

---

Subject: Re: MAKE\_DLL for DLMs  
Posted by [Haje Korth](#) on Thu, 20 Jan 2005 13:09:09 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

JD,  
yes, I have used it and I did not have to spend much though on it to get it running. Some example code is located here:  
<http://dysprosium.jhuapl.edu/idltoaacgm/>. Download the UNIX version, Windows is binaries only.

Cheers,  
Haje

"JD Smith" <jdsmith@as.arizona.edu> wrote in message  
news:pan.2005.01.19.22.39.13.418876@as.arizona.edu...

>  
>  
> Has anyone used MAKE\_DLL for DLMs? I've used it with CALL\_EXTERNAL  
> with no problem, but I'm confused about the DLM search path. Is  
> !MAKE\_DLL.COMPILE\_DIRECTORY included on it by default? It seems not.  
> If so, what's the typical way people are discovering and loading DLMs  
> that end up in ~/.idl/idl\_6\_X\_X\_linux\_x86\_m32\_f64/compile\_dir or the  
> like? I suppose you can reference it directly with DLM\_REGISTER, but  
> isn't that defeating the purpose?  
>  
> JD  
>

---

Subject: Re: MAKE\_DLL for DLMs  
Posted by [JD Smith](#) on Thu, 20 Jan 2005 16:58:37 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Thu, 20 Jan 2005 08:09:09 -0500, Haje Korth wrote:

> JD,  
> yes, I have used it and I did not have to spend much though on it to get it  
> running. Some example code is located here:  
> <http://dysprosium.jhuapl.edu/idltoaacgm/>. Download the UNIX version, Windows  
> is binaries only.

Your solution is the one I stumbled on as well: rather than let  
MAKE\_DLL build the shared library in its ~/.idl/idl\_blah/compile\_dir  
(!MAKE\_DLL.COMPILE\_DIRECTORY) directory, output it instead to the  
directory which contains the .dlm file. However, this isn't really  
ideal for site-wide, read-only setups, in which the source directories

aren't writable. Any other thoughts? I suppose I could add !MAKE\_DLL.COMPILE\_DIRECTORY to the !DLM\_PATH on startup, but I hate to muck with people's paths.

JD

---

---

Subject: Re: MAKE\_DLL for DLMs  
Posted by [Haje Korth](#) on Fri, 21 Jan 2005 13:03:45 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

JD,  
I usually write all my software for myself and never worry about network installation. Why does every user need to compile the dlm separately? Can't you compile it and throw it in the idl binary directory for everyone to use? The only other possibility I see is that the user uses make\_dll in a local directory and they set ! dlm\_path to point to it.

Cheers,  
Haje

"JD Smith" <jdsmith@as.arizona.edu> wrote in message  
news:pan.2005.01.20.16.58.37.149684@as.arizona.edu...  
> On Thu, 20 Jan 2005 08:09:09 -0500, Haje Korth wrote:  
>  
>> JD,  
>> yes, I have used it and I did not have to spend much though on it to get  
it  
>> running. Some example code is located here:  
>> <http://dysprosium.jhuapl.edu/idltoaacgm/>. Download the UNIX version,  
Windows  
>> is binaries only.  
>  
> Your solution is the one I stumbled on as well: rather than let  
> MAKE\_DLL build the shared library in its ~/.idl/idl\_blah/compile\_dir  
> (!MAKE\_DLL.COMPILE\_DIRECTORY) directory, output it instead to the  
> directory which contains the .dlm file. However, this isn't really  
> ideal for site-wide, read-only setups, in which the source directories  
> aren't writable. Any other thoughts? I suppose I could add  
> !MAKE\_DLL.COMPILE\_DIRECTORY to the !DLM\_PATH on startup, but I hate to  
> muck with people's paths.  
>  
> JD  
>

---

Subject: Re: MAKE\_DLL for DLMS  
Posted by [David Fanning](#) on Fri, 21 Jan 2005 14:05:06 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Haje Korth writes:

> I usually write all my software for myself and never worry about network  
> installation. Why does every user need to compile the dlm separately? Can't  
> you compile it and throw it in the idl binary directory for everyone to use?  
> The only other possibility I see is that the user uses make\_dll in a local  
> directory and they set ! dlm\_path to point to it.

I'll speak for JD, who is too busy counting parentheses this morning to answer.

The problem with distributing software is that not all of your users are in the building. Some are in their own buildings, states, countries, etc. And most don't have the foggiest idea of what they are doing.

So...you have to do it for them. Making someone change his own path is perilous. Half the users don't know how to. Of the other half, half of them will put the path in the wrong order. Perhaps half of one percent will actually know how to figure out which COLORBAR routine they are actually using.

