
Subject: Re: IDL 7 and Subversion

Posted by [Paul Van Delst\[1\]](#) on Tue, 18 Dec 2007 13:40:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

b_gom@hotmail.com wrote:

- > Looks like a great resource, but I have a couple of questions:
- >
- > -Can you say whether it is "worth it" to set up subversion? What are
- > the benefits and do they outweigh the hassle of setting it up and the
- > learning curve for using it?

I have never found subversion a hassle to set up at all. It might be; via the IDL workbench (i.e. eclipse). But, setting up a repository on its own via the usual svnadmin commands is pretty easy. Using subversion is also easy - although I am assuming one has used version control software before. Even then, the svn book provides a beginner with excellent information (i.e. the why along with the how).

- > -On the webpage you state that this is designed for a single user. How
- > much of the guide is useful for a multi-user environment?

I haven't read the guide so I don't know what you mean by "this". Do you mean the IDL workbench stuff, or subversion? If it's the latter, then yes it is accessible by multiple-users - that's the point of it. (Note I avoided the use of the word "designed". See below). Keep in mind though that a version control system doesn't mean the developers don't need to communicate regularly. For multi-user development, I would recommend setting up conventions for committing code to the repository. What we're looking at doing (cribbed from the WRF developers):

- 1) Developer publishes to the developer email list what he/she is planning on doing.
- 2) Interested parties respond (good idea, bad idea, waste of time, etc).
- 3) Prior to commit (of completed, or partially complete, code), the developer makes code available for review (with a lead time to permit adequate evaluation of code)
- 4) Interested parties review the code and provide feedback regarding adherence to coding standards, overall design, efficacy of algorithms, etc
- 5) Development lead for the affected portions of code gives go ahead or refusal for commit. (each "portion" of code has a Grand Poobah person that has veto power for a commit. What "portion" means for a group is defined by them)
- 6) Developer commits code (or provides a tarball for commit if they don't yet have access).
- 7) Developer is also responsible for providing documentation and test cases for code.

As far as I can tell, that process seems pretty much tool-independent - i.e. convention over configuration. The actual version control system used also depends a lot on the development process. Subversion doesn't work for everyone. For example, see: <http://subversion.tigris.org/subversion-linus.html>. On the other hand, subversion is the repository for many large, multi-developer software projects; like, for example, the apache web server. I also know of a good number of people that prefer ClearCase for multiple developer projects - there's a *lot* more functionality built-in for distributed developers, but you pay big \$\$ for licenses (sorta like IDL :o).

I doubt that the IDL Workbench hamstrings the ability of multiple developers to access a remote repository (of whatever flavour; subversion cvs, perforce, etc.). Once it's all set up correctly, of course. :o)

cheers,

paulv
