

---

Subject: Version control for IDL software

Posted by [Richard French](#) on Sat, 19 Nov 2005 20:39:08 GMT

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

---

I'm working with several students on the development of a set of data analysis tools in IDL, and I'd like to use some sort of version control for the IDL software. The programs reside on an OSX server but are accessible from a variety of other Mac machines. I'm running IDL under Mac OSX.

I'd appreciate hearing from UNIX folks about the relative merits of CVS and RCS, or other approaches, to keeping track of versions of procedures and functions that are called from other IDL routines, or if this is even possible within the CVS/RCS paradigm.

If these tools don't do the job, I'd like advice on other ways to make sure that I have a record of the exact versions of program files called by a given large IDL program. I'm less worried about two users editing the same file at the same time, and more concerned with coming up with a sensible scheme whereby we can 'freeze' a given version of a routine and feel fairly confident that we know which version is used at any given time.

What I've done in the past is to use a version number in the name of the procedure or function - for example:

```
Pro complicated_procedure_v2,arg1,arg2
End
```

which would then be called from another program as  
complicated\_procedure\_V2, arg1,arg2

However, I don't bump up the version number very often, and it can be a nuisance to change other code to call this new version of the program.

Another approach might be to adopt a convention whereby every function or procedure that I write in the future has a VERSION keyword that would let me determine what version is being used during a given run of the code. This seems a bit clumsy but perhaps some of you have adopted this or a better scheme.

What makes life a bit complicated is that lots of my routines call other utility routines from nifty libraries contributed by many of you folks.

---

---

Subject: Re: Version control for IDL software

Posted by [Craig Markwardt](#) on Sun, 20 Nov 2005 04:11:45 GMT

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

---

"Richard G. French" <[rfrench@wellesley.edu](mailto:rfrench@wellesley.edu)> writes:

- > I'm working with several students on the development of a set of data
- > analysis tools in IDL, and I'd like to use some sort of version control for
- > the IDL software. The programs reside on an OSX server but are accessible
- > from a variety of other Mac machines. I'm running IDL under Mac OSX.
- >
- > I'd appreciate hearing from UNIX folks about the relative merits of CVS and
- > RCS, or other approaches, to keeping track of versions of procedures and
- > functions that are called from other IDL routines, or if this is even
- > possible within the CVS/RCS paradigm.

If you are talking about version control, then yes, I use CVS all the time and it works great. I'm sure that any of the version control systems that have been developed over the past few years would work well too.

- > If these tools don't do the job, I'd like advice on other ways to make sure
- > that I have a record of the exact versions of program files called by a
- > given large IDL program. I'm less worried about two users editing the same
- > file at the same time, and more concerned with coming up with a sensible
- > scheme whereby we can 'freeze' a given version of a routine and feel fairly
- > confident that we know which version is used at any given time.

I would call this "configuration control." Here is what I do for a reasonably large system.

For each of the important routines in the project, I have a line like this:

```
idl_x_cue_rcsid = '$Id: idl_x_cue.pro,v 1.7 2001/07/21 17:41:12 craigm Exp $'
```

Of course, each time I change the module, CVS will automatically update this line to the new version info, so this implants version information into the procedure automatically.

Then step two is a simple cron job which runs every night that does  

```
grep "_rcsid =" *.pro > idl_x_rcsids.pro
```

This works fine if you are using your development version for real work. Of course if you have a production-level "release" where the modules are frozen, you can do this manually once and then freeze the `idl_x_rcsids.pro` file too.

