

DutchTiVoGuide datamodel

Design document

v1.6 – d.d. 7-Dec-2005 – Dennis van Onselen

1 Disclaimer

DutchTiVoGuide is a system which consists of a database and a set of Perl scripts that can retrieve TV guide data from XMLTV-files and convert it into the slice format used by the TiVo.

Limitations:

- The license on the software only allows it to be used in those parts of the world which cannot obtain a program subscription service from TiVo, Inc.
- The license on the software only allows the program to be used for non-commercial purposes. You cannot sell this program, nor can you use this program to sell a service.

2 Abstract

TiVo Inc. doesn't provide a TV program guide service in the Netherlands like it does in the US or the UK. DutchTiVo is a Dutch initiative to make TiVo's work in the Netherlands. Since TiVo doesn't sell and support any of the TiVo-concept in the Netherlands, we are almost on our own. Almost, because there are similar initiatives which started way before I did. I'm talking about TiVoCanada (www.tivocanada.com) and OzTiVo (an Australian initiative, see www.oztivo.net). The guys behind these names paved the road for me. I only had to look at how they paved it, so I could learn from it and make this system for the Netherlands.

This document describes the way slice-information is created, to make a TiVo work in the Netherlands. This document is not intended to create an alternative for the US and UK TiVo-content system, because that's illegal. It was created because there is no 'official' way to make it work outside the US and UK.

The reason that this document is written in English and not in Dutch is because I believe in sharing knowledge. And knowledge on the internet is better shared in English.

3 Why do you create your own database and software? Didn't these guys from Canada and Australia already did this?

Yes. The only thing is that I encountered a lot of difficulties adjusting these scripts to make them work for the Netherlands. Furthermore I found that it was very difficult to maintain these scripts and their separate datafiles in their various directories at that time. Finally this new setup provides much more modularity: you can build as many parsers and scripts around the database as you like, to provide even better quality guide data. Furthermore this setup provides more flexibility: when a new provider is added for instance, all the corresponding slices are automatically being generated.

A lot of credit needs to go to Warren Toomey and the guys from TiVoCanada, since they pioneered all of this. I just spent hours and hours on trying to understand everything that goes on under the hoods of their scripts and my TiVo. With that information I was able to build my own system.

4 Wat is needed?

Let's talk business: what equipment, software and data is needed for us to make this work?

4.1 Hardware

Here's a hardwarelist:

- A series 1 U.S. TiVo (UK PAL-version works as well)
- Turbonet-card or similar, to provide Ethernet networking. The phone-connection or a serial connection (might) work as well.
- VCR or any other external receiver with IR-support, and/or a PAL-tunermod (the standard TiVo-tuner only handles NTSC-cable).

- A harddrive. The bigger, the more tv-programs can be stored.

4.2 Software

There are two important things which we need here:

1. An adjusted TiVo-image is necessary to be able to use essential things as the network functionality and time-settings. For this purpose a special nTiVo-image has been created, which can be obtained from www.nltivo.net.
2. Furthermore we are needing TV-Program data. Your TiVo won't really work without it.

4.3 Data

Since TiVo won't provide us with necessary Dutch data to make the device useful in The Netherlands, we have to obtain our own. We need the following data:

1. Lineup-data
2. Guide-data

When this data is available it needs to be presented to your TiVo by using a TiVo-emulator or by manually uploading these files to your new gadget.

4.3.1 Line-up data

Line-up data tells what channels can be received in a certain area and through which medium (satellite (or any other external receiver) or cable). It also tells your TiVo at which channel the stations will be positioned (i.e. '4' = RTL4). This line-up data is presented to your TiVo via so-called headend-slices. (Frequencies of channels are not stored in the headend-slices, they are stored on your TiVo through the 'palmod'-kernelmodule. Despite this the frequencies are stored in the database to make automatic generation of the config-file for palmod possible.)

4.3.2 Guide-data

Guide-data tells the TiVo what's on TV. Together with information like genres, ratings, actors, episode-information and of course date and time information, your TiVo will be able to record programs for you and generate a profile of the things you like to watch. Without this guide-data your TiVo is next to useless.

5 How do we get all this information?

Well, luckily we have XMLTV. XMLTV (www.xmltv.org) presents tv-guide data in XML-format. Dutch tv-data is also available in this format. TiVo expects two weeks of guide-data ahead. In The Netherlands we only have six or seven days worth of guide-data ahead available. This will make the TiVo nag that it has not enough data. We will solve this by setting up a fake channel (NoNag) and fill the guide-data for that channel with two weeks of fake data. That way, your TiVo will think it has enough TV-program information. The only thing we have to do, is to make sure that the guide-data on our TiVo is updated at least every few days (the Dutch emulator updates guide-data once a day).

