February 24, 2007
Alexander Noé
alex@alexander-noe.com

Contents

1 Introduction

This section is not relevant to the usage of *PxScan* or *PxView*, it merely explains why I actually started to work on those tools and why I still work on these tools.

1.1 In the beginning...[1]

After having done my first review of a DVD writer, I realized how much I missed the possibility to run several tests on the same disc in a row, without user interaction, with PlexTools[2]. Also, some badly recorded discs showed error rates around 10 in one section, and around 1000 in another section (cf. Figure 1), requiring a logarithmic scale in order to properly visualize the result. While KProbe[3], which can be used in order to scan CDs and DVDs using drives equipped with a MediaTek chipset, has offered a logarithmic scale from the beginning, Plextor has not yet implemented this. I didn't like either that discs with a
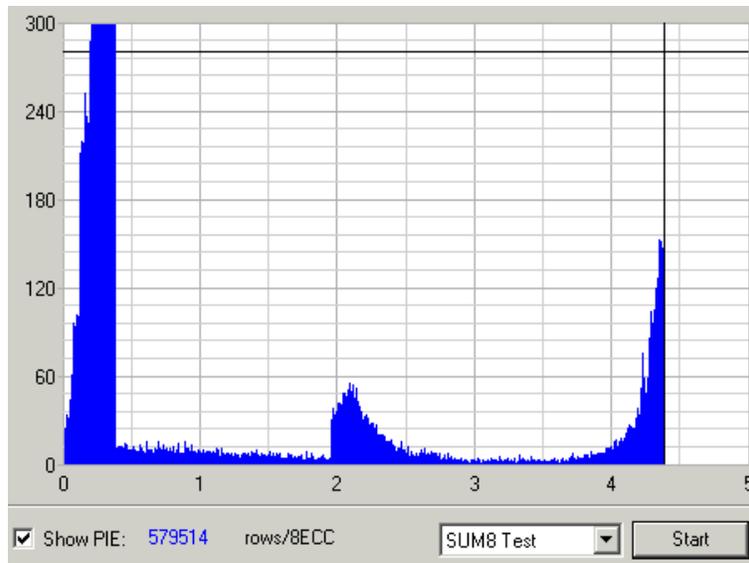
Figure 1: `PIE` scan with PlexTools Professional v2.17 of a bad recording

few evenly distributed errors usually had a lot more blue than discs with error spikes and more errors in total. This is explained in detail in section 4.1.

---

[1]Note that this title does not accidently coincide with a Babylon 5 title

[2]I'll refer to PlexTools Professional by simply 'PlexTools' for simplicity. This feature was first introduced in version 3.02 of PlexTools Professional XL long after *PxScan* allowed this

[3]http://www.k-probe.com/

So I decided to start to work on my own scanning application. The article on cdrlabs[4] about ASPI-programming[5] is worth being mentioned in this context, since this was how I started to learn how to use ASPI only a few weeks before I started to work on *PxScan*.

Another important point was that such an application would make it very easy for developers interested in operating systems besides Windows to develop a scanning tool working for example on Linux. In the meantime, this has been accomplished[6].

1.2   After the beginning...

First, I want to say that, considering how easy it was to conclude the commands, and to find out how to use them, using a simple protocol showing which commands PlexTools sent in order to initiate scans, I had absolutely no idea why no one tried (and accomplished) this before.

Only on May 29, 2005 did I learn why: Appearently, other authors of scanning software have asked Plextor before doing such a thing and did not continue after having got (or not having got) an answer. Although I knew through inofficial communication channels that Plextor's parent company Shinano Kenshi had not exactly been a fan of the idea that scanning software other than PlexTools existed for their drives, I did not really believe that they would be going to send a cease-and-desist, even less one with a couple of spelling errors, factually simple wrong accusations, and accusations which are no infringement of applicable law.

1.3   Some notes on those allegations

After Plextor sent this cease-and-desist, discussions started all over about whether or not Plextor's claims were justified. I will shortly explain why all accusations are either factually wrong or do not constitute an infringement of applicable law.

1.3.1   Unfair commercial practices

Although 'unfair commercial practices' could constitute an infringement of applicable law, there somehow needs to be commerce involved. Thus, i'm not sure what this allegation

---

[4]http://www.cdrlabs.com

[5]http://cdrlabs.com/articles/index.php?articleid=3

[6]https://sourceforge.net/projects/qpxtool/

was based upon.

1.3.2   Violation of the contract between me and Plextor

I have not signed a contract with Plextor.

1.3.3   Reverse-engineering Plextor's software

Looking at data sent or received by software or hardware in order to understand how it works is perfectly legal in Germany, creating software that contains functions other already existing software already contains is also perfectly legal. While you could argue which word you have to use in order to appropriately describe these actions, whatever you call it, it does not constitute an infringement of applicable law.

1.3.4   Damage Plextor's good name and reputation

Look at the following thread on cdfreaks which is a collection of quality scans made by a lot of forum members: `http://club.cdfreaks.com/showthread.php?t=115081`. Look for scans of the PX-716 model with early firmwares. Note, in this context, that Plextor Europe had delayed the launch of this drive due to quality problems[7] : Plextor Europe did not consider the drives responsible for those results [8] good enough for sale. Consequently, anyone who said the early models of that drive were not good enough for the price they cost merely repeated what Plextor Europe said and confirmed it by using means Plextor provided by including those scanning functions.

1.3.5   I said *PxScan/PxView* were a complete and full alternative for PlexTools

Either Plextor's parent company's legal departement sucks terribly at english, or they intentionally added an accusation to their letter that is factually so wrong that I wasn't sure if I should laugh or cry.

Correct is that the PxLinux sourceforge project was stating it was supposed to become a complete alternative for PlexTools running on Linux one day. I did not write one single line about this project on sourceforge. Thus

---

[7]`http://www.heise.de/newsticker/result.xhtml?url=/newsticker/meldung/52869&words=`
`Plextor`

[8]`http://club.cdfreaks.com/showpost.php?p=753383&postcount=7`

- no one ever claimed that anything were a *true and full alternative*[9] for PlexTools
- the only claim that somehow had the words 'full' or 'complete' as well as 'PlexTools' and 'alternative' in it was about PxLinux, only about an operating system Plextor doesn't offer any scanning software for and only referred to a distant future.

As you can see, this accusation is either the result of malicious intentions (they knew it was wrong) or of arrogance (they didn't consult an interpreter in order to understand who said what although it would have been necessary).

1.4   Reactions

1.4.1   Administrators on sourceforge

Sourceforge decided that Plextor has to file a DCMA complaint if Plextor wants that those administrators do not ignore this claim.

1.4.2   CDRlabs

CDRLabs published the e-mail they received from Plextor's attorneys, stating that they did not understand what this was supposed to be about and that they would not obey to the demand to stop me from offering my software on the forum. Here, the news message[10]:

Monday, May 30th, 2005
Plextor Takes On PxScan And PxLinux
4:25 PM CST   **Posted By: Ian**

As many of you already know, Plextor's parent company, Shinano Kenshi, has sent letters to the developers of PxScan and PxLinux, accusing them of any number of trademark and copyright violations. For whatever reason, their lawyers have dragged CDRLabs into this mess, ordering us to stop the developers from using our forum to promote the "illegal versions" of their software.

> *Our clients have established that illegal copy versions of their software are now available under GPL license and under LINUX under the name PXSCAN and PXVIEW.*

> *These versions are released by Mr. Alexander Noe (GPL) and Mr. Eric Fernandez (acting under the name Zeb) (LINUX version) . We have noticed that both persons use your internet site and forum in order to promote the illegal versions of our clients software.*

> *Please confirm that you will take all measures in order to prevent all future use of your medium infringing my clients rights.*

From what I can gather, Shinano Kenshi is afraid that people won't buy their PlexTools XL software. No matter what way you slice it though, this does not put Shinano Kenshi, and in turn Plextor, in a good light.

We have yet to hear back from Plextor America or get a good explanation as to why PxScan and PxLinux are "illegal". Until we do, CDRLabs is not planning on taking any action. If you'd like to read more about the situation, join the ongoing discussion in our forum or the one over at CD Freaks.
[discuss this in the forum]

---

[9]citation from their e-mail

[10]http://www.cdrlabs.com/news/archive.php?my=1114920000

1.5   Problems with discussions about legal stuff

Some people who tried to discuss this issue on various forums turned out to use words like 'decompilation' without actually knowing what decompilation was. This section shows what that can lead to.

1.5.1   People who got attacked on their own

One user on hardware.fr claimed that what I did was illegal because the company he was working for got itself attacked by Adobe. He even claimed the makers of LaTeX had special contracts with Adobe allowing them to create PDF files (the company was attacked for offering a tool that created PDF files) because he couldn't stand the idea that the attack on *PxScan* was as unjustified as the attack his company saw itself confronted with might have been.

1.5.2   Random collections of words

On computerbase forums, a discussion started. The worst result of all was a sentence which would roughly translate to "so the reverse-engineering has been decompiled". If anyone understands that sentence, please explain it to me, since I don't.

1.5.3   Forum trolls

Note: This section contains sarcasm.

Some people claim that a Fiat was a knockoff of a BMW: Fiat obviously stole the idea to use four tyres, thus, you cannot tell the difference between those two easily, also, Fiat is known to copy the layout of their cars so that potential customers could mistake a Fiat for a BMW at first glance. Since a Fiat is cheaper than a BMW, people buy Fiats and then are proud of having obtained a really cheap BMW.

Those people then conclude that *PxScan* must be a knockoff of PlexTools: As you can see in figure 1 and 3, *PxScan/PxView* and PlexTools are just as easy to confuse as a Fiat and a BMW. Thanks to `http://club.cdfreaks.com/showpost.php?p=1251760&postcount=25` for the Fiat-BMW-idea.

1.6 Current state

In December 2005, it has been confirmed to me that Plextor continues to make unjustified accusations and defamatory claims, namely that *PxScan* was illegal software. Also, a representative of Plextor Europe stated that Plextor did not have the resources required to support Linux[11].