The last step is to have a simple routine like this to print all the relevant version information.

```
pro idl_x_prversion
  print, !version.release, !version.os, !version.arch, $
  format=('"    IDL Version: ",A0," ",A0," ",A0)'
  cd, current=cwd
```

```

print, ' Current Directory: ', cwd
hostname = getenv('HOST')
if hostname(0) EQ " then spawn, 'hostname', hostname
print, ' Current Host: ', hostname
@idl_x_rcsids.pro
vars = routine_info('idl_x_prversion', /variables)
for i = 0, n_elements(vars)-1 do begin
  if strpos(vars(i), '_RCSID') GE 0 then $
    dummy = execute('print, '+vars(i))
endfor
end

```

This could be done differently, by stuffing the rcsids into a structure or something like that, but I was looking for the minimum amount of effort that preserved the maximum version information. And the most important thing is that it is all automatic so I never have to touch anything for it to happen.

Hope that helps. Good luck!  
Craig

--

-----  
 Craig B. Markwardt, Ph.D.    EMAIL: craigmnet@REMOVEcow.physics.wisc.edu  
 Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response  
 -----

---

Subject: Re: Version control for IDL software  
 Posted by [mmiller3](#) on Mon, 21 Nov 2005 14:38:21 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

>>>> > "Craig" == Craig Markwardt <craigmnet@REMOVEcow.physics.wisc.edu> writes:

> For each of the important routines in the project, I have a  
 > line like this:

> idlx\_cue\_rcsid = '\$Id: idlx\_cue.pro,v 1.7 2001/07/21  
 > 17:41:12 craigm Exp \$'

I've been trying to think of a nice way to do something like this for objects. I thought that an object could inherit from "versioned", which would provide a version attribute and methods for querying it. I haven't yet sorted out how best to do this with multiple inheritance though. For example, if objects A and B inherit from versioned, no problem. If A and B inherit from versioned, and A inherits from B, then there is a name clash that results in "% Conflicting or duplicate structure tag

definition"s. There may be no way around this (with IDL), so perhaps an object that is a version registrar would be the way to go.

At any rate, to address the original poster's question, our group has been using CVS extensively with good success. If there are to be multiple developers (and there almost always are), go with CVS rather than RCS.

Mike

---

---

Subject: Re: Version control for IDL software  
Posted by [Ben Panter](#) on Mon, 21 Nov 2005 16:48:02 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Richard G. French wrote:

- > I'd appreciate hearing from UNIX folks about the relative merits of CVS and
- > RCS, or other approaches, to keeping track of versions of procedures and
- > functions that are called from other IDL routines, or if this is even
- > possible within the CVS/RCS paradigm.

Just to add to the others singing the praises of CVS, it's great - even if you're not working in a team environment. Versioning and forcing myself to comment versions on a spiralling-out-of-control code base has helped me immensely.

JD said a while ago that IDLWAVE can use the emacs CVS functions, but I tend to use eclipse - which looks exactly the same on my unix, linux and windows boxes. For a processor farm there is the script-friendly unix command line version.

HTH

Ben

--

Ben Panter, Garching, Germany.  
Email false, <http://www.benpanter.co.uk>  
or you could try ben at ^^^^^^^^^^^

---

---

Subject: Re: Version control for IDL software  
Posted by [JD Smith](#) on Mon, 21 Nov 2005 18:49:02 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Mon, 21 Nov 2005 17:48:02 +0100, Ben Panter wrote:

> Richard G. French wrote:

>

>> I'd appreciate hearing from UNIX folks about the relative merits of CVS  
>> and RCS, or other approaches, to keeping track of versions of procedures  
>> and functions that are called from other IDL routines, or if this is  
>> even possible within the CVS/RCS paradigm.

>

> Just to add to the others singing the praises of CVS, it's great - even if  
> you're not working in a team environment. Versioning and forcing myself to  
> comment versions on a spiralling-out-of-control code base has helped me  
> immensely.

>

> JD said a while ago that IDLWAVE can use the emacs CVS functions, but I  
> tend to use eclipse - which looks exactly the same on my unix, linux and  
> windows boxes. For a processor farm there is the script-friendly unix  
> command line version.

