blog.SOAL.org

None of this is true. The rest is.

Preserving File Permissions with Ant Tasks

January 23rd, 2011 Robin Leave a comment Go to comments

I confess I rant about Ant a fair bit, but in truth Ant tasks provide a lot of useful functionality. Even if you don't use Ant directly, you can still take advantage of the tasks by using, say, Groovy's AntBuilder.

Ant and File Permissions

Ant's file system tasks, however, have one big disadvantage: they tend to lose or ignore Unix file permissions. This well-documented limitation stems from the fact that Java historically had no built-in APIs for interacting with file permissions. Some Ant tasks, such as Zip, allow you to hardcode specific permissions on fixed filename patterns, but nothing allows you to just leave file permissions as-is – which can be a real pain if you have many executables that get copied around, zipped and unzipped as part of your build/deploy process.

Common workarounds include:

- Avoiding the Ant tasks altogether, and spawning a native copy, zip, tar etc.. processes. This can be messy and prevents you from benefiting from the richness of the Ant tasks (Ant's copy can do way more than cp, such as path mapping, filtering, pattern substitution etc…)
- Chmodding the files as required after the copy/unzip step, either directly from Ant or using a deploy script. This is also messy, particularly if the executables produced by your build change regularly.

Neither of these approaches seemed satisfactory to me, so instead, I had a go at modifying a couple of Ant tasks – Copy and Zip – such that they preserve permissions.

Java and File Permissions

To implement the change, the main question to answer was: what's the best way to interact with file permissions from Java code?

- One option is to spawn native stat, ls -l or chmod processes to read and set permissions. This is as messy as the options above – and potentially expensive if you end up spawning a process for every file you're zipping up or copying.
- Java 6 added some APIs for reading the current user's permissions, and setting the current user's or all users' permissions. Unfortunately, these (arguably half-baked) APIs don't give complete control over the permissions – for example, you can't set different access levels for each of owner, group and others. I understand that Java 7′s NIO2 will remedy this, but that's no good right now.

- We could invoke C standard library functions from Java, so as to stat files to lookup their permissions and chmod them to set the permissions. This can be achieved over JNI, and more easily with JNA, with which you don't need to bother with the C code to glue Java to the native calls.

It turns out the jna-posix library (originally developed for JRuby) provides JNA based utility classes to do exactly what I need. It even drops back to Java's built-in best efforts if it can't find a suitable native library at runtime (which could happen if the code is running on an exotic platform).

CopyWithPerms, ZipWithPerms

So, here's a slightly modified version of the Zip and Copy tasks that use jna-posix to preserve permissions. You should be able to drop the jar in your Ant libs, and just invoke <CopyWithPerms> and <ZipWithPerms> instead of <Copy> and <Zip>.

I also submitted the changes as a patch to Ant's core Copy and Zip tasks.

Categories: Uncategorized Tags: ant, java
Comments (0) Trackbacks (0) Leave a comment Trackback

1. No comments yet.

1. No trackbacks yet.

miho Name (required)
info@michaelhoffer.de E-Mail (will not be published) (required)
www.mihosoft.eu Website

Hi Robin,

thanks for this!

Subscribe to comments feed
Submit Comment
Top 4 Things to do in Australia Change
RSS

Links

- Robin on Facebook
- Robin on last.fm
- Robin on LinkedIn
- Robin on twitter

Tags

ant ASCII Atlassian Australia Banana bread bash charsets classpath conference db2 dns domain pointing dpc dpc09 EBCDIC eclipse EFTPOS encoding github groovy Ibis java jawful Jira Studo junit nginx OAuth pdo php projectzero proxy repository signpost sMash spl subdomain Sydney testfest Tim Tams truncated Twitter VPS zero zero.data zOS

Robin on Twitter

- rewbs: Love the idea that this http://t.co/idbVayf is a mini Dyson sphere, as described here: http://t.co/eWO468r August 15, 2011
- rewbs: http://bit.ly/pKPsFo August 4, 2011
- rewbs: http://bit.ly/nCveyB August 3, 2011

Categories

- Uncategorized

Blogroll

- Ant’s Blog
- ProjectZero blog
- Rob’s Blog
- Robin on Facebook
- Robin on last.fm
- Robin on LinkedIn
- Robin on twitter
- Zoë’s Blog

Archives

- October 2011
- April 2011
- January 2011
- November 2010
- June 2010
- October 2009
- July 2009
- June 2009
- May 2009
- April 2009
- March 2009

Meta

- Log in

Top WordPress

Theme by NeoEase. Valid XHTML 1.1 and CSS 3.