

device-handler

Marius Gröger

Copyright © CopyrightÂ©1995 by Marius Gröger

COLLABORATORS

	TITLE : device-handler		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Marius Gröger	August 3, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	device-handler	1
1.1	Device Handler	1
1.2	Device Handler - WARNING	1
1.3	Device-Handler - Feaures	2
1.4	Device Handler - Requirements	2
1.5	Device Handler - Motivation	2
1.6	Device Handler - Installation	3
1.7	Device Handler - Usage / Stream names	3
1.8	Device Handler - Caveats	4
1.9	Device Handler - Future plans	6
1.10	Device Handler - History	6
1.11	Device Handler - Copyright / Author	7
1.12	Device Handler - Number conversion	8
1.13	Device Handler - Stream name conversion	8

Chapter 1

device-handler

1.1 Device Handler

Raw device-handler - Written by Marius Gröger.

This software requires AmigaOS 2.0 (V37) or higher.

WARNING - PLEASE READ THIS!

Features

Requirements

Motivation

Installation

Usage / Stream names

Caveats

Future plans

History

Copyright / Author

1.2 Device Handler - WARNING

This software is intended for advanced users only. I strongly recommend all people who do not understand every detail given herein to obtain further information about AmigaDOS-Devices, Handlers and their implementation on top of Exec device drivers.

WRONG USAGE OF THIS SOFTWARE WILL MOST LIKELY
RESULT IN SEVERE LOSS OF DATA!

This software is freeware. It is provided as-is and is subject to change; no warranties are made. All use is at your own risk. No liability or responsibility is assumed.

1.3 Device-Handler - Features

Features

- ~Easy access to any block oriented device
- ~Comfortable stream name specification
- ~Completely pure code
- ~Fast operation due to improved buffer management
- All `OpenDevice()` parameters to a device, i.e. name, unit and flags are derived from the AmigaDOS-Handler, releasing you from dealing with it and thus making the usage much safer!
- ~Distributed as Freeware, complete source code is supplied

1.4 Device Handler - Requirements

Requirements

This package requires AmigaOS Release 2.04 or higher. However, the supplied Installer script assumes an AmigaOS Release 2.1, and tries to install a `DOSDriver` file in `DEVS:DOSDrivers`.

I don't suppose that there are potential users out there running 2.0 who want to use the Device-Handler. If I'm wrong, though, it shouldn't be a major problem for such people to create a `Mountlist` entry themselves.

There are no further requirements than to have an Amiga computer with at least one block oriented device attached. Your built-in floppy will do.

1.5 Device Handler - Motivation

Motivation

Amiga users sometimes glance enviously towards the UNIX world, where - as a result of the system-concept - raw device access on shell level is very easy.

Several solutions to implement this for Amiga computers have been provided in the past. However, none of them fully met my personal needs, so I decided to do my own approach. The early releases of this package, which appeared also in the Aminet, were in some detail derived from the solution of Matthias Scheeler, which is also provided with source code and which gave me quite a good start. The previous versions therefore also contained the original archive of Matthias. However, the current version has no more similarities to Matthias's handler, so I will no longer include his code.

For my own solution I wanted a handler which didn't need the (currently)

floppy-specific TD_GETGEOMETRY-IORrequest to obtain the physical device layout. That is important, because unfortunately a lot of harddisk- (and other) drivers don't implement this command.

So I decided to derive the Exec-device selection from on the AmigaDOS stream name in such way, that the user only provides the name of an AmigaDOS-Handler instead of the Exec-device.

With the new handler, you are able to refer to the raw device's byte stream of the built-in floppy drive just typing "DEV:df0". Further improvements include the ability to limit the total size which is allowed to read or write, and to specify the low- and high cylinders to be used.

Further benefits are a noticably higher speed due to a optimized buffer handling, more control on which sections of a device are involved and a much nicer source code :-)

1.6 Device Handler - Installation

Installation

It is best to use the supplied installer script in order to install the device-handler properly. If you still want to do the installation manually then follow the steps listed below.

As usual for any AmigaDOS-Handler, the actual handler code is best copied to "L:":

```
AmigaShell> copy device-handler l:
```

The mount specification entry may be copied to "DEVS:DOSDrivers" in order to have it always mounted during system startup:

```
AmigaShell> copy DEV DEV.info DEVS:DOSDrivers
```

Copy it to "SYS:Storage/DOSDrivers", if you want to use it more rarely:

```
AmigaShell> copy DEV DEV.info SYS:Storage/DOSDrivers
```

This way you will have to mount it manually in order to use it, though.

