

Re: [linux-dvb] DST/BT878 module customization (.. was: Critical points about ...)

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-05/msg00699.html>

- *From:* Trent Piepho <xyzyzy@xxxxxxxxxxxxxxxx>
 - *Date:* Tue, 1 May 2007 15:57:47 -0700 (PDT)
-

On Tue, 1 May 2007, Simon Arlott wrote:

On 30/04/07 22:17, Markus Rechberger wrote:

From my side I do not see any problem with that patch, if someone else has a problem with it please state out the reason.

I have no problem with the patch since it has nothing to do with my DVB card but you're only encouraging Uwe to be abusive since it seems to help get him what he wants:

I've been aware of the problem with dst not fully using the dvb customization systems(*) for a long time. It came up when dvb_attach() et al were first being integrated, well before any rejected patches or strongly worded emails or whatever from certain people that I'm aware of.

I saw some discussion about dst by Markus, Mauro, and Andrew Morton, and since I already know about the issues here, I felt I should post a patch for them any other reasonable developers who might spend time on this.

If there is an abusive person, I'm not going to let it affect my behavior. You lose if you let them influence your decisions one way, or influence them another way.

(*) There are two customization/dependency control systems in DVB. One is dvb_attach(), the other is DVB_FE_CUSTOMISE. They are not two entirely separate systems, but overlap in their design a great deal.

The significant part, common to both, is the overall design of the driver framework. DVB uses what I would describe as an object oriented system. A driver for a certain type of hardware exports a single attach function, which returns an object for one instance of that hardware. All control of that hardware is done via methods defined in this object. There is typical hierarchy, where an 'adapter' object will contain a 'frontend' object which will contain a 'tuner' object. Of course hardware designers are not

Re: [linux-dvb] DST/BT878 module customization (.. was: Critical points about ...)

constrained by the software frameworks we create, so sometimes it's more complex (e.g., dst).

This design means the actual hard link between different drivers is limited to each driver's single attach function (**). By breaking this one link, we can control which drivers must be loaded or linked to only those necessary or wanted. `dvb_attach()` and `DVB_FE_CUSTOMISE` are two different ways of controlling these links.

`dvb_attach()` is based on `symbol_request()`

- A. It's only useful with modules
- B. It doesn't prevent drivers from being compiled
- C. It allows one to build support for hardware, yet not actually load that support until it is needed. This allows supporting a wide array of possible hardware without a large amount of wasted resources, useful for distribution kernels for example.

`DVB_FE_CUSTOMISE` is based on `Kconfig` and static inline stub functions

- A. It works with drivers compiled into the kernel, not using modules.
- B. It prevents drivers from even being compiled in the first place.
- C. Disabled drivers are truly disabled, it is not possible to have support for hardware and yet not load it.

This is useful for the smallest & simplest kernel, for set top boxes and the like. It's entirely possible to use both at once: compile some drivers into your kernel, leave others as modules, not compile modules for hardware you don't have, only load the modules for the hardware you are using at the moment, and still support hardware you might connect later.

(**) The `dvb_pll` module still exports more symbols than just `dvb_pll_attach()`, that is why the customization systems don't fully work with it yet. It also exports `dvb_pll_configure()`, an obsolete interface which only a couple remaining users that have yet been converted. And it exports PLL definition structs, which isn't a difficult problem and I know several ways to fix it, we just haven't decided or actually done it yet.

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to `majordomo@xxxxxxxxxxxxxxxxxx`

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>