
Subject: Re: Source code management with IDL Workbench
Posted by [Paul Van Delst\[1\]](#) on Thu, 13 Dec 2007 18:31:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Chris Torrence wrote:

- > Hi all,
- >
- > Just thought it would be useful to post a quick summary of how I'm
- > using a source code management (SCM) system like Perforce, Subversion
- > or CVS, from within the IDL Workbench. This is essentially the same
- > info as Chris Padwick's earlier post.
- >
- > 1. Install your favorite SCM plugin feature. I'll use Perforce as an
- > example.
- > 2. Start with a clean IDL Workspace. Do **not** create any projects.
- > 3. Change perspectives to the Perforce perspective (or whatever yours
- > is).
- > 4. Set up your connection to your server. Make sure you can see the
- > files and folders within your depot/repository/
- > 5. Navigate to the folder that contains the IDL code that you want to
- > work on. This folder can contain subfolders, doesn't matter. You need
- > to decide on how best to split up your repository into IDL Projects.
- > 6. Right click on the desired folder within the depot/repository view,
- > choose Team->Import as Project. (menu item may be slightly different
- > depending on your SCM)

I would like to ask some possibly simple/stupid question:

- 1) Code that I use is typically scattered across several directories, organised in a logical (to me at least) manner according to what it does. Does that mean I have to import **all** my directories **every** time I want to create a Project?
- 2) What is the point of importing versioned code as a "Project"?

The second question is the big one I guess. What does the IDE offer me that I don't get via using a version control system and a syntax and context sensitive editor. All I see in using the Workbench is a bunch of extra overhead.

cheers,

paulv

p.s. I can't speak for Perforce, but I don't consider subversion and CVS as source code management systems. A more common definition of SCM I come across all the time is Software Configuration Management - of which version control (using perforce, subversion, whatever) is just one part. E.g. see GForge or Trac for a good start, but what about automated testing, build configuration management, etc etc.

>
> At this point, the IDL Workbench will import that folder into your
> Workspace as a new Project. Because it contains IDL code, it will
> automatically recognize it as an IDL Project. Note that the files are
> **not** copied. They still live over in your "sandbox", wherever that
> happens to be on your system. You don't care.
>
> Some tips on how to use your SCM plugin. This assumes that the IDL
> Project has been imported using the above steps.
>
> * Open up the Preferences dialog, type the word "decoration" into the
> filter box. This should show all pages that have to do with marking or
> decorating the files within the Project Explorer. Go under the first
> page, which should be General->Appearance->Label Decorations. Be sure
> to enable decorations for your SCM plugin. Then go under the other
> decoration pages and change the settings to what you want.
>
> * When you begin editing a file within an IDL Project, it should
> automatically check it out and mark it as writeable. If you've set up
> your decorations correctly, the Project Explorer should now have some
> sort of little decorator indicating that the file is checked out. You
> should also be able to see if other people have the same file checked
> out.
>
> * To add a new file, either create the file within the desired folder
> within your Project, or copy an existing file into the folder. Then
> right click on the file, choose Team->Open for Add (or whatever your
> menu item is). This will mark the file for "add", but won't check it
> in.
>
> * To sync your project (if multiple people are working on the same
> repository), just do Team->Sync to Head. You may need to also do a
> Team->Refresh if you have synced manually outside of the IDL
> Workbench.
>
> Hope this helps.
> -Chris
> ITTVIS