Yep, but in fact it's just Emacs which interfaces to CVS via its VC mode. It also interfaces to RCS, subversion, arch, and others, so really you can use any of them. I use CVS and am relatively happy with it. I especially like the color-coded "annotate" mode, which shows who last modified which line when (Emacs 22 will improve this as well). My most used command is C-x v =, which shows the differences between the current and checked in versions of a file.

The only drawback to CVS (which is widely known, and part of the impetus for developing new tools like Subversion) is that changing the name of or relocating files is somewhat awkward. I can recommend this free introductory book to CVS:

<http://cvsbook.red-bean.com/cvsbook.html>

JD

---

---

Subject: Re: Version control for IDL software

Posted by [Michael Wallace](#) on Mon, 21 Nov 2005 19:09:58 GMT

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

---

> The only drawback to CVS (which is widely known, and part of the  
> impetus for developing new tools like Subversion) is that changing the  
> name of or relocating files is somewhat awkward.

---

---

Subject: Re: Version control for IDL software

Posted by [Michael Wallace](#) on Mon, 21 Nov 2005 19:13:42 GMT

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

---

Like everyone else, I'll recommend CVS. I've been using it for a while, and it is especially powerful when you use the advanced features of tagging and branching. While that level of version control might seem overkill, tagging is a great way to keep your "freezes" frozen. That is, one of your needs is to make sure that everything works well with everything else. When you reach that point, you can create a tag for your files/revisions. You can now work towards the next version, but it's really easy to just pull one of your previous tags. So, people who need a specific version of the files can get that, but others can continue development.

> The only drawback to CVS (which is widely known, and part of the  
> impetus for developing new tools like Subversion) is that changing the  
> name of or relocating files is somewhat awkward.

I am a fan of Subversion, but because it is newer, there aren't as many tools that have it integrated in by default. If you're just going to use it on the command line, this isn't an issue.

-Mike

---

---

Subject: Re: Version control for IDL software

Posted by [James Kuyper](#) on Mon, 21 Nov 2005 20:07:22 GMT

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

---

Michael Wallace wrote:

> Like everyone else, I'll recommend CVS. I've been using it for a while,  
> and it is especially powerful when you use the advanced features of  
> tagging and branching. While that level of version control might seem  
> overkill, tagging is a great way to keep your "freezes" frozen. That  
> is, one of your needs is to make sure that everything works well with  
> everything else. When you reach that point, you can create a tag for  
> your files/revisions. You can now work towards the next version, but  
> it's really easy to just pull one of your previous tags. So, people who  
> need a specific version of the files can get that, but others can  
> continue development.

We use RCS version labels for that purpose. Can you compare CVS tags with RCS version labels for me?

---

---

Subject: Re: Version control for IDL software

Posted by [Michael Wallace](#) on Mon, 21 Nov 2005 20:26:58 GMT

- > We use RCS version labels for that purpose. Can you compare CVS tags
- > with RCS version labels for me?

I believe they are the same thing. CVS just uses a different name.

-Mike

---

---

Subject: Re: Version control for IDL software  
Posted by [Antonio Santiago](#) on Tue, 22 Nov 2005 07:06:25 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Richard G. French wrote:

- > I'm working with several students on the development of a set of data
- > analysis tools in IDL, and I'd like to use some sort of version control for
- > the IDL software. The programs reside on an OSX server but are accessible
- > from a variety of other Mac machines. I'm running IDL under Mac OSX.
- >
- > I'd appreciate hearing from UNIX folks about the relative merits of CVS and
- > RCS, or other approaches, to keeping track of versions of procedures and
- > functions that are called from other IDL routines, or if this is even
- > possible within the CVS/RCS paradigm.

Rememver SVN. It is the big/new brother of CVS.

- >
- > If these tools don't do the job, I'd like advice on other ways to make sure
- > that I have a record of the exact versions of program files called by a
- > given large IDL program. I'm less worried about two users editing the same
- > file at the same time, and more concerned with coming up with a sensible
- > scheme whereby we can 'freeze' a given version of a routine and feel fairly
- > confident that we know which version is used at any given time.

