

Re: "Generic USB Input Device Support" for keydisks?

Source:

<http://www.tech-archive.net/Archive/WinXP/microsoft.public.windowsxp.embedded/2004-03/0683.html>

From: Slobodan Brcin \((eMVP\) (sbrcin_at_ptt.yu)

Date: 03/12/04

Date: Fri, 12 Mar 2004 22:08:31 +0100

Konstantin,

- > *The Hardware IDs and Compatible IDs come from PnP enum subkeys of*
- > *[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum] (REG_MULTI_SZ values).*
- > *Please correct if I am wrong.*
- > *Do you know how Compatible IDs get in to the registry? I don't think that*
- XP*
- > *PnP sets this info. I think Compatible ID info are in INF files (maybe,*
- not*
- > *always) but INF importer just does not parse it and sets all the PnP ID*
- > *resources to have IsCompatibleID flag set to FALSE.*

This registry entries are virtual anyway. Those values are provided during hardware enumeration by bus driver. Response on IRP_MN_QUERY_ID with parameter BusQueryComptibleIDs.

In most cases value is read directly from hardware. But there are two reasons why inf files usually do not contain this values.

Yes they are read from hardware like any Hardware Ids.

Only difference is when PnP tries to match driver to install them from inf files.

Priority always goes to best match. There are criterions how best match is determined, but it is sufficient so say that all Hardware Id matches come first before Compatible Ids.

Supporting short Compatible Ids would require from driver manufacturer to make driver that will support forward compatibility with all new hardware for many years in future.

Also providing descriptive Hardware Ids enable driver manufacturers to describe each adapter with special text and to populate registry with some special hardware requirements.

- > *However, if for example, you take a look into the MS default monitor*
- > *driver's INF you will see that its Compatible IDs is there (*PNP09FF).*
- >
- > *I am not an expert in INF format so I may be wrong here.*

microsoft.public.windowsxp.embedded: Re: "Generic USB Input Device Support" for keydisks?

You can use compatible ids for dummy drivers or for generic drivers that will support all hardware in given class (with at least satisfactory support).

If inf file with Hardware Id is found it will be always used instead of inf file and driver with compatible id.

There is description in DDK how this works. (But we have no issues with this since we don't want to force hardware manufacturers to change their inf files, and that would be bad solution anyway).

What I want is to define some standard for us developers and possibly MS how to detect drivers that can be configured trough one component and to create one component to cover wide range of devices.

Maybe example of what I'm currently using will help you to understand what I want.

Look at components beginning with "PCI standard *" or "Standard *" you will notice that they all use compatible IDs and that some of them have critical device set to TRUE.

Having components like these your image will allow you to generically support all hardware types. Some of these components are just placeholders to set some info in critical database and some offer driver functionality trough files like pci.sys pciide.sys, atapi.sys, etc.

This will allow us to reach FBA PnP phase and as you have noticed if we just copy new drivers and inf files they will override generic drivers (which btw are used by Intel etc with only name modification from inf file same MS binaries are used) that override some of this functionality.

Check component: PCI standard RAM Controller. This hardware resource does not use any driver it just exist as hardware resource access point. Other manufacturers just change name of this node trough inf file. They do this by providing hardware Id that always have better match than compatible Id provided by MS.

Bottom line is if we use few MS components to make system boot, everything else will be handled by PnP.

So we need a way to create component from let us say Intel 845.inf, 865.inf and 852.inf files.

Ok we can have 6–7 components but not 15–20 components. Also all that components should know that they are using functionality (generic drivers provided by MS and not third party) and according to these knowledge CD could create component that will use compatible ID that we see in component "PCI standard RAM Controller".

If we had these options then it could be possible for MS to improve TD so we could sort components in central panel trough their meaning chipset drivers of some type, or drivers generally, etc... (topic for another discussion)

Uhhh... I hope that I explained more successfully what I want.

Re: "Generic USB Input Device Support" for keydisks?

microsoft.public.windowsxp.embedded: Re: "Generic USB Input Device Support" for keydisks?

Regards,
Slobodan

~~~~~  
Have an opinion on the effectiveness of Microsoft Embedded newsgroups? Tell  
Microsoft!

<https://www.windowsemergedeal.com/community/newsgroups>

~~~~~  
> >...>
> >
> > > ... > >
> >
> > *Fortunately compatible ids are not required for hardware display driver
> and
> > network category. This category does not require us to provide any
> resources
> > like service initialization or some registry entries.
> > Since during installation PnP will use all required info from inf file
> itself.*
>
> *I see.*
>
> > *Merging of all sections from inf file could be even dangerous, since
there
> > is possibility that some hardware require some specific service
installed
> > while some other does not require it.*
>
> *That is why I was referring to having "simple" and "complex" INF import
> processes. This may always be up to dev (XPe, manufacturer) to decide
> whether it is dangerous to merge the components. We have to test images
> anyway.*
>
> > *So for this type of drivers only file copy should be enough and auto
> > detection from PMQ file would be nice.*
> >
> > *Also if you use different manufacturers in same image like ATI and
NVIDIA.
> > It is not good to set resolution from driver components.
> > In my image I'm using directly prototype component "Device: Display" it
> > allows me to set resolution independently of drivers I choose to put in
my
> > image. Also this component should support settings for more that one
> monitor.*
>
> *Yup, if you support more than one platform – prototype "Device: Display"
is
> a good choice.
> However, how do you want to tell TAP to choose the "Device: Display"
> automatically and not to match the real display driver PnP with an XPe
> database component?*

Re: "Generic USB Input Device Support" for keydisks?

microsoft.public.windowsxp.embedded: Re: "Generic USB Input Device Support" for keydisks?

>
>>> ... >> *trying to avoid it? If I understand this, I will understand your
> idea.*
>> *I'm following the logic "The less things you have in registry the less
>> things can go wrong". And who knows it might speed boot process for few
> mil. sec :D*
>
> *Agree. It also simplifies the image build process.*
>
>>> ... >>
>
>> *Let me think.*
>> *1. It would be nice to have native mechanism to create components like
> this for our purposes, right?*
>
> *Sure. A custom INF parser should do the trick if we come to agreement on
the*
> *Compatible IDs.*
>
>> *2. MS could make few new components if they have time to spare. (keeping
> old for compatibility)*
>
> *So, you don't change existing XPe database. Then a few multiply TAP
results*
> *used in one configuration may not work properly with your concept.*
>
>> *3. Big picture. In few years we will have LHe. Do you think that they
> should keep compatibility with XPe components and way to componentized
> drivers?*
>
> *As I heard LH (not embedded) is going to be much more componentized in
first*
> *place than, say, XP Pro.*
> *Basically, there may be LHe only and LH will be built as one of the
possible*
> *configurations (the biggest one!).*
> *However, these rumors seem mostly apply to the software stack. I don't
know*
> *MS plans for the hardware stack in XPe, though. I don't think it is
possible*
> *to change the hardware support dramatically maintaining compability to all
> the hardware supoprted by XP Pro now.*
> *My personal vote would be to re-design current XPe database (not only
around*
> *the hardware categories, btw).*
>
> *Regards,*
> *Konstantin*
>
>

Re: "Generic USB Input Device Support" for keydisks?