Although no one can say for sure whether or not this is true, it is, in connection with this issue, a bogus statement. No one is demanding that Plextor create Linux software using their own resources. Also, this statement is not an answer to the question why Plextor does not allow independant third party developers to do for free what Plextor lacks, according to this representative, the necessary resources for. Thus, the only way to interpret this statement is the following: Plextor does not have the resources required to completely ignore or to tolerate the existence of third party software allowing to use Q-Check features Linux.

I have learned that one point seems to be that Plextor does not want scans that have been performed at high read speeds to be published. However, since you can easily change the read speed of PlexTools by using Nero DriveSpeed directly after initiating the scan, without PlexTools doing anything about it, this cannot be meant too serious.

---

[11]http://www.pro-linux.de/news/2006/9151.html

2 Measurements on CDs and DVDs

This section explains what kind of errors and properties of a CD or DVD can be measured using *PxScan* or similar software. It does not explain how to read a result, how to judge whether a disc is good or bad: This question will be dealt with in section 4.

Note in advance that error rates, which are explained below, measured after recording a disc don't tell anything about the lifespan of a CD/DVD. If you want to know if your disc will still work in a year, scan it once a month and check if the error rates are increasing.

2.1 Errors on CDs

When scanning a CD for errors, you get several error values: `C1`, `C2`, `CU`[12]. In order to understand what those numbers mean and how to judge whether or not a CD has been recorded at good quality, you need to understand, at least a little bit, how data on a CD is arranged and protected against damage.

A CD consists of sectors of 2352 bytes each. A sector is split into 98 groups of 24 bytes each. For each of those groups 4 bytes are calculated and used for `C2` error correction. Then, after moving around some data, 4 more bytes are calculated for each group of 28 bytes and are used for C1 error correction.

When reading a CD, there are 32 bytes ($24 + 4 + 4$) in each of the 98 groups. If one bytes is damaged, you have an E11 error, if 2 bytes are damaged, you have an E21 error, if more than 2 bytes are damaged, you have an E31 error. E11 and E21 errors are guaranteed to be corrected using C1 error correction only. If there are more damaged bytes, the result after doing this 'moving around' backwards is that those errors are then distributed over several groups of 28 bytes each, hopefully with not more than 2 errors per group. One error in such a group is called an `E12` error, 2 errors are called an `E22` error (also called `C2` error), more than 2 errors are called an `E32` error (also called `CU` error).

On audio CDs, an `E32` error means that error concealing techniques might be necessary in order to minimize the risk of ticks and clicks on playback. Thus, an audio CD that contains `E32` errors could be damaged beyond repair (the original data cannot be retrieved anymore) and should immediately be backuped. A CD containing `E22` errors but not `E32`

---

[12]Plextor drives with Q-Check function are able to report `E11`, `E21` and `E31` seperately, however, this is not yet implemented in *PxScan*

errors is not yet damaged beyond repair, but you might want to make a backup before `E32` errors show up. On data CDs, there is a third layer of error correction (that's why you only have 2048 bytes of data in one sector of a data CD), so that a certain amount of `E32` errors can still be corrected.

On CDs, error measurement is usually performed over an interval of one second, which equals 75 sectors. Very good discs can show an error rate below one `C1` error per second and no `C2/CU` errors.

If you want to know more about error correction on CDs, or CDs in general, have a look at ECMA-130[13].

2.2   Errors on DVDs

On DVDs, error correction is performed on blocks of 16 logical sectors, i.e. on blocks of 32 kB. One such block is called a physical sector. Those 16 logical sectors of 2048 bytes each are arranged in a 4x4 matrix (where a sector is a matrix itself), then error correction data is calculated line-wise and column-wise. In the end, the result is a $208x182$ matrix.

When reading a physical sector, you have

- a **PIE**, parity inner error, when one line is damaged (not more than 4 errors)

- a **PIF**, parity inner failure, when more than 4 errors occur in one line

- a **POE**, parity outer error, when data in one column is damaged. PlexTools report an error type it calls POE, however, whatever it reports, it seems to be neither the number of bytes damaged after applying PI error correction nor the number of columns.

- a **POF**, parity outer failure, when data in one column cannot be recovered.

`PIE` is usually counted over groups of 8 physical sectors, `PIF` and `POF` is counted over physical sectors. A DVD that has 1 or more `POF`s is damaged beyond repair. 8 consecutive physical sectors should not have more than 280 `PIE`s, one physical sector should not have more than 4 `PIF`s. Since each row can at most contribute one `PIE` and one `PIF`, The worst possible values are:

- PIE: 1664
- PIF: 208

---

[13]http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-130.pdf

If you want to know more about error correction on DVDs, look at ECMA-267[14].

2.3   Jitter and Asymmetrie

Data on CDs and DVDs is encoded using pits and lands. The information is contained in the length between 2 transitions between pits and lengths. Those transitions can only have predefined lengths, as explained in section 2.4. Jitter measures how much those transitions lengths deviate from the lengths they were supposed to have. Unfortunately, Plextor drives don't allow a measurement in units like per cent, they only report some absolute values which can be considerd 'higher' or 'lower' than other values.

Asymmetrie is an indication how well pits and lands are in balance.

2.4   TA

Although this could theoretically be checked for CDs and DVDs, current Plextor drives only allow to run this test on DVDs[15].

The length between 2 transitions resembles one piece of information. Those transitions should have a length of 3, 4, 5, ..., 10, 11 or 14 on DVDs. When running a TA test, the drive returns a couple of thousands of such lengths. Then, it is up to the scanning software to build a histogram and calculate statistically interesting values, like root mean square error.

When running a `TA` test using *PxScan*, a few hundred thousand lengths will be used per test zone to build the histogram. Last time I checked, PlexTools at least queried the same number of transition lengths, however, since the it is not open-source software, no one besides Plextor can really say what it is actually doing.

---

[14]http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-267.pdf

[15]Trying to run it on CDs crashes the drive's firmware. *PxScan* prevents the user from trying.

3   *PxScan*

This section explains how to use *PxScan* in order to perform disc quality tests with Plextor hardware.

3.1   Getting started

If you have not used *PxScan* before, you should simply download the full package.
If you have downloaded and unzipped *PxScan* only, and not the full package, you need an ASPI layer. An ASPI layer is one possible way for applications to communicate with drives. There are several ASPI layers available, however, I did not verify correct behaviour with all of them. The following ASPI layers are supposed to work with *PxScan*:

- *Frog ASPI*[16]
- *Nero ASPI*[17]

Adaptec ASPI is NOT recommended to be used in connection with *PxScan*, since USB2/Firewire drives are often not recognised, sometimes no drive at all is found. I do not know whether *PxScan* or *Adaptec ASPI* is the problem here, all I can say is that the combination of those two does not work properly. Also note that *Frog ASPI* does not work on Windows 9x/ME, so you cannot use it if you are using such an operating system.

PxScan expects the filename of the ASPI layer to be either `WNASPI32.DLL` or `FrogASPI.DLL`. Small or capital letters do not matter on Windows.

Updates for *PxScan* do not include an ASPI layer. The full package contains, with the author's permission, FrogASPI.

3.2   Getting a list of drives

If you need to get a list of all drives *PxScan* finds, launch the command line interface of Windows (for example by launching the bundled `cmd.bat` file) and type `pxscan`. You will get a list of all drives *PxScan* could find. For drives containing the string `'PX-7'` or `'PREMIUM'` the serial number will be displayed.

The reason why *PxScan* does not display the serial number for other drives is simple:

---

[16]`http://www.frogaspi.com/`

[17]`http://www.nero.com/nero6/en/WNASPI32.DLL.html`

```
PxScan v1.7.7, built Jan 19 2006 23:06:59

The following drives have been found:
  #0 at 0:0:0: ASUS      DRW-1608P2S      1.37 ATAPI
  #1 at 1:0:0: PLEXTOR   DVDR    PX-712A   1.07 ATAPI S/N 229930 TLA #0102
  #2 at 1:1:0: TOSHIBA   DVD-ROM SD-M1712 J004 ATAPI
  #3 at 4:0:0: HA1133Q   FCA891D          1.0  SCSI
  #4 at 4:1:0: HA1133Q   FCA891D          1.0  SCSI
  #5 at 5:0:0: PLEXTOR   DVDR    PX-716A   1.09 ATAPI S/N 153669 TLA #0202
  #6 at 5:1:0: HL-DT-ST DVDRAM GSA-4120B A117 ATAPI

Use  pxscan help  to obtain a list of options
```

Figure 2: *PxScan* - query drive list

The PX-130 drive, when using firmware 1.02, crashes *PxScan* when attempting to read its serial number and takes the OS with it, meaning that the system must be rebooted. In order to avoid such incidents, only the serial number of drives that are likely to be scanning drives is display. This is helpful since *PxScan* allows the user to identify a drive by its serial number: Create a `.bat` file that launches *PxScan* the way you want once, and by identifying the drive using its serial number you avoid that this file must be changed when the drive is connected to another port, for example changed from USB2 to Firewire.

If USB2 or Firewire drives are not found, then the reason is probably that you are using Adaptec ASPI.

3.3   Using *PxScan*

Once showing the list of drives succeeds, you can initiate a scan. Initiating a scan requires you to tell *PxScan* which drive to use (or let *PxScan* select the drive automatically), which scans to perform, which speed to perform those scans at, which file the result should be saved to and more. At first, a simple example:

```
pxscan pie speed=12
```

This means *PxScan* will look for a drive that can scan and that has a disc mounted, will run a PIE test (also called PiSum8) at 12x speed and save the result to a filename that is created automatically. If you want to specify a file name, refer to section 3.3.8. Note that scanning should be done at 2x speed if you want to be able to draw conclusions from a bad scan result. The fast scan for which the command is shown above rather works the other way around: if the result is good even at 12x scanning speed, the disc is pretty good.