Note that the DEV icon has set the tooltype 'ACTIVATE' set to 0, which tells AmigaDOS to defer the actual loading of the handler-code until the first usage. Therefore you only loose some memory for the AmigaDOS-internal representation of the DOS-driver, if you don't use it.

1.7 Device Handler - Usage / Stream names

Usage and the construction of stream names

When using the device-handler, you have to construct a so-called

stream-name which contains all the information needed to access the raw device in question.

The stream name is constructed as following:

```
DEV:<DOSHANDLER>[/<MAXLEN>] [/LOCYL=<NUM>] [/HICYL=<NUM>]
```

First, a Stream name conversion will be done.

Next, the following ReadArgs() template is applied the the stream name.

```
DOSHANDLER/A,MAXLEN,LOCYL/K,HICYL/K
```

DOSNAME

A valid, AmigaDOS-device name such as DH0, df0, ... The device has to be mounted, though.

MAXLEN

optional: maximum length of bytes to read or write. The (final) lo/hicyl may enclose a smaller space: the smallest space is always used. This value must appear anywhere behind DOSHANDLER. Number conversion is applied.

LOCYL

Starting cylinder read from/write to. If higher than effective high cylinder (i.e. after evaluating HICYL), low cylinder is casted to high cylinder. Number conversion is applied.

HICYL

Ending cylinder. There are no assumptions made on the plausibility of this parameter, because the actual physical ending cylinder of the underlying device does not have to be the one of the handler specified by DOSNAME. Number conversion is applied.

Some example stream names, which are all valid (but which do not mean the same, though)

```
DEV:df0
DEV:dh0/1024
DEV:dh0/1k
DEV:dh0/LOCYL=0
DEV:dh1/HICYL=1k/LOCYL=0M/20M
DEV:df0/$1000
DEV:df0/0x2000/LOCYL.4
"DEV:df0/0X200/LOCYL=4"
```

Please read also the Caveats section.

1.8 Device Handler - Caveats

Caveats

This section tries to address some problems which can arise due to,

peculiarities of certain storage devices, the usage in conjunction with other pieces of software or other circumstances.

I will always try to give the user the necessary technical background to understand the problem and, if possible, to include a work-around.

If any user has found a similar problem which is not mentioned in this document, he or she should contact me in order to append it to this section.

device-handler and Harddisks: the RigidDiskBlock problem

If you intend to use DEV: for UNIX tar, especially for transferring software between systems, you will have to start writing the media exactly at block 0. For floppy-based devices this will be correct automatically. In contrast, nearly every harddisk maintains in the very first sectors the so-called RigidDiskBlock (RDB), which contains the partitioning information. Therefore, the actual partitions are located in the blocks following the RDB area. The first partition has usually a low-cylinder of about 3. To achieve the intended behaviour (i.e. to start reading or writing exactly at block 0), you should create a special DOSDriver/mountlist entry, such as:

```
RDH0:                                /* raw dh0 */
    Device = scsi.device
    Unit = 1
    Flags = 0
    Surfaces = 5
    BlocksPerTrack = 17
    LowCyl = 0 ;                      /* specify here full range */
    HighCyl = 1023                    /* of cylinders */
    Stacksize = 4000
    Priority = 5
    GlobVec = -1
    BufMemType = 0
#
```

Most fields will have to be adapted for your specific host adapted. Note that this includes the RDB area, and will destroy any partitioning!

Another way is to use the LOCYL stream name option. Again note: playing around with LOCYL or HICYL is a pretty dangerous thing. It is always advised to write a dedicated mountlist/DOSDriver.

Using the device-handler with MultiFileSystem

If you have installed Nicola Salmoria's excellent MultiFileSystem (MFS) (don't confuse with the MultiUserFileSystem or the former East-German secret service), there can arise quite a subtle problem.

The MFS provides an automatic file system detection, which is configurable by a supplied commodity program. If you are using a multi-volume archiver such as tar, and you are not sure if all your disks are the same format (i.e. Amiga or MS-DOS/MAC), you should switch from "Automatic" to any specific file system. It is of no matter which one you use, simply decide for one. This decision will

only influence the storage capacity.