>

Yes, they can edit the same file and "merge" it later. You can restore a previous version, as old as it exists in the repository, instead to rewrite some modifications. You can create new branches of development and tag a state as a "first test version".

- > What I've done in the past is to use a version number in the name of the
- > procedure or function - for example:
- >
- > Pro complicated\_procedure\_v2,arg1,arg2
- > End
- >
- > which would then be called from another program as
- > complicated\_procedure\_V2, arg1,arg2
- >

> However, I don't bump up the version number very often, and it can be a  
> nuisance to change other code to call this new version of the program.  
>  
> Another approach might be to adopt a convention whereby every function or  
> procedure that I write in the future has a VERSION keyword that would let me  
> determine what version is being used during a given run of the code. This  
> seems a bit clumsy but perhaps some of you have adopted this or a better  
> scheme.  
>  
> What makes life a bit complicated is that lots of my routines call other  
> utility routines from nifty libraries contributed by many of you folks.

>

--

-----  
Antonio Santiago Piñerez  
( email: santiago<at>grahi.upc.edu )  
( www: http://www.grahi.upc.edu/santiago )  
( www: http://asantiago.blogspot.org )  
-----

GRAHI - Grup de Recerca Aplicada en Hidrometeorologia  
Universitat Politècnica de Catalunya  
-----

---

Subject: Re: Version control for IDL software  
Posted by [Jean\[1\]](#) on Thu, 24 Nov 2005 17:21:52 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

> What I've done in the past is to use a version number in the name of the  
> procedure or function - for example:  
>  
> Pro complicated\_procedure\_v2,arg1,arg2  
> End  
>  
> which would then be called from another program as  
> complicated\_procedure\_V2, arg1,arg2  
>  
> However, I don't bump up the version number very often, and it can be a  
> nuisance to change other code to call this new version of the program.

For this specific issue, just write a generic procedure  
Pro complicated\_procedure,arg1,arg2 that will call  
complicated\_procedure\_v2,arg1,arg2  
So all your pros can always call the same procedure, and you just have

to update this one to reflect the changes!

.. but CVS seems much more elegant! ;)

Jean H.

jghasban@DELTHIS.ucalgary.ANDTHIS.c.a

---

---

Subject: Re: Version control for IDL software

Posted by [Richard French](#) on Thu, 24 Nov 2005 17:40:14 GMT

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

---

Thanks to Jean and Craig and others - I've now started using CVS and I'm beginning to get the hang of it. I haven't yet sorted out the 'tags' part of things, or freezing versions of a full set of routines, but I'm getting there!

Dick

---

---

Subject: Re: Version control for IDL software

Posted by [Craig Markwardt](#) on Fri, 25 Nov 2005 18:25:09 GMT

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

---

"Richard G. French" <rfrench@wellesley.edu> writes:

> Thanks to Jean and Craig and others - I've now started using CVS and I'm  
> beginning to get the hang of it. I haven't yet sorted out the 'tags' part of  
> things, or freezing versions of a full set of routines, but I'm getting  
> there!

Here's a data point for you: I've never used CVS tags or frozen versions in my life. I use CVS to document changes; to embed version info into source modules; to synchronize changes between environments; to clone a library to another machine, and in my "real job," to synchronize with multiple developers.

My suggestion is for you to get used to CVS check-in/out, update and commit, and that will probably be all you need.