Figure 3: *PxScan* performing a scan

3.3.1 Selecting a drive

There are 3 ways to select a drive:

- `drivenbr=n` selects the n-th drive from the list above
- `serial=s` selects the first drive of which the serial number contains the string s
- Have it selected automatically

If you select a drive that does not exist, for example if you select drive #17 in a system with less than 18 drives installed, *PxScan* will abort. If no media is mounted in the drive you selected, *PxScan* will wait up to one hour. If the drive is not ready after that, *PxScan* will abort.

If you select no drive at all, but select tests to perform, the first drive meeting the following requirements will be selected:

- The drive is capable of Q-Check
- A disc is mounted in the drive
- the drive is not already used by another instance of PxScan (you can scan two discs at a time if you own two drives capable of Q-Check)

3.3.2 Supported drives

Only Plextor drives supporting Q-Check can be used. At the moment, the following drives do support Q-Check features:

- `Premium`
- `PX-708A2`
- `PX-712`
- `PX-714`
- `PX-716`
- `PX-755` NOT fully supported by *PxScan*
- `PX-760` NOT fully supported by *PxScan*

The reason why the `PX-755/760` is not supported is the following: This drive features 'protected commands'. Such commands include Q-Check, and can only be executed after doing the following steps:

1. query a 16 byte challenge code from the drive
2. calculating the correct response and set it

Step 2 is a problem, since the way it is done is of course not publicly known.

As you can see, Plextor is intentionally making it hard to allow other applications than PlexTools to perform quality checks or enable or disable some of the other special features. Firmwares up to 1.01 (`PX-755`) / up to 1.00 (`PX-760`) protect at least the following commands:

- some Q-check features (FE/TE test does not seem to be protected)
- write simulation on DVD+R media
- SilentMode
- GigaREC
- Bitsetting
- Hide CD-R
- Single-Session

Beginning with firmware 1.02 (`PX-755`) / firmware 1.01 (`PX-760`), the bitsetting commands are no longer protected, meaning that Plextor finally decided to allow applications like Nero Burning ROM, Nero CD-DVD Speed or ImgBurn[18] to change the bitsetting configuration of the drive.

---

[18]`http://www.imgburn.com`

Note that `PX-714` is equal to `PX-716` and that `PX-708A2` is equal to `PX-712`. These drives being equal to drives that do support Q-Check, they also do support Q-Check. Whereas PlexTools doesn't allow to run Q-Check tests on `PX-714` and `PX-708A2`, *PxScan* does.

3.3.3 Selecting tests

On CDs, you can run the following tests:

- `CX`: Perform a `C1/C2/CU` test.
- `JB`: Perform a `jitter/beta` test. This test is run at 4x speed

On DVDs, you can run the following tests:

- `PIE`: Perform a `PIE` test.
- `PIF`: Perform a `PIF` test.
- `JB`: Perform a `jitter/beta` test. This test is run at 2x speed
- `TA`: Perform a `TA` test[19]. You can also indicate the positions (up to 6) where to run the test using the following syntax:

  `TA=100MB,500MB,1300MB,3500MB,4000MB,4480MB`

  Do not inculde any additional spaces. Note that this test can only be run on recorded DVD media, but not on DVD-ROM. Since attempting to run this test on a DVD-ROM can crash the drive's firmware, *PxScan* will prevent the user from doing so. If a DVD is identified as DVD-ROM although it is a recorded DVD, then you have most likely installed some driver that (intentionally) impairs media type detecting.

You simply indicate all tests you want to run in the command line, for instance:

`pxscan pie ta=100MB,2000MB,4000MB drivenbr=2`

If you want to perform all scans the drive is capable of at default speed, you can simply launch the bundled `scan.bat` file, which does nothing else than execute

`pxscan all`

---

[19]requires a `PX-714/716` or newer drive

3.3.4   Scanning speed and duration

You can select the scan speed using this:

```
speed=speed
```

However, only speeds that are supported by the firmware are possible. Generally, only the following speed settings are supported:

- CDs: 4x CLV, 8x CLV, 24x CAV, 32x CAV, 40x CAV, 48x CAV
- DVDs: 2x CLV, 5x CAV, 8x CAV, 12x CAV, 16x CAV

The limits for specific disc types are:

- `Premium`: 24x CD-R/W[20], 24x GigaREC CD-R[21]
- `PX-712`: 16x DVD-ROM-5, 12x DVD-ROM-9, **12x DVD±R DL**, 12x DVD±R/W, 48x CD-R, 40x CD-RW, 8x GigaREC CD-R
- `PX-714/716`: 16x DVD-ROM-5, 12x DVD-ROM-9, **8x DVD±R DL**, 12x DVD±R/W, 48x CD-R, 40x CD-RW, 8x GigaREC CD-R

Since reading a disc at higher speeds is usually more difficult, more errors are usually reported when increasing the scanning speed. However, this can still be useful: A disc reporting very low error rates even at high scanning speeds is pretty good (cf. Figure 4). Unfortunately, as you can see in Figure 4, `PIF` results are inconsistent at higher read speeds even for very good discs: a few peaks are likely to occur. Please read section 4 if you are not yet familiar with those scan results and how to read them.

Figures 5 and 6 illustrate that, sometimes, scan results at different speeds can be very different, and that sometimes scan results at all available speed settings show a bad result, but some other time a scan result at high speed suggests a disc that is not very good, whereas the scan result at low speed shows no particular problem.

When scanning a complete single layer DVD at 2x speed, `PIE`, `PIF` and `Jitter/Beta` scan will require about 28 minutes each, when scanning a double/dual layer DVD[22] at 2x speed, it will require about 50 minutes per scan. The `TA` test should take about 15 seconds per test zone. Since you cannot interrupt a test and continue later, make sure you won't need

---

[20]Although the maximum read speed is 52x, the maximum scanning speed is limited to 24x

[21]Although read speed can be 32x/40x for GigaREC 1.3x/1.2x discs, scanning speed is limited to 24x

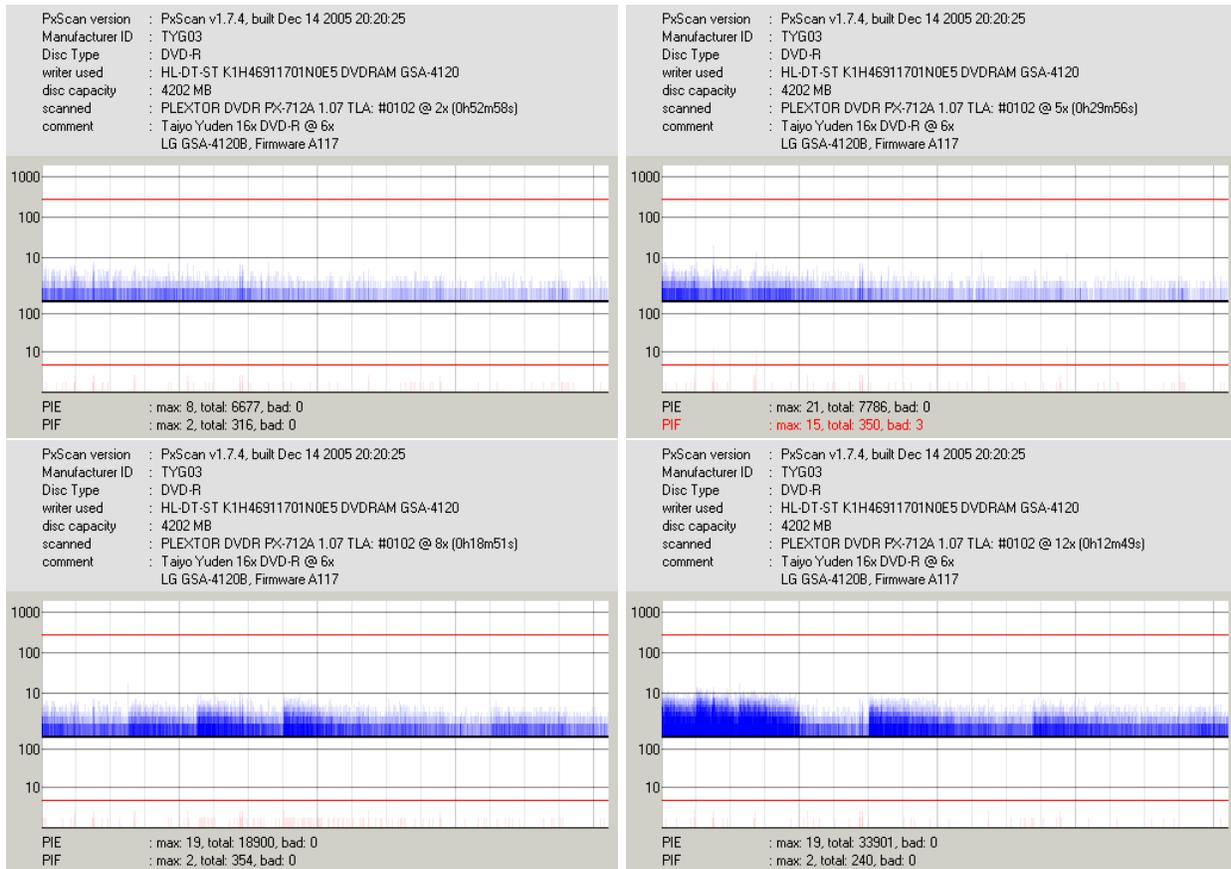[22]DoubleLayer = DVD+R DL vs. DualLayer = DVD-R DL

Figure 4: Influence of scanning speed - sample 1

the scanning drive nor another drive connected to the same IDE channel within that time.

At 12x speed, a `PIE` or `PIF` test takes about six and a half minutes on a single layer DVD, however `jitter/beta` tests cannot be performed any faster than at 2x speed. If you don't want to make a full scan, I recommend to perform only a `PIE` scan at maximum speed. If the result is good, then you can be pretty sure that the DVD is good, otherwise, perform a full scan at lower speed. Please note that scanning double/dual layer DVDs any faster than at 5x usually leads to very high error rates.

