 2:54

answered Jan 4, 2009 at 3:49

Cheeso

192k • 105 • 484 • 734

3 Why downvoted anonymously? Don't you like accurate information? – Cheeso Jul 26, 2009 at 13:09

3 I'm still not sure why this answer is being downvoted. It's *still* accurate and correct information. If anyone has any objections, or if anyone thinks the information in this answer is wrong, please speak up. – Cheeso Jun 1, 2010 at 16:32

8

Miracles indeed happen, and Sun/Oracle did really fix the long-living bug/rfe:

Now it's possible to set up filename encodings upon creating the zip file/stream (**requires Java 7**).

Share Improve this answer Follow

edited Jun 18, 2012 at 11:34

answered Jul 28, 2010 at 11:26

Anton Kraievyi
4,332 ● 5 ● 29 ● 42

2 Here's a blog post, live as of June 2012, about the Miracle: blogs.oracle.com/xuemingshen/entry/non_utf_8_encoding_in – Cheeso Jun 15, 2012 at 16:50

7

You can still use the Apache Commons implementation of the zip stream : http://commons.apache.org/compress/apidocs/org/apache/commons/compress/archivers/zip/ZipArchiveOutputStream.html#setEncoding%28java.lang.String%29

Calling setEncoding("UTF-8") on your stream should be enough.

Share Improve this answer Follow

answered Dec 17, 2010 at 16:03

Fengtan
71 ● 1 ● 1

3

From a quick look at the TrueZIP manual - they recommend the JAR format:

> It uses UTF-8 for file name encoding and comments - unlike ZIP, which only uses IBM437.

This probably means that the API is using the java.util.zip package for its implementation; that documentation states that it is still using a ZIP format from 1996. Unicode support wasn't added to the PKWARE .ZIP File Format Specification until 2006.

Share Improve this answer Follow

answered Sep 20, 2008 at 0:09

McDowell
109k ● 31 ● 206 ● 270

0

Did it actually fail or was just a font issue? (e.g. font having different glyphs for those charcodes) I've seen similar issues in Windows where rendering "broke" because the font didn't support the charset but the data was actually intact and correct.

Share Improve this answer Follow

answered Sep 19, 2008 at 23:29

stephbu

5,082 ● 28 ● 42

Thanks for your reply, it is not a font issue because I can create a similarly named file and then zip it and it will display properly. – Micke Sep 19, 2008 at 23:36

0

Non-ASCII file names are not reliable across ZIP implementations and are best avoided. There is no provision for storing a charset setting in ZIP files; clients tend to guess with 'the current system codepage', which is unlikely to be what you want. Many combinations of client and codepage can result in inaccessible files.

Sorry!

Share Improve this answer Follow

answered Sep 19, 2008 at 23:47

bobince

536k ● 110 ● 671 ● 843

According to PKWare's spec, there is a provision for noting that the filename in question is encoded with UTF-8. UTF-8 encoding in zip files is not yet widely supported (== not yet supported in Windows Explorer). When the UTF-8 bit is unset in the zip entry header, the zip spec says that the filename ought to be encoded in IBM437. But you are correct, some apps (WinRar) just encode with the system default code page. Not sure if Windows Explorer does this. Not using the proper encoding when reading a zip can in fact result in inaccessible files. – Cheeso May 19, 2009 at 15:20