Can RPM and INSTALLP work side-by-side?

Contributed by Michael Felt

INSTALLP and RPM - can they co-exist?

Historically - common practice would seem indicate the answer is yes. However, I would say history has proven they cannot for short or long term.

Found my examples!

Visiting a customer this week - I rediscovered examples of why I starting "rolling my own", i.e., downloading, configuring and installing OSS independently.

At this customer the command rpm -qa returns a list of about 20 rpm packages installed. Most of these have never been updated since being installed between 2010 and 2012.

Note: RPM packages are not part of the standard TL-SP updates - the exception being the occaisional update to the fileset rpm.rte

OpenSSL

rpm -qa shows openssl-0.9.7-something installed. Fortunately this is not the true situation because openssl.base.1.0.1.517 is also installed (which has overwritten the files installed years ago by the rpm packaging).

This is an easy example of why I abhor mixing rpm and installp - installp does not know about rpm files - and vice versa. Obviously, the rpm openssl packaging should be "erased" - but when that happens (if it can happen! rpm -e openssl might just say "cannot be done!") it will remove files it no longer actually controls - as many of the files are now part of the installp package openssl.base.

bzip2

Another example - smaller impact perhaps - is bzip2. This program is installed by default on AIX (as part of rpm.rte). There is also an rpm for bzip2. Why install the rpm version? Maybe because the IBM provided version is ancient - e.g., on AIX 5.3 TL7 the version of bzip2 is 1.0.2 (from rpm.rte.3.0.5.41) The version installed by the customer - if it is following the rpm is version 1.0.5 - but maybe it is an updated version from AIX at version 1.0.6 (rpm.rte.3.0.5.52). And again, if it is the version 1.0.6 (from rpm.rte) - what happens when rpm -e bzip2 is run?

Mixing installp and RPM

Personally - I emphasize personally, I feel mixing RPM and installp packaging a disaster waiting to happen. And above - I give just two examples I have seen - and felt - several times.

Yes, there was a logical reason back in 1999-2000 to start supporting RPM as an install method - Project Montery (Linux) was working on a 64-bit version and it was to run on 64-bit POWER(3) and Intel IA64. Besides 64-bit Linux the IA64 processor was also going to be a target processor for AIX (this was actually available in AIX 5.0 (early adopter version) but never in a GA (generally available) release. And it seemed fitting and logical that AIX/Linux on POWER/IA64 would need support for multiple package managers.

Futher - my humble opinion here - is that RPM packaging helped distinguish between IBM product supported software and open-source-software (OSS) provided by various parties - ASIS.

My bad - and learning mkinstallp

Unfortunately, anno 2005-2006 I kept running into issues when updating AIX when I had also installed many RPM packages - which was quite common at the time. I did not understand why or how (as clearly as I do now). The pain was frequent enough that I started learning how to use the AIX command mkinstallp. In the last 10+ years I have done a lot of experimenting with packaging software using mkinstallp. After having made what I consider "a better mousetrap" I started a second portal "AIXTOOLS" where I put mainly software I am using to support ROOTVG. I also put software that people have requested.

The key message is:

Be careful with mixing RPM and installp. When the contents of the .spec file do not conflict with any files - now (see bzip2 above for an example of "the conflict is now" or in the future (see openssl above for what can happen when the conflict is later) - then no harm done. But without careful and exact scrunity you - as a system admin - will never get a message about a package that has "taken control" of a file.

There is nothing wrong with RPM, but...

Nothing wrong with the technology itself, but RPM (and yum that uses RPM) mixes poorly with installp - in both short and long term - when the goal is comprehensive software management.