

Re: New driver loading scheme for Project Evil, need input

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/current/2005-04/0898.html>

From: Bill Paul (wpaul_at_FreeBSD.ORG)

Date: 04/20/05

To: eksffa@freebsdbrasil.com.br (Patrick Tracanelli)

Date: Wed, 20 Apr 2005 17:08:02 +0000 (GMT)

> [skip]
>
>> - What should the script be called? *wintobsd.sh* sounds kind of lame.
>
> *mkndisko.sh*? =)
>
>> Any suggestions or comments on my mad scheme would be welcome.
>
> Is it possible to detect if the driver needs other files and prompt the
> user for this files' path (or assume a path to be looked for first) to
> copy it to */compat/ndis/* ?
>
> Ie, the user inputs */tmp/driver-foo_winxp-ver-bar/driver-foo.inf* (same
> for sys) and the script tries to find the necessary files *foo-a.bin* and
> *foo-b.bin* under */tmp/driver-foo_winxp-ver-bar/* and copies it, or ask
> for the path if cant find it...

The way Microsoft does it, the *.inf* is supposed to tell the installer about all the files included in the distribution which have to be copied into *C:\WINNT\SYSTEM32*. This includes any firmware images or additional files that have to be loaded via *NdisOpenFile()*.

Doing this in Project Evil would require a much more sophisticated *.inf* file parser. The existing parser in *ndiscvt(8)* is only smart enough to extract the registry keys and device ID info. It could be made smarter, but I think this is a project for another day. For now, I will probably just have the script ask the user if he/she has any additionaly firmware images that need to be processed.

Note that there are actually two methods for getting firmware files loaded. The simple method is to copy them to */compat/ndis* and just let *NdisOpenFile()* grab them, but this only works if you *kldload* your driver after the system is up and running multiuser. If you pre-load the driver via */boot/loader.conf*, this will fail: during initial kernel bootstrap, the filesystems aren't mounted yet.

freebsd-current: Re: New driver loading scheme for Project Evil, need input

So in that scenario, you need the second method, which is to convert the firmware file into a .ko and pre-load that along with the driver. This conversion process is much simpler: the file is packaged up using objcopy(1) and the turned into a shared object with ld(1). Once the file is kldloaded or pre-loaded, NdisOpenFile() is smart enough to find it by scanning the loaded module list. A similar trick can also be used to link the firmware .o and the driver .o into a static kernel image. (This is mostly handled by the -f flag to ndiscvt(8).)

-Bill

```
--  
=====
```

-Bill Paul	(510) 749-2329	Senior Engineer, Master of Unix-Fu
	wpaul@windriver.com	Wind River Systems

```
=====
```

<adamw> you're just BEGGING to face the moose

```
=====
```

freebsd-current@freebsd.org mailing list
<http://lists.freebsd.org/mailman/listinfo/freebsd-current>
To unsubscribe, send any mail to "freebsd-current-unsubscribe@freebsd.org"