Technical background: Tar, for instance, seems to determine the disk size once at startup-time. However, the device-handler is opened and closed for each volume. If you insert differently formatted disks, MFS will switch between them and will provide the device-handler one time with the trackdisk.device, and the other time the mfm.device. Tar, unfortunately, will always assume a constant capacity, and so either some blocks will be left unwritten, or free storage space will get rather thin in the end...

General: choose your raw-device's names carefully

It is always dangerous to use the device-handler. To reduce the risk of providing the handler with a wrong device, try to give the DOSDrivers meaningful names, which tell you at one glance much information:

- the corresponding SCSI address or floppy drive number
- a prefix 'R' identifies it to be a raw device
- capacity
- initials of manufacturer

Probably even the example entry "RDH0" is not enough descriptive. A nice name is "rSQ270.4", telling that you are dealing with the raw device of your SyQuest 270 meg drive at SCSI address #4.

A good idea is also to keep a copy of the RDB area of all your harddisks at a safe place - together with the corresponding RDB-recover utility.

1.9 Device Handler - Future plans

Future plans

Perhaps I will implement some more features:

- support for directory examination packets and pseudo file locks, so on could CD to DEV: and issue Dir. This could print out the list of currently usable block devices.
- ~Startup entry in mount descriptor could contain a default stream name
- A acception/rejection pattern to exclude certain handlers (e.g. "REJECT=(DH|HD)[0-9] ACCEPT=R#?.[0-9]")

If you have any suggestions, contact me.

1.10 Device Handler - History

Development history of "device-handler"

Release 37.1

- ~bumped to version 37 to reflect the os requirement
- added AmigaGuide® documentation
- added Installer Script
- ~optimized versions for different CPU types
- Original Archive is no longer includes as the similarities are completely vanished
- LOCYL option was broken and used always #0
- HICYL is now always taken as it comes from the user; it is no longer compared with the AmigaDOS device entry as it is intended as an option to override this value.

Release 1.5

- fixed bug which prevented multi-volume tar archives to be handled correctly

Release 1.4

- ~first public release
- stream options LOCYL, HICLY, MAXLENGTH

Technical history of "device-handler"

For more technical details, please refer to the source code. In the header section of each source file, you find the pseudo-keyword \$HISTORY: after which follows the complete revision history of this file.

1.11 Device Handler - Copyright / Author

Copyright and author information

This software is freeware. It is provided as-is and is subject to change; no warranties are made. All use is at your own risk. No liability or responsibility is assumed.

It's strictly forbidden to include this archive in any kind of software collection except Fred Fish's AmigaLibrary, Aminet, Aminet CD's and BBS fileareas.

You may make any changes to the source for your own use. If you consider them useful for everybody, please get in touch with me. This way I can try to include them in the next public release.

If you like this software, or have any suggestion how to improve it, or just want to complain about it, feel free to email me.

Home address:

Marius Gröger,
Bärstadter Str. 4
65307 Bad Schwalbach (GERMANY)

Internet email addresses:

mag@sysgo.de
409@informatik.fh-wiesbaden.de

1.12 Device Handler - Number conversion

Number conversion

Any number which is to pass to the handler may start with '\$' or '0x' or '0X' to indicate hexadecimal notation (default is decimal). A trailing 'M' tells the handler to interpret it as a mega-value, (i.e. multiply by 1024^2) and 'K' to signal a kilo value (factor 1024). All characters are case-insensitive.

Examples

1k, 1K, \$400, 0x400, 0X400	mean all 1 KB = 1024 Byte
1M, 1024K	mean all 1 MB = 1024 KB = 1048576 Byte

1.13 Device Handler - Stream name conversion

Stream name conversion

The following steps are performed during stream name conversion:

1. All slashes are removed and replaced by blanks.
2. All dots ('.'), that appear after the first slash and which are not part of a quoted text, are substituted by an equation symbol ('='). That comes in handy when using the LOCYL or HICYL options. AmigaDOS strips all equations that are not quoted. Thus, the command "type dev:df0/locyl=3" would lead to an error, as the handler would get the text "dev:df0/locyl" instead of "dev:df0/locyl=3". So, either use the dot or quote the whole stream specification.