Subject: Re: One file for each procedure/function?
Posted by David Burridge on Fri, 19 Apr 2002 13:44:01 GMT
View Forum Message <> Reply to Message

Hi Pepijn,

"Pepijn Kenter" <p.t.kenter@twi.tudelft.nl> wrote in message
news:M%Tv8.115$2I3.3490@castor.casema.net...
<snip>
> I'm under the strong impression that this is not the normal way of working
> and that you should create one file for each function or procedure, which
is
> than compiled automaticly when it's needed. These files should be put in a
> directory that is included in the !path system variable.
>
> Still, I would like to keep functions that belong to each other grouped
> together; if not in one file, then at least in one directory. So I've
> devided the routines over a few directories and used the expand_path
> function to include these directories in the !path variable.
>
> Is this the best way of working?
> Or are there better way's of ordering your routines (like units in
pascal)?
> I want to keep using the command line to start my programs (i.e. i don't
> want to be dependend on .prj files)
>
> I'm not happy with the 'one file for each function' concept but if this is
> the way IDL is designed I think I'd better stick to it, rather than using
> obscure tricks to circumvent it.

As you mention, one file per routine is best and it makes the code easier to debug too, as the routines are easier to find. Some programmers collect routines together in one file when the code is interdependent. For example, if you put the event handler for a widget program at the top of the widget program file, it gets compiled when the main program is called, so is always 'ready-compiled' when it's needed.

I would advocate the use of directories in you case, coupled with the '+' switch in your path statement. If you want to package up your code into 'libraries', an alternative is to put them into a save file which will also cut the compile overhead.

Hope this is helpful,

Dave