Subject: Re: Coyote Library Updates
Posted by David Fanning on Tue, 09 Aug 2011 20:43:57 GMT
View Forum Message <> Reply to Message

Paul van Delst writes:

> I don't think this applies to you, but I've found that to be a common
feeling amongst people that resist using version
> control: it immortalises all their mistakes.
>
> My response to that is: meh. I tend to learn much more from my mistakes and use them as teaching moments (i.e. "What not
> to do" slides in seminar presentations). Anyway...

Yes, I agree that the way most people *become* experts
is by exploring the possibility space in the widest
possible way. It's just a messy business, and observing
it tends to diminish the awe and reverence most experts
are held in, and gives people the idea that "I could do
this, too", which is entirely the wrong idea to be giving
out if you depend on your expertise for making a living. :-(

> It is enough. But it really only makes sense to you
> because you use it all the time. Outside users just
> want a single, reliable place to go to to get updates.

Really!? Then how come you are the only one I ever
hear complaining about this. ;-)

> From my Coding Review and Acceptance policy document (cribbed from the
WRF UCAR document) for the CRTM package I work on:
>
> * A ?major release? contains important new scientific and/or software engineering advancements that significantly
> alter the behavior and/or performance of the CRTM. Numbering for major releases increments
> the first element of the version number. Examples: ?CRTM 2.0.0?, ?CRTM 3.0.0?. Major releases will be
> synchronized across all development subgroups and scheduled well in advance. There will be no more than
> one major release per year.
>
> * A ?minor release? contains minor feature additions and enhancements. Model behavior is not fundamentally
> altered. Numbering for minor releases increments the second element of the version number.
> Examples: ?CRTM 2.1.0?, ?CRTM 3.2.0?. Minor releases will be scheduled well in advance and will
> be coordinated across all developers. However it is not required that every subgroup schedule minor releases

> at the same time. This allows for quicker turnaround of minor feature enhancements by individual
> development subgroups.
>
> * A ?patch release? contains bug fixes. No new features are added. Coordination and scheduling are identical
> to minor releases, except that less advance notice is required. This allows for quick turnaround of bug fixes
> by individual development subgroups. Numbering for patch releases increments the third element of the
> version number. Examples: ?CRTM 2.1.1?, ?CRTM 2.1.2?
>
> Some package also tack on a patch number to a release, e.g. the current ruby release version is 1.9.2-p290. Yeah, 290
> patches to release 1.9.2. No one thinks those folks don't know what they're doing. Have a look at:
> http://www.ruby-lang.org/en/
>
>> And, how would these tagged releases be any different from
>> the files that are already in the SVN repository? I mean, isn't
>> this just extra work?
>
> Yes. But a tag name (e.g. "release_2.1.6") tends to be easier to remember than a particular revision in the repository,
> and it (generally) has more meaning to a development team. Additionally, the tag itself contains all the history
> associated with the files.

Alright, I'll do a little more reading and give this another
try. I'm in and out a little bit this week, taking advantage
of the great weather to do a bit of backpacking and such, so
it may be later in the week before I get to it.

> An anonymous zipfile name has no traceability and, more importantly,
no associated history. You don't know what version
> of your package users have when they ask about problems they're having. Even a date has no meaning if they downloaded
> the zipfile before you fixed it.

I've often wondered why I have the zip file under version
control at all. Do you think it is necessary? My problem is
I need to make a zip file, and it has to have the same name
all the time so my web pages can point to it. How would you
handle this for people with no access to the SVN repository?

> Maybe you should look into using some sort of continuous integration tool to manage all that for you. There are plenty
> third party ones out there. I would assume the google thing you use has something available. (I

think I can hear you
> groaning all the way across the country :o)

You know me WAY too well! ;-)

If I get around to it later, I'll Google "continuous integration tool"
and see if there is anything there I can understand. :-)

> Apologies for the monologue.

Always a pleasure to hear from you!

> And the nagging.

You are like my wife. Both of you always seem to be right. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")