When scanning CDs, there is a special issue: If you are trying to scan a CD that is protected with *CactusDataShield 200,* the drive might limit the read speed to 24x or 8x automatically. If you have selected a higher scanning speed when this situation occurs, *PxScan* will enable Single-Session mode and reload the disc in order to fix this problem.
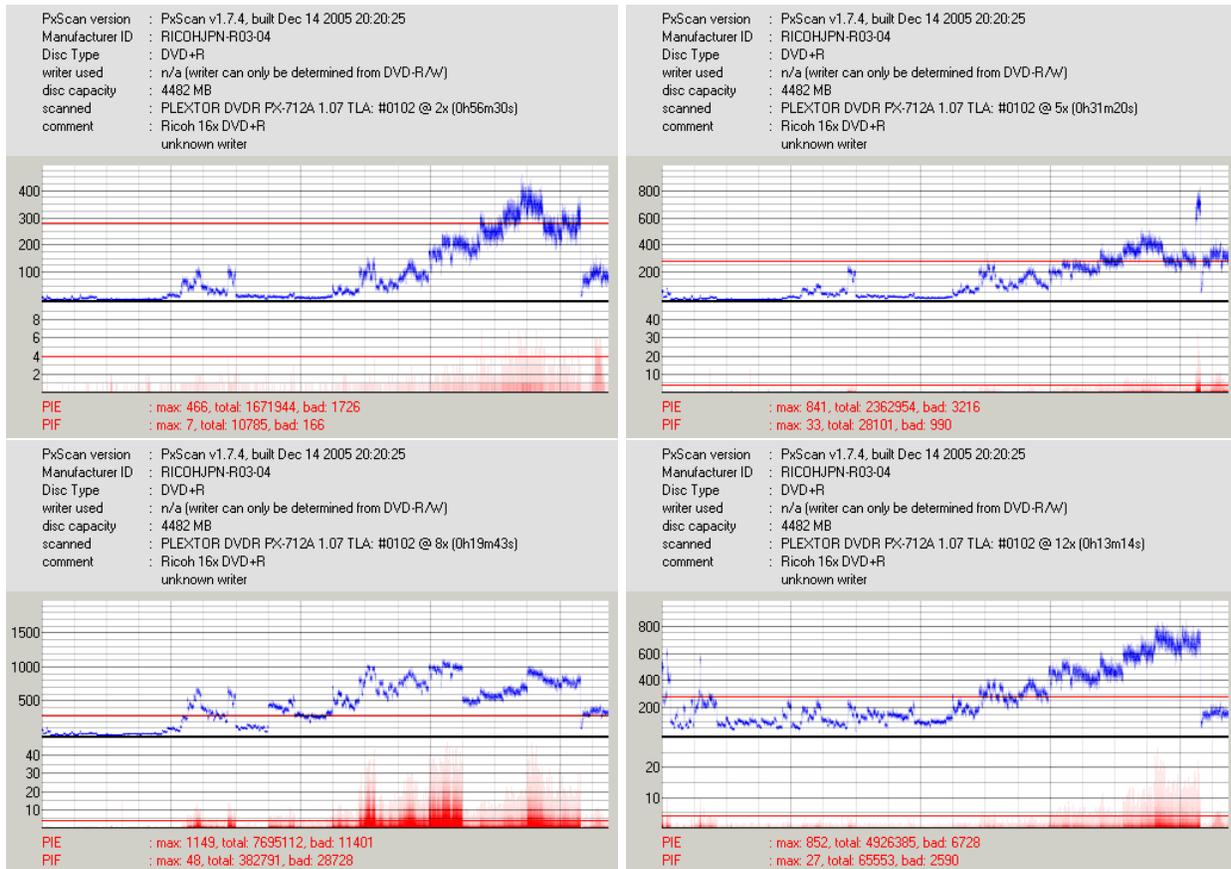
Figure 5: Influence of scanning speed - sample 2

3.3.5 Scanning several discs

If you know in advance that you are going to scan several discs, you can run *PxScan* using the `multi` option, like this:

```
pxscan all multi
```

When using this option, *PxScan* will eject the disc after having finished a scan and will wait for you to insert the next discs. It will continue when the drive has recognised the next disc.

If you do not include an option indicating which drive you want to be used for scanning and let *PxScan* automatically select a drive, a drive will automatically be chosen for the first disc. After having finished the scan and ejected the disc, *PxScan* will wait for the same drive to become ready again. It will NOT look for other drives that could be used instead.
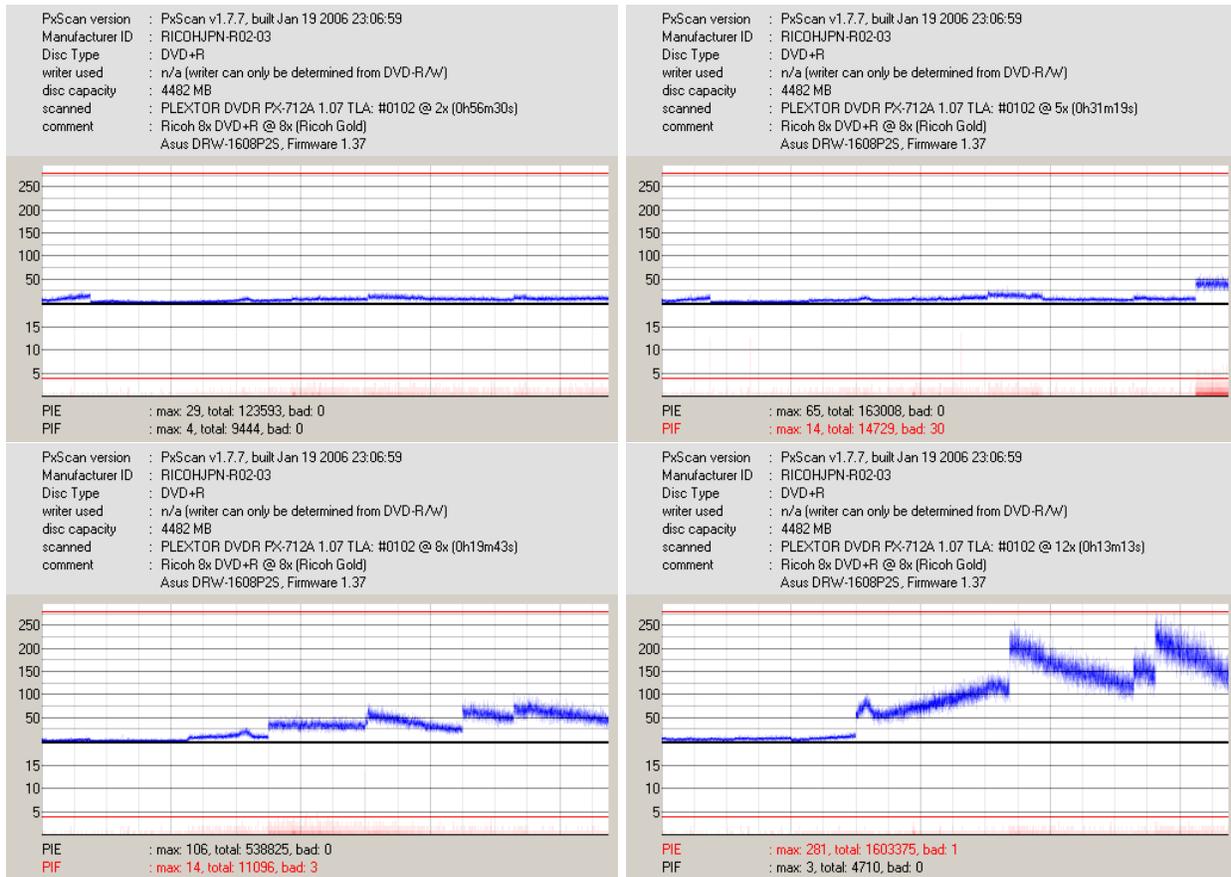
Figure 6: Influence of scanning speed - sample 3

If you want to run several scans on several drives at the same time, run multiple instances of *PxScan*.

3.3.6 Launching *PxView* after scanning

If you want *PxView* to be launched immediately after the scan has finished, use the `autolaunch` option like this:

```
pxscan speed=8 pie pif autolaunch
```

3.3.7 Aborting scan on very bad discs

When scanning very bad discs, the scan can take longer than usual, especially when the drive has trouble reading the disc at all. If you want a scan to be aborted after *n* hard read errors, use the option `abort=`*n*.

3.3.8   Output file

You can specify an output file name using `destfile=`*`filename`*. In this case, the output file name can contain placeholders:

- **%WRITER** will be replaced by the drive used to record the disc being scanned. Note that this information can only be retrieved from DVD-dash discs.

- **%MID** will be replaced by the manufacturer ID of the disc being scanned.

- **%DATETIME** will be replaced by the date and time the scan was finished

- **%RSPEED** will be replaced by the scanning speed

- **%DRV** will be replaced by the name of the drive performing the scan

You can use `destfile=` in order to have a unique filename created by *PxScan*. You cannot use the '%' character for other purposes than a placeholder. Beginning with *PxScan* v1.7.11, the output folder may contain folder names, and *PxScan* will create the folder if required. If the folder cannot be created, it will create a default file name and store it in the current directory.

