Subject: Documentation time
Posted by Robbie on Wed, 07 Jan 2009 02:20:52 GMT
View Forum Message <> Reply to Message

I'm in the process of documenting a fairly large IDL code base (47K lines of code in 272 files).
The code base might seem excessive, but I assure you it is nowhere near as large as iTools (152K in 508 files).

I've started migrating to IDLdoc 3.0 and the rst style of documentation. I'm using the IDL workbench and the code is managed using CVS. I have only documented a small portion of the existing code using IDLdoc 2.0

Is there a way to automatically migrate from the old IDLdoc 2.0 format to the new rst format?

Does it make sense to CVS commit the documentation, or should I add it to .cvsignore or just keep it out of the IDL_PATH to be sure?

What value should I put on API documentation? I think that I shouldn't spend more than 50% of my time on API documentation as opposed to other more helpful methods of documentation.

Around 30-50% of my code is redundant code (i.e. deprecated). Should I remove them completely from the CVS tree? Wouldn't it just be confusing to keep it in the source tree?
I'm keen to take a snapshot now and start to manage the non-deprecated code as a completely new project.

Thanks

Robbie

Subject: Re: Documentation time
Posted by Michael Galloy on Wed, 07 Jan 2009 14:32:46 GMT
View Forum Message <> Reply to Message

On Jan 6, 7:20 pm, Robbie <ret...@iinet.net.au> wrote:
> I'm in the process of documenting a fairly large IDL code base (47K
> lines of code in 272 files).
> The code base might seem excessive, but I assure you it is nowhere
> near as large as iTools (152K in 508 files).
>
> I've started migrating to IDLdoc 3.0 and the rst style of
> documentation. I'm using the IDL workbench and the code is managed
> using CVS. I have only documented a small portion of the existing code

> using IDLdoc 2.0
>
> Is there a way to automatically migrate from the old IDLdoc 2.0 format
> to the new rst format?

No, but you can do the migration piecemeal. Don't change your IDLDOC
command and put

```
docformat = 'rst'
```

as the first line of the rst format files. Then old files will use the
IDLdoc style and new files will use the rst format. Migrate over files
as necessary by adding the docformat line.

> Does it make sense to CVS commit the documentation, or should I add it
> to .cvsignore or just keep it out of the IDL_PATH to be sure?

I don't like to add generated file to my source control. I would put
it in .cvsignore and keep it out of the IDL_PATH.

> What value should I put on API documentation? I think that I shouldn't
> spend more than 50% of my time on API documentation as opposed to
> other more helpful methods of documentation.
>
> Around 30-50% of my code is redundant code (i.e. deprecated). Should I
> remove them completely from the CVS tree? Wouldn't it just be
> confusing to keep it in the source tree?
> I'm keen to take a snapshot now and start to manage the non-deprecated
> code as a completely new project.

I would remove it. That's why we have source control.

Mike
--
www.michaelgalloy.com
Tech-X Corporation
Associate Research Scientist