Even projects don't help as much as you would hope. Often, my client applications will include several different libraries or projects. To get them distributed properly, I have to make a single project for the client. But that means I have multiple copies of my library on my machine. Aaaachhhh!

Life would be so much easier if we could just combine several projects in the same project file. Perhaps I'm going to have to go get the ol' Emacs manual out again. :-)

Cheers,

David

--

David Fanning, Ph.D.  
Fanning Software Consulting, Inc.  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

---

Subject: Re: MAKE\_DLL for DLMS  
Posted by [Haje Korth](#) on Fri, 21 Jan 2005 15:28:27 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David,

Hm, it looks like I avoid a whole set of problems by developing mainly for myself instead of others. The problem with compiling for others starts with UNIX OS. While on Windows, people are in a not always functioning but controlled environment, while on UNIX you have fifty versions with a hundred flavors each, and compiling for all these combinations is a real pain. So I decided to leave compiling to the experts on UNIX, and for idiots like me using Windows I offer conveniently compiled versions...

Cheers,  
Haje

"David Fanning" <davidf@dfanning.com> wrote in message  
news:MPG.1c5ab9fbe9b6529c9898e0@news.frii.com...

> Haje Korth writes:

>

>> I usually write all my software for myself and never worry about network  
>> installation. Why does every user need to compile the dlm separately?

Can't

>> you compile it and throw it in the idl binary directory for everyone to  
use?

>> The only other possibility I see is that the user uses make\_dll in a  
local

>> directory and they set ! dlm\_path to point to it.

>

> I'll speak for JD, who is too busy counting parentheses this morning  
> to answer.

>

> The problem with distributing software is that not all of your users  
> are in the building. Some are in their own buildings, states, countries,  
> etc. And most don't have the foggiest idea of what they are doing.

>

> So...you have to do it for them. Making someone change his own  
> path is perilous. Half the users don't know how to. Of the other  
> half, half of them will put the path in the wrong order. Perhaps  
> half of one percent will actually know how to figure out which  
> COLORBAR routine they are actually using.

>

> Even projects don't help as much as you would hope. Often, my client  
> applications will include several different libraries or projects.

> To get them distributed properly, I have to make a single project  
> for the client. But that means I have multiple copies of my  
> library on my machine. Aaaachhhh!

>

> Life would be so much easier if we could just combine several  
> projects in the same project file. Perhaps I'm going to have  
> to go get the ol' Emacs manual out again. :-(

>  
> Cheers,  
>  
> David  
>  
> --  
> David Fanning, Ph.D.  
> Fanning Software Consulting, Inc.  
> Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

---

---

Subject: Re: MAKE\_DLL for DLMs  
Posted by [David Fanning](#) on Fri, 21 Jan 2005 16:02:03 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Haje Korth writes:

>> Even projects don't help as much as you would hope. Often, my client  
>> applications will include several different libraries or projects.  
>> To get them distributed properly, I have to make a single project  
>> for the client. But that means I have multiple copies of my  
>> library on my machine. Aaaachhhh!

The worst, of course, is the client who never installs my software updates in a single directory, but who makes a new directory for everything I send him. Those programs *\*never\** run correctly (although I can't do any debugging since they run perfectly on my machine). Usually, I have to go out there with a computer vacuum cleaner and delete half his hard drive. :-(

Cheers,

David

--  
David Fanning, Ph.D.  
Fanning Software Consulting, Inc.  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

---

---

Subject: Re: MAKE\_DLL for DLMs  
Posted by [JD Smith](#) on Fri, 21 Jan 2005 17:55:21 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Fri, 21 Jan 2005 08:03:45 -0500, Haje Korth wrote:

> JD,

- > I usually write all my software for myself and never worry about network
- > installation. Why does every user need to compile the dlm separately? Can't
- > you compile it and throw it in the idl binary directory for everyone to use?
- > The only other possibility I see is that the user uses make\_dll in a local
- > directory and they set ! dlm\_path to point to it.

The main reason MAKE\_DLL exists is to provide a platform-agnostic way to compile C files into DLMS, which will work on most systems which have a compiler, now or in the future. Distributing IDL code is relatively easy: take care not to use indistinct names that will clash, tell the user to drop the package in with their other IDL packages, and off you go, with nary a care about updated IDL versions, etc.

Distributing externally compiled binaries raises the bar considerably, as you now have to contend with various OS/compiler/library/IDL exported interface versioning issues. So, compiling once and throwing it in with the IDL source just doesn't cut it. You can either have a tedious Makefile that the user is required to run, or simply use MAKE\_DLL. This solution is almost perfect, except that the default user-writable location for compiled libraries (in ~/.idl/idl\_blah/compile\_dir) isn't visible to IDL's DLM search, which means you have to move it somewhere which is, which means you have to make assumptions about write permissions. Not pretty.

JD

---