The file *PxScan* creates contains all error values as well as some meta information about the disc that you have scanned. The following additional information is stored:
- the name, model and TLA[23] number of the scanning device
- start and end time of scan (using Earth Standard Time)
- read speed
- disc type
- writer used to record the disc[24]
- random disc ID number [24]
- GigaREC setting used to record the disc[25]
- disc capacity
- the entire ATIP[26]
- position of layer break[27]

---

[23]TLA is not stored for Plextor Premium since it cannot be retrieved

[24]only `DVD-R, DVD-RW, DVD-R DL`

[24]only `DVD-R, DVD-RW, DVD-R DL`

[25]only `CD-R`

[26]only `CD-R, CD-RW`

[27]only `DVD±R DL`

3.4   Comparison of scan results

When reading about error scanning in forums, you will often encounter questions about

- results of the same drive with the same disc on different scanning applications
- results of different drives with the same disc

You can safely ignore any discussions about the first one: When scanning a disc in a Plextor drive, the following steps are necessary when performing a `PIE`, `PIF` or `CX` scan:

1. set the drive to scan mode
2. read a couple of sectors (e.g. 75 for CDs)
3. retrieve the number of errors the drive has encountered
4. go to 2 if end of disc is not reached
5. set the drive to normal mode

As you can see, the drive is doing all the error counting, the scanning application only has to sum them up and calculate an average value, if necessary. Thus, any discussions about differences between results of PlexTools, *PxScan*, DVDInfoPro and qpxtool are useless, unless someone is claiming that one of those applications would manipulate the scanning result. In the case of *PxScan* and *qpxtool*, everyone can easily look into the source code and verify that results are not falsified.

When comparing scans of different drives, you have to consider that

- drives use different scanning intervals
- different drives, different hardware can see a totally different number of errors. Again: When scanning, you only see how many errors your reading device saw.

Consequently, a comparation of results of different drives is not really possible and not really useful, unless you check a couple of different discs (singelayer, double/dual layer, recordable, rewriteable, different quality, different write speeds) and find out whether or not the two drives of which you want to compare scan results produce comparable results.

3.5   Known issues and limitations

This section lists a few problems that I know about, some are bugs, some are firmware or drive limitations. Some of those have already been mentioned earlier.

- Plextor PX-712 can read the manufacturer ID of a DVD+R DL, but still mistakes those discs for DVD-ROMs. This is a Plextor firmware issue

- Plextor PX-712 cannot read the manufacturer ID or writer used for recording from DVD-R DL. This is a Plextor firmware issue

- Plextor PX-712/714/716 can read CDs recorded using GigaREC at 8x speed at most. This is a firmware limitation

- Plextor PX-712/714/716 can read and scan audio on CD-RWs only at 32. This is a firmware limit.

- Plextor PX-716 with firmware below 1.09 cannot retrieve the writer used to record a DVD-R DL. This was a Plextor firmware bug.

- *PxScan* cannot retrieve the TLA version of Plextor Premium drives. This is a missing feature of the drive.

- In a few cases, performing a TA scan can crash the PX-716's firmware. This will require you to reboot the system. In a few cases, the system must be rebooted even when the drive was mounted in a USB/Firewire enclosure. Switching it off in a situation where the mounted drive hangs often works, but in a few cases, it does not! The reason for this problem is unknown.

- Plextor Premium can scan CDs at 24x speed at most. This is a firmware limitation

- Do not use drivers that hide ATIP, otherwise disc type identification will malfunction. An exception is software that simply activates Plextor's Hide CD-R function, as *PxScan* can temporarily disable this function in order to retrieve the ATIP.

- Adaptec ASPI does not work reliably with *PxScan*.

- When a USB bar is connected to the system at the moment *PxScan* is launched using Frog ASPI, Windows XP refuses to disconnect this USB bar while the scan is being performed. This is a bug of *PxScan* or of the ASPI layer.

- DVD-R DL scanning will most likely not work for discs recorded in LayerJumpMode.

4   *PxView*

After having performed a scan using *PxScan,* you can get a graphical representation of the result using *PxView.* This sections deals with its usage, its graphical representation of error measurement results and with its default settings.
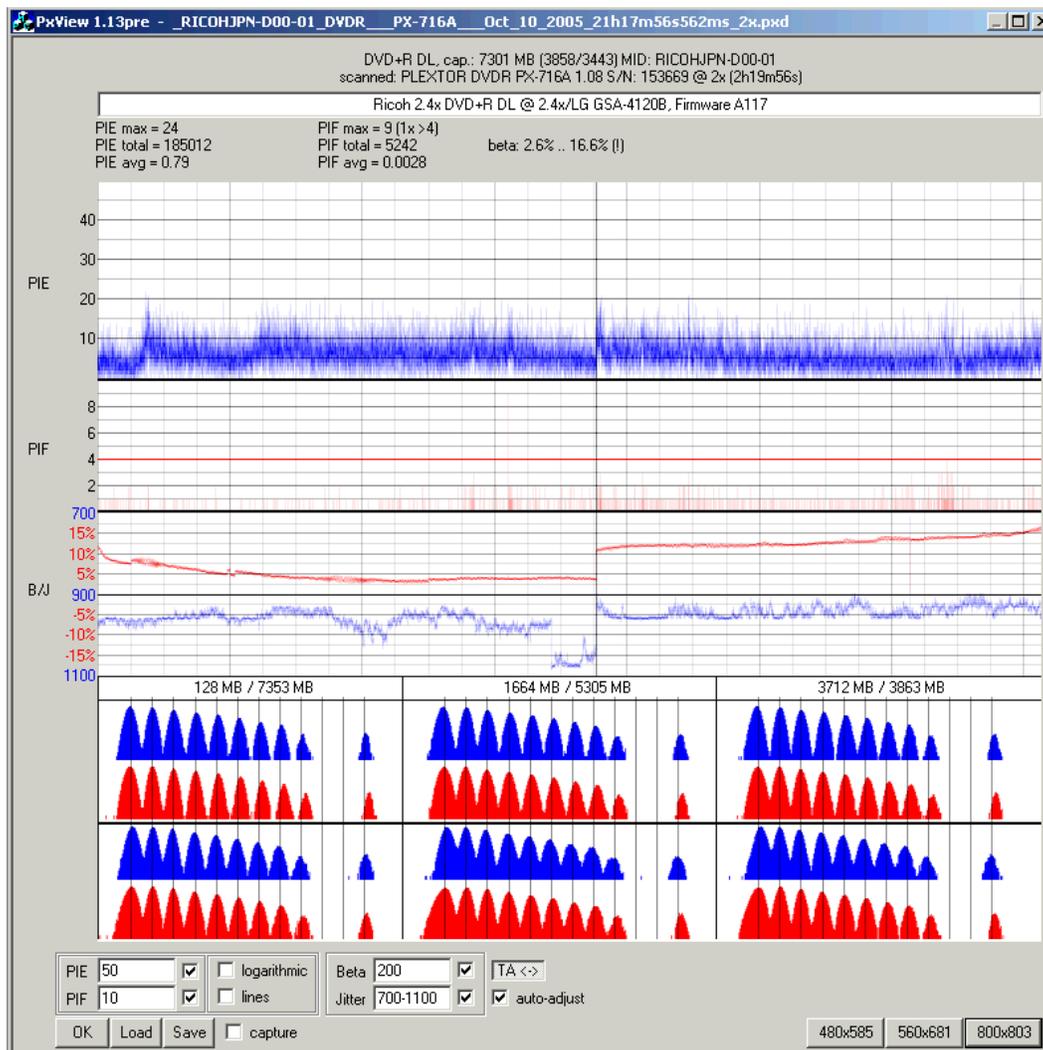

Figure 7: *PxView* - Overview

4.1   Color shading in *PxView*

In order to understand how color shading is done in *PxView*, you need to know a few numbers: One physical sector on a DVD contains $32kB$ of data, an entirely recorded DVD+R contains $143,444$ of those. Consequently, a PIE test with a scan interval of 8 physical sectors will yield almost $18,000$ values, a PIF test will return one value per physical sector. Now if you are using a display resolution of $1600x1200$, which usually results in letters too small to be readable for screens smaller than 21", one vertical line in a `PIE` scan result representation has to represent 11 values, one vertical line in a `PIF` scan result contains information about 90 values. If you are using a lower screen resolution, one vertical line contains information about even more values.

Figure 8 illustrates how PlexTools and *PxView* deal with this issue. You can see two scans of a disc that has been preparated using a needle, simulating a tiny flaw in material, or a tiny bit of dirt on the disc while burning the disc. In *PxView* the darkness of one line says how many of those values represented by one vertical line exceed a certain value. Although you cannot *exactly* say how many values exceed a certain limit, you can still say if it's only a few, or a few more than only a few. In PlexTools' picture, on the other hand, you can only see the highest value of all values represented by one such line.

