

Re: sources/Build.exe alternate source file location question

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.kernel/2004-05/0742.html>

From: Mark Roddy (*mrodgy_at_nospam.spam*)

Date: 05/29/04

Date: Sat, 29 May 2004 08:32:40 -0400

On Fri, 28 May 2004 13:55:41 -0700, Ray Trent
<rat@synaptics.com.spamblock> wrote:

>Mark Roddy wrote:

>> I give up, what do 'compiling things as libraries' and having 'a dedicated

>> build team' have to do with each other?

>

>Setting this up is a hassle that has nothing to do with actual coding.

>It's not a trivial thing to do with BUILD either, unless you're already

>a build engineer that understands it well enough that you don't have to

>do computer science to figure it out every time :-).

>

>To be fair, the documentation's much better these days than it was when

>we developed our build process (back in the NT4 days when it was

>basically completely undocumented). Every time you want to change

>something, yet more management of the build process becomes necessary.

>All of this overhead is something that some companies deal with using a

>dedicated build team.

>

>> So for NT style builds you just create program libraries for your two share

>> directories and link against them. For other build environments you can

>> either compile the individual source modules or also use libraries.

>

>Yes, well, there are also shared header files, which can't be compiled

>into libraries at all... Yes, it's possible to hardcode the paths or put

>them all in a big pile somewhere and point to that pile, but again, it's

>a pain to manage, and not very well organized.

>

Header file paths are not a problem. It is only source file paths that have restrictions. You can leave your source tree intact. All you have to change, for NT, is to build the source files in question into one or more static library modules and link your targets against them. This is simply not a big deal. An afternoons work ought to suffice.

microsoft.public.win32.programmer.kernel: Re: sources/Build.exe alternate source file location question

>> *Generally there are better ways to share source code modules than to try to
>> force all platform build environments to fit into one common build system.
>> For example: use a good source code management system, partition your common
>> code base out as a separate module under your SCMS, and share it out of the
>> SCMS into each platform specific build system in the manner best suited to
>> that system.*
>
>*Ok, that's great, but many people don't want to pay for and train up on
>that kind of SCMS, and prefer to use something like CVS.*

CVS works just fine with this sort of arrangement. Had you said
'Visual Source Safe' you would have a point :-)

>*Also again,
>managing something like this takes time that could usually be better
>spent on writing code and fixing bugs.*

Well I disagree there. Not paying the cost of creating and maintaining
a good build system is, in my opinion, always a mistake, and will
inevitably come back and clobber you at a (murphy's law determined)
most unfortunate point in your development cycle.

=====

Mark Roddy
Windows .NET/XP/2000 Consulting
Hollis Technology Solutions 603-321-1032
www.hollistech.com