Craig

--

-----  
Craig B. Markwardt, Ph.D.    EMAIL: [craigmnet@REMOVEcow.physics.wisc.edu](mailto:craigmnet@REMOVEcow.physics.wisc.edu)  
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response  
-----

---

Subject: Re: Version control for IDL software  
Posted by [Paul Van Delst\[1\]](#) on Mon, 28 Nov 2005 22:04:41 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Craig Markwardt wrote:

> "Richard G. French" <rfrench@wellesley.edu> writes:  
>  
>> Thanks to Jean and Craig and others - I've now started using CVS and I'm  
>> beginning to get the hang of it. I haven't yet sorted out the 'tags' part of  
>> things, or freezing versions of a full set of routines, but I'm getting  
>> there!  
>  
>  
> Here's a data point for you: I've never used CVS tags or frozen  
> versions in my life. I use CVS to document changes; to embed version  
> info into source modules; to synchronize changes between environments;  
> to clone a library to another machine, and in my "real job," to  
> synchronize with multiple developers.  
>  
> My suggestion is for you to get used to CVS check-in/out, update and  
> commit, and that will probably be all you need.

I'm not disagreeing with you, but I also don't think investigation into how to use CVS tags should be dismissed so lightly. I think it depends greatly on how someone sets up or uses their repository or, more likely, structures their software development. Your case is just a single data point.

Tags can be a powerful help when he has to retrieve a particular release of software from XX months/years ago, or he wants to patch older software (via branches) for some users. If he never needs to do either of those things, then, yes, tags and branches may complicate rather than simplify his life.

To Richard: the standard documentation that typically comes with CVS (the Cederqvist et al manual) is a great resource but I've found it a bit light on examples when it comes to doing more than the basic things. Experimentation on a dummy repository, or subdirectory within your "regular" repository, may be helpful in that case. The CVS newsgroup is also a trove (especially if you're prepared to search the group).

paulv

--

Paul van Delst  
CIMSS @ NOAA/NCEP/EMC

---

---

Subject: Re: Version control for IDL software  
Posted by [Ken Mankoff](#) on Sun, 04 Dec 2005 16:30:41 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Thu, 24 Nov 2005, Richard G. French wrote:

- > Thanks to Jean and Craig and others - I've now started using CVS
- > and I'm beginning to get the hang of it. I haven't yet sorted out
- > the 'tags' part of things, or freezing versions of a full set of
- > routines, but I'm getting there!
- > Dick

Looks like I'm too late to influence your decision, but I switched to subversion from CVS about 1 year ago and its been great. Many improvements over CVS. I suggest it instead.

Whichever you use, a nice wrapper like Trac, webCVS, etc. can make it easy to see whats going on. I think Trac supports CVS, if not, it will in an upcoming release.

<http://projects.edgewall.com/trac>

-k.

---

Subject: Re: Version control for IDL software

Posted by [Paul Van Delst\[1\]](#) on Mon, 05 Dec 2005 16:05:49 GMT

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

---

Ken Mankoff wrote:

- >
- > On Thu, 24 Nov 2005, Richard G. French wrote:
- >
- >> Thanks to Jean and Craig and others - I've now started using CVS and
- >> I'm beginning to get the hang of it. I haven't yet sorted out the
- >> 'tags' part of things, or freezing versions of a full set of routines,
- >> but I'm getting there!
- >> Dick
- >
- >
- > Looks like I'm too late to influence your decision, but I switched to
- > subversion from CVS about 1 year ago and its been great. Many
- > improvements over CVS. I suggest it instead.

We'll soon be switching to subversion for a new CVS repository here too. I've experienced issues with CVS commits over a network when ssh is used, \*and\* some (or all) of the files are being watched by other developers -- commits seem to fail unpredictably. Since CVS doesn't have atomic commits (subversion does), I have to keep an eye on exactly what has been committed before a failure occurs (if at all). That can be a real pain when you're committing or tagging a whole bunch of files spread through several subdirectories heirarchies.

But, for a single person-repository not handling a lot of code, CVS is probably good enough.

paulv

> Whichever you use, a nice wrapper like Trac, webCVS, etc. can make it  
> easy to see whats going on. I think Trac supports CVS, if not, it will  
> in an upcoming release.  
>  
> <http://projects.edgewall.com/trac>  
>  
> -k.

--

Paul van Delst  
CIMSS @ NOAA/NCEP/EMC

---