Such a tiny flaw in the material does not cause error rates, PIE as well as PIF, to increase too much per scan interval (i.e. 8 or 1 physical sector(s)), meaning that the error values you will see in an error graph at one certain position are not much higher than before. However, the number of errors in a larger interval will increase, the errors will be more dense. Thus, the color becomes darker in *PxView*.
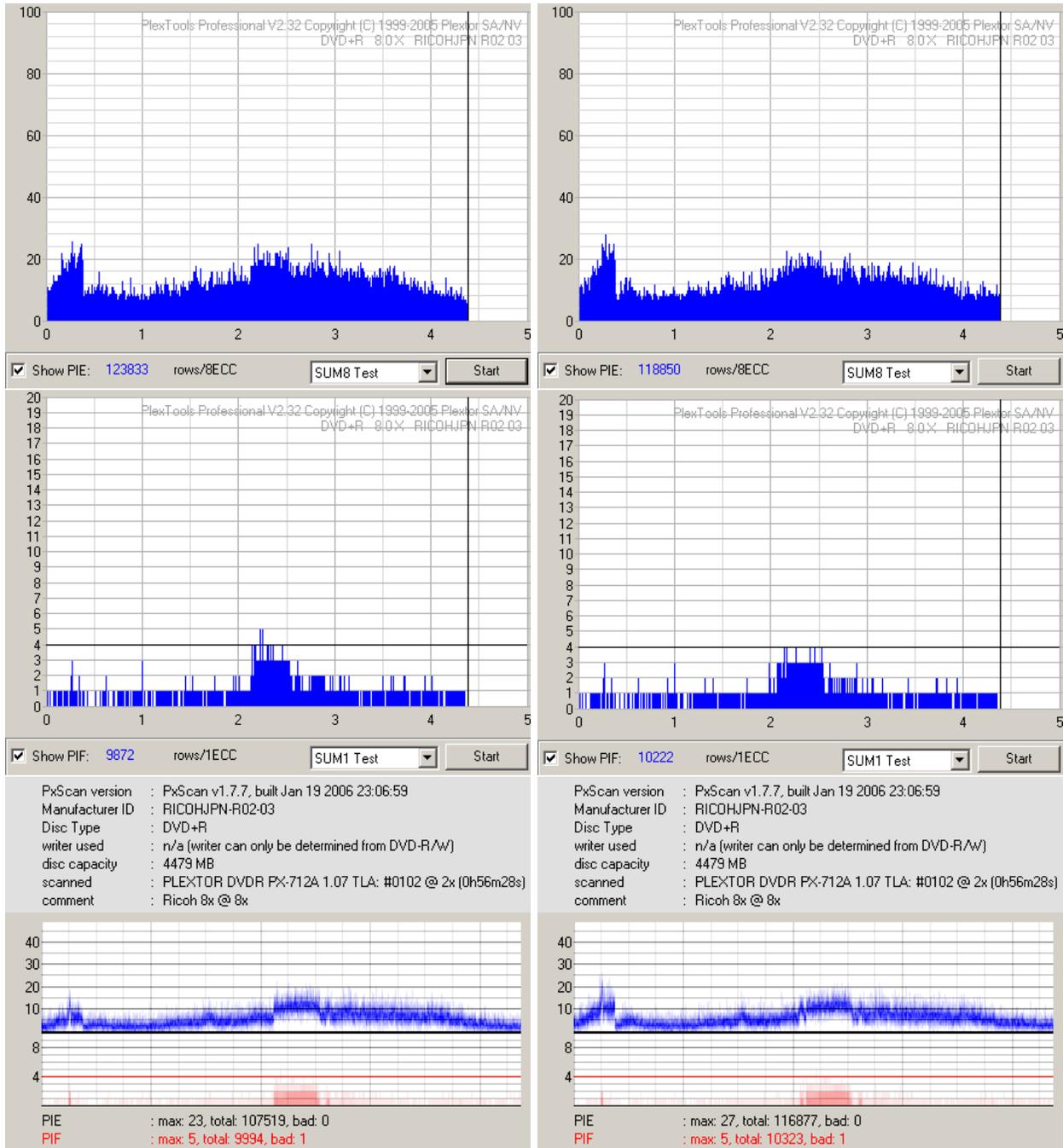
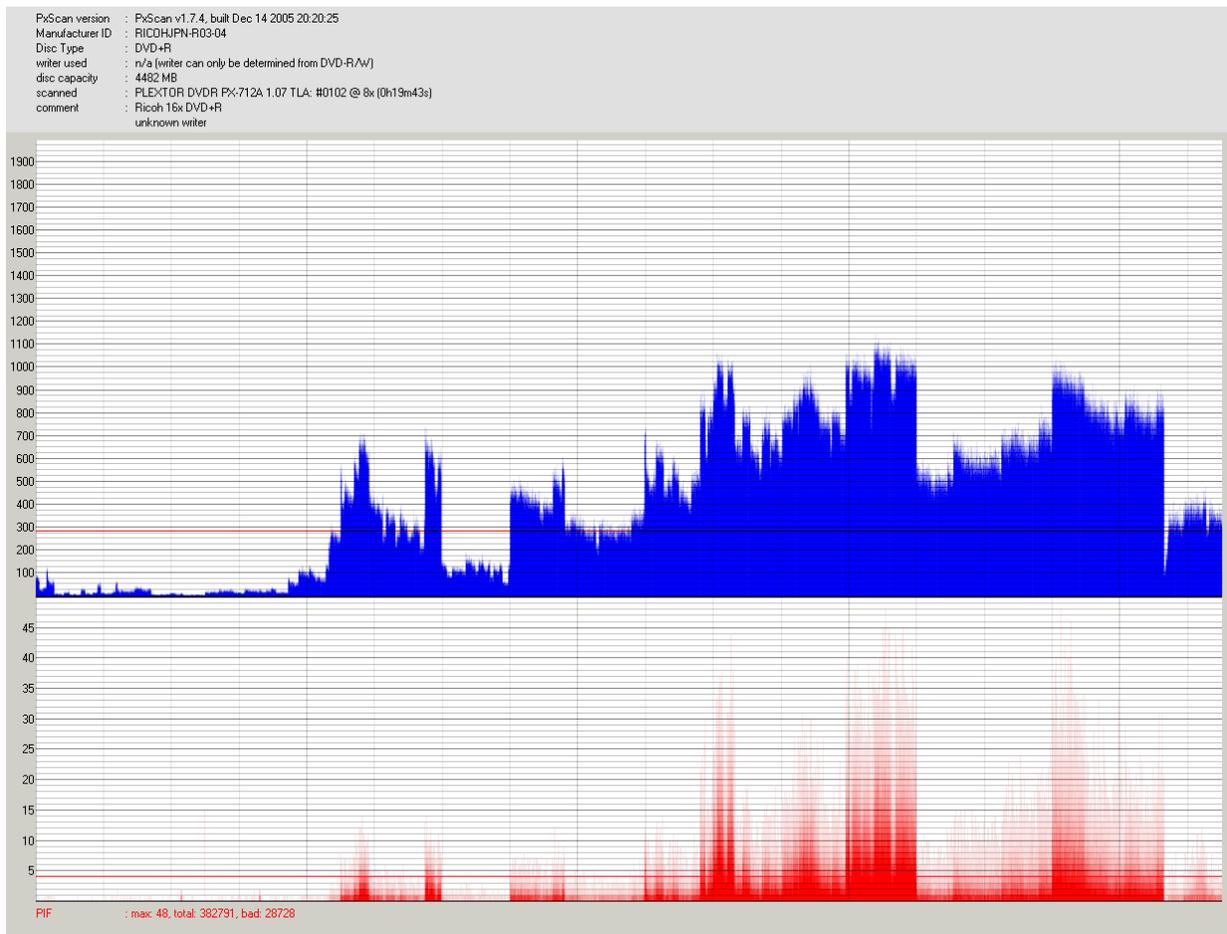Figure 8: Scan of a physically damaged disc - before and after

Figure 9: Output in full screen mode

This obviously has one downside: Rendering the graph takes a lot of time, especially when a scan contains a lot of `PIF` errors. On my development system[28] rendering results such as the one illustrated in Figure 9 in full screen mode (1280x1024) can require up to 1 Gclk[29] without using SSE or SSE2 extensions and more than 600 Mclk when using SSE2 extensions.

4.2   Using *PxView*

PxView does not require any special software or hardware to run, however, if you want to use PNG output when saving picture files, you need `libgfl`, which is available on the *PxScan/PxView* download page. This `.dll` must be put into *PxView*'s directory. Since this

---

[28]Pentium 4 HT, 2.8GHz, 2 GB RAM

[29]billion CPU clock cycles

`.dll` file is larger than *PxScan* and *PxView* together, it is not included in each new version.

4.2.1   Headline

When saving scan results as BMP or PNG files, you can use a headline: In this headline, you use a slash ('/') in order to start a new line.

4.2.2   Scan results

If you have opened a scan result that contains several results, you can select which results you want to see by enabling or disabling the check boxes next to `PIE`, `PIF`, `Jitter` and `Beta`.

4.2.3   Scale range

You can select the scale range manually or you can have it automatically selected.

When selecting the scale range manually, you are not required to, but you should disable `auto-adjust`, since otherwise it will be automatically modified again for example when you switch between linear and logarithmic scale, or when you enable or disable full-line mode.

When having the scale range selected automatically, you can still influence *PxView*'s behaviour to some extent by modifying its `config.ini` file accordingly. See section 4.6.3.

Although beta and jitter scale can also be set automatically, *PxView*'s behaviour in automatic mode cannot be changed. The maximum beta scale will be set to 10%, 20%, 30% or 40% when `auto-adjust` is enabled. The scale will be devided into steps not smaller than 1%, usually it will be 5% or 10%, depending on picture size and minimum/maximum beta values of the disc.

4.2.4   Scale type

The most important setting is the scale type, which can be *linear* or *logarithmic*. Have a look at Figure 10 and you'll easily see when you should use the logarithmic scale, and when the logarithmic scale only makes reading specific values more difficult. Although it is easy for a human to decide whether a logarithmic or a linear scale should be used for a given scan result, this is not at all easy to do automatically. Thus, the `auto-adjust` setting only affects the maximum values used for each scale, but it does not affect the scale type.
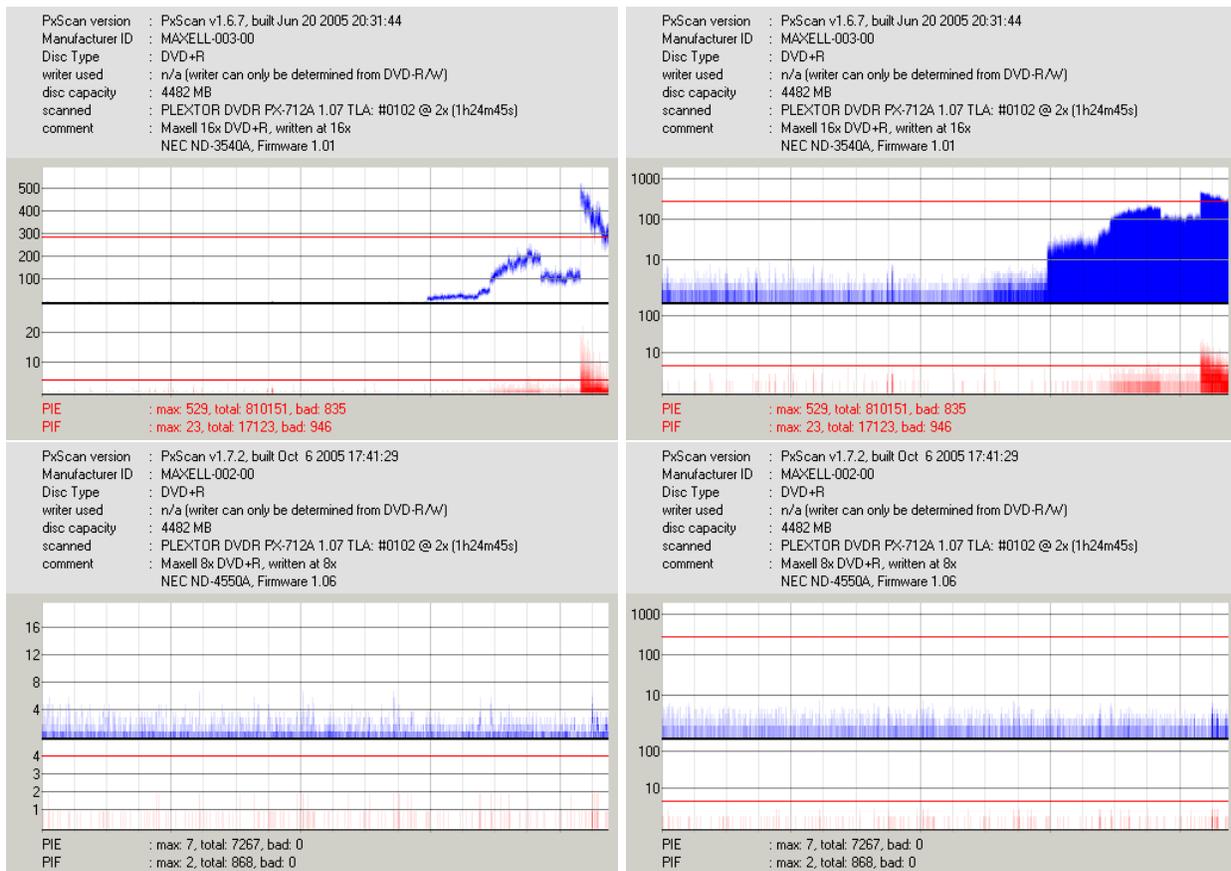
Figure 10: Linear vs. logarithmic scale

4.3 Capturing - display scan result while scan is in progress

When enabling the 'capture' check box, the scan result is shown while the scan is being performed. However, meta data will not be transferred this way. This issue will be taken care of in v1.8 of *PxScan* and v1.13 of *PxView* . Until then, open the file created by *PxScan* instead of directly saving the result *PxView* has captured in capture mode!

4.4 Evaluating a scan result

This section explains what makes a scan result good or bad.

While some scanning tools, like `DVDInfoPro`[30] or `Nero CD-DVD Speed`[31], are trying to evaluate scan results automatically, *PxView* will not try to do something that can only fail. If

---

[30]`http://www.dvdinfopro.com/`

[31]`http://www.cdspeed2000.com/`

you seriously want to check your discs, you must learn how to interpret these results on your own.
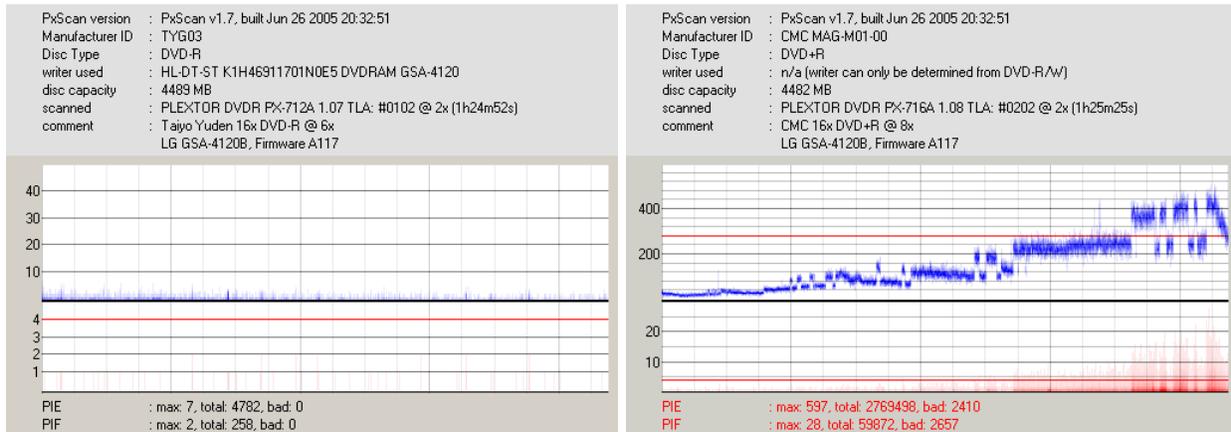
4.4.1 Evaluating `PIE/PIF`

Figure 11: `PIE/PIF` - a good and a bad example

The fewer errors there are, the lower the highest error value is and the more evenly the errors are distributed, the better is the result. Figure 11 shows a good and a bad `PIE/PIF` scan result. The bad example here is showing a typical bad disc, with error rates getting higher and higher towards the end of the disc despite of a low scanning speed.

The red lines you can see in Figure 11 indicate the maximum values allowed for `PIE/PIF`, which are 280 and 4. Beneath the picture, the `bad` value indicates how many values of the corresponding test exceed that limit. A few isolated `PIF` spikes higher than 4 are usually no problem and can be ignored, however, good `PIE` results do not even come close to 280.

4.4.2 Evaluating `Beta`

A Beta scan result is good if there are no stair-case shaped jumps. Also, the beta graph should stay between -5% and +15% (for DVDs) and between -8% and +8% for CDs. However, stair-case like jumps are more problematic than beta values outside the given range. Figure 12 shows an example of a good and a bad beta scan result.

4.4.3 Evaluating `Jitter`

Unfortunately, only very rough evaluation of jitter scans is possible using Plextor hardware: Plextor drives report only some absolute values, without any obvious unit, so you can only
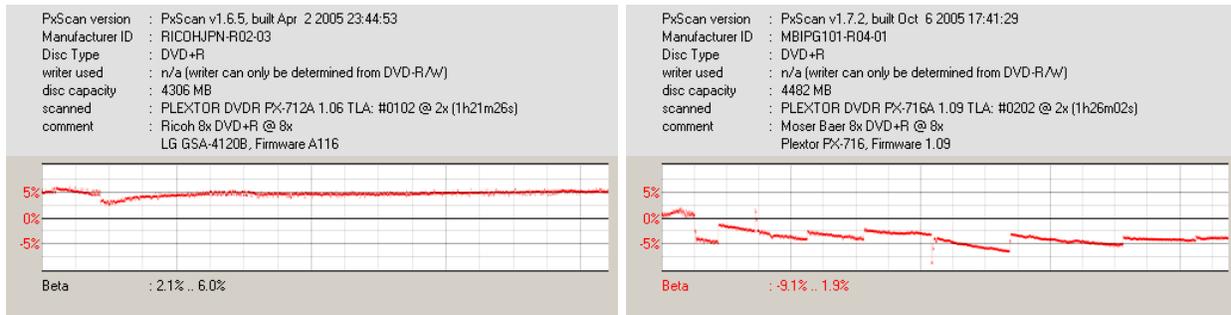
Figure 12: `Beta` - a good and a bad example

say things like

- jitter is increasing towards the end of the disc
- jitter is about the same everywhere on the disc

However, it is not possible to draw conclusions like 'jitter is high' or 'jitter is low'.

4.4.4   Evaluating `TA`

A `TA` test result is better when the peaks are more clearly seperated from each other (cf. Figure 13). The numbers given beneath the graph indicate the root mean square error of the `TA` result data. Higher values mean higher jitter, which is bad.

Also, the peaks should have their maximum each one on its black line. However, this is not yet automatically checked. If the maximum is too far off the black lines, you have high (and bad) *peak shift*.

Please note that the histogram is always presented using a logarithmic scale. This cannot be changed.

4.4.5   Evaluating `C1`, `C2`, `CU`

A CD should not show `C2` or `CU` errors and should show as few `C1` errors as possible. In Figure 14 you can see a good and a bad `CX` scan result. Note that the bad result shows the scan of 'Grönemeyer - Mensch', a pressed CD protected using *CactusDataShield 200*: These CDs are intentionally made in a way to show `CU` errors, so that the player must use error concealing techniques. In the `C2` graph, red is `C2` and green is `CU`.
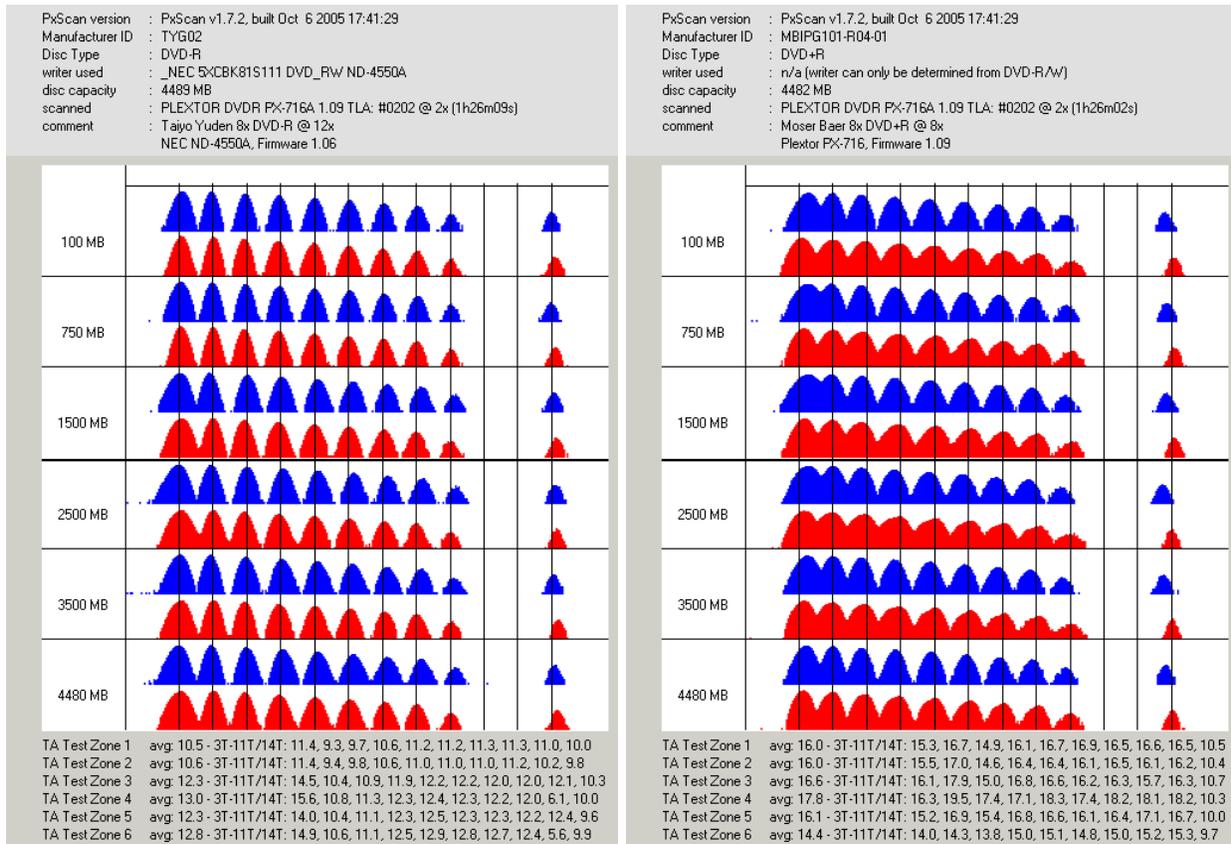
Figure 13: `TA` - a good and a bad example

4.5   Known issues, solutions and limitations

- (Should be solved by now!) *Black picture instead of a graph on Athlon XP CPUs.* This happens because there is something wrong with the SSE code. The code works on Pentium 4 CPUs and Athlon 64, but somehow it does not seem to work on Athlon XPs. When you experience this problem, change the the following lines in the config.ini file: `no_sse=0` and `no_sse2=0` to `no_sse=1` and `no_sse2=1`

- *Nothing is drawn at all.* This can happen when your desktop is running at 24 bits per pixel. Use 16 or 32 bits per pixel with this application, but not 24 bits per pixel. Since my own graphic card does not even support 24 bits per pixel, I cannot add and test support for this mode.

- *Certain scale settings look ugly*. This occurs for example on linear scale when being set to 1001: The upper pixel line is clipped. I'll fix this when I find a reasonable solution. In the meantime, don't use weird scale settings...
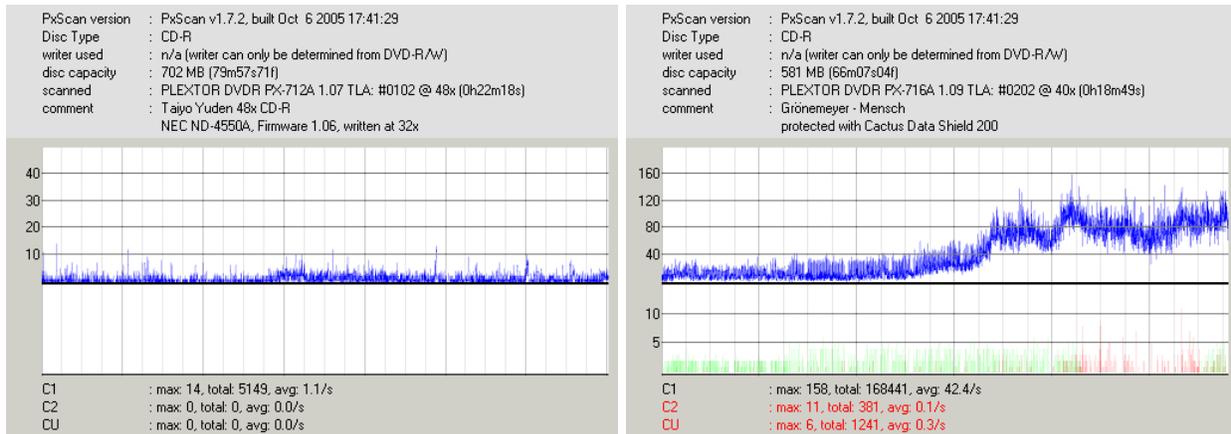
Figure 14: `CX` - a good and a bad example

4.6 The `config.ini` file of *PxView*

You can edit this file in order to change default settings, like coloring, default scale type, window sizes when pressing one of the three picture resolution buttons, and more.

4.6.1 Default window sizes

You can define 3 default window sizes which you can choose by clicking on one of the three buttons being labeled with those sizes. In order to set a certain size, change one set of lines

`button`*n*`_x=`*width*
`button`*n*`_y=`*height*

to whatever size you like. Note that, when trying to set a window size larger than your screen resolution, *PxView* will automatically reduce the window size.

You can choose which size should be chosen per default when launching *PxView* by setting

`default_size_button_index=`*n*

with *n* being the index of the button to choose, i.e. it must be in the range `[1..3]`.

The size you indicate will be used for a scan result that contains

- `PIE`
- `PIF`

- `Jitter/Beta`
- `TA` results for 3 test zones with all 3 result being displayed in one line

If you load a file containing a different set of test results or display only 1 `TA` test result per line, the sizes will automatically be adapted, so that the proportions and sizes used to represent each test result and each `TA` result line are kept the same as the conditions above were met.

The ratio between `PIE`-, `PIF`-, `Beta/Jitter`- and `TA` graph height is changed using `pieheight`, `pifheight`, `betaheight` and `taheight`, for example

```
pieheight=0.6
pifheight=0.4
betaheight=0.5
taheight=0.4
```

As you can see, it is not necessary that the sum of those values be 1. *PxView* will normalize them automatically.

4.6.2   Test results displayed by default

You can select which test results are displayed or hidden by default when launching *PxView* using the following entries in the `config.ini` file: `displaypie`, `displaypif`, `displaybeta`, `displayjitter`, `ta_displaymode`. Note that `displaypie` refers to `PIE` as well as `C1` and that `displaypif` also refers to `C2`.

`ta_displaymode` can be set to

- `0` in order to hide `TA` test results
- `1` in order to display one `TA` test zone per line
- `2` in order to display three `TA` test zones per line

4.6.3   Configuration of scale setting

You can enable or disable automatic scale adaption per default by setting `autoscale=1` or `autoscale=0`. You can enable or disable logarithmic scale by default by setting `logscale=1` or `logscale=0`. The default scale settings are defined by `pisum8scale`, `pifscale`, `betascale`, `jitterminscale` and `jittermaxscale`.

Use `autoscale_steps` to define which scale settings *PxView* can use when enabling automatic scale adaption. Such a line looks like this:

> `autoscale_steps=`*type*,*count*,n_1,n_2,...,n_{count}

type can be 0 to indicate `PIE` and 1 to indicate `PIF`. Here a complete example:

> `autoscale_steps=0,8,10,20,50,100,200,500,1000,2000`

The line above would make *PxView* adjust the `PIE` scale to 10, 20, 50, 100, 200, 500, 1000 or 2000, choosing the smallest possible value of those above which is larger than the highest `PIE` error number of the disc.

Those settings are always used for linear scale mode. You can disable auto scale adaption for logarithmic scales by using `autoadjust_log_scales=0`.

4.6.4 Segmentation

In the picture files made with *PxScan* you can see vertical lines of which the only purpose is to improve readability. Per default, a vertical line is drawn

- each 256 MB (DVDs) / each 25 MB (CDs) in light gray
- each 1024 MB (DVDs) / each 100 MB (CDs) in darker gray

In *PxView*, you can configure the interval in which these lines are drawn as well as their intensity. You can change the following lines in the `config.ini` file in order to change the default settings:

```
strong_vertical_line_interval_dvd=1024
weak_vertical_line_interval_dvd=256
strong_vertical_line_interval_cd=100
weak_vertical_line_interval_cd=25
strong_vertical_line_intensity=96
weak_vertical_line_intensity=32
```

For `*intensity` lines, a higher values means a darker gray is used. Note that `strong_vertical_line_interval_`*x* must be divisible by `weak_vertical_line_interval_`*x* and that the unit which is used is $1MB := 2^{20} bytes$.

4.6.5   Configurating colors

You can also alter the colors used to draw the results. The following lines can be changed in order to achieve this:

```
pie_color=FF0000
pif_color=0000FF
jitter_color=FF0000
beta_color=0000FF
ta_color_pit=FF0000
ta_color_land=0000FF
```

A color is indicated using its 3 components blue, green and red. The values must be indicated hexadecimally. Thus, a value of `FF0000` means 'maximum intensity for blue' and no green/red.

4.6.6   Optimizations

Certain operations during drawing the graph can profit from `SSE` and `SSE2`. However, in case there should be something wrong (such code is harder to write and harder to debug than 'normal' code), you can disable it by setting

```
no_sse=1
no_sse2=1
```

References

[1] Frog ASPI
     http://www.frogaspi.com/

[2] Nero ASPI
     http://www.nero.com/nero6/en/WNASPI32.DLL.html

[3] K-Probe
     http://www.k-probe.com/

[4] CDR-Labs
     http://www.cdrlabs.com

[5] Article about ASPI on CDR-Labs
http://cdrlabs.com/articles/index.php?articleid=3

[6] QPXTool
https://sourceforge.net/projects/qpxtool/

[7] Plextor scans @ cdfreaks
http://club.cdfreaks.com/showthread.php?t=115081

[8] Heise.de about quality problems with PX-716
http://www.heise.de/newsticker/result.xhtml?url=/newsticker/meldung/
52869&words=Plextor

[9] Example for a bad result
http://club.cdfreaks.com/showpost.php?p=753383&postcount=7

[10] Plextor says they don't have the resources to support Linux (i.e. to tolerate that inde-
pendant developers add Linux support for their drives)
http://www.pro-linux.de/news/2006/9151.html