6 Uhhm, but XMLTV-data cannot be fed into your TiVo, right?

That's correct. That's what this document is all about: describing the way to feed our database and the way to make the necessary slices.

7 The idea

This is the objective:

To be able to dynamically produce TiVo headend- and guide-slices for the Netherlands, based on the location and channel-provider we are using.

Since I'm not planning on entering provider-data at once for all the providers in the Netherlands, I want to be able to dynamically add or remove providers, stations and/or channels without changing the code. When I insert or remove a provider, station or channel in the database, I want the program to create the according headend- and guide-slices dynamically.

8 The datamodel

To make this idea work, we need to store a lot of data. We need a smart way of doing this. I've thought about it for many hours, and this is what I came up with (it can be perfected even more, but this works fine):

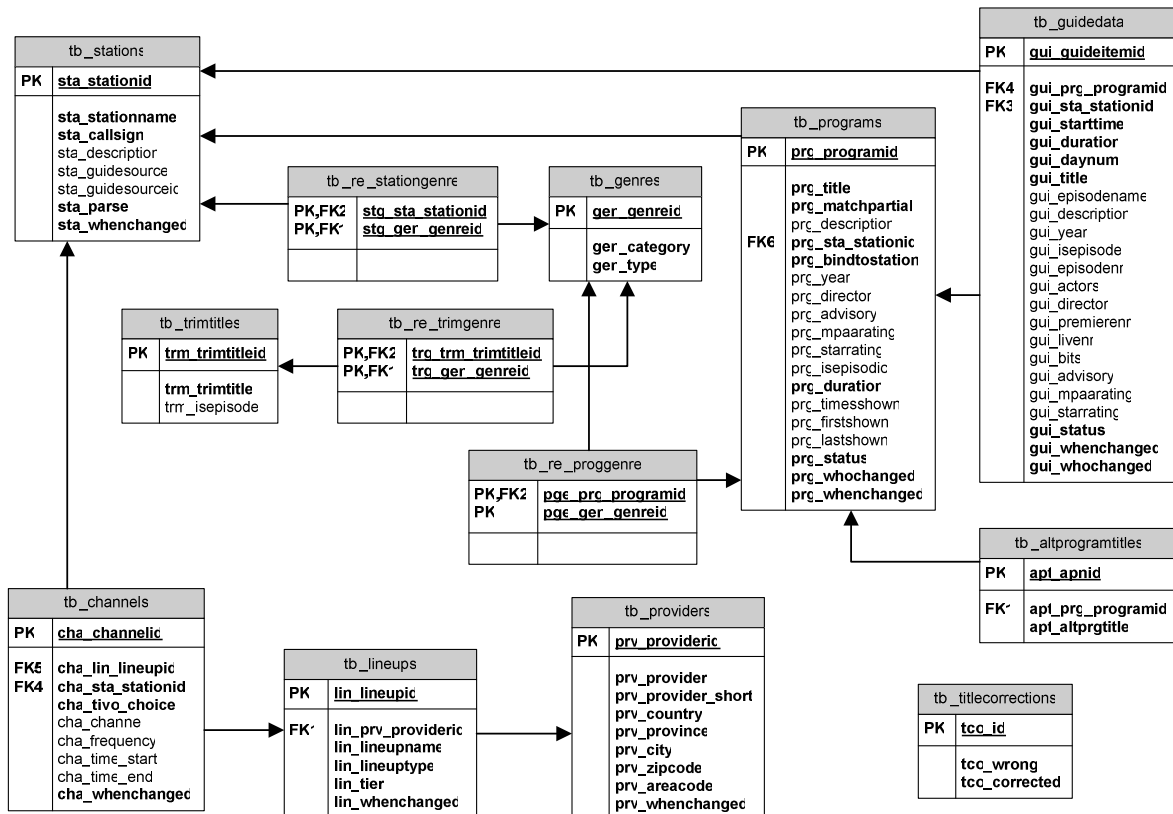


Figure 1 – ERD DutchTiVoGuide

In figure 1 the Entity Relationship Diagram (ERD) of DutchTiVoGuide is displayed. There are ten main tables and three link tables:

1. tbl_stations: holds *all* the existing stations, regardless whether your provider(s) carries these stations.
2. tbl_providers: holds all providers (including detailed information) known to DutchTiVo.
3. tbl_channels: holds the channel-information per provider, including TiVo-lineups for that provider.
4. tbl_lineups: holds all the lineups for the providers.
5. tbl_genres: holds every genre known to your TiVo (predefined by TiVo Inc.).
6. tbl_trimtitles: holds standard parts of program-titles which need substitution in the program-description (i.e. 'mini-serie:', 'aflevering:', 'movie:' or 'film:'). This table is only used by our scripts to prepare the guide-data.
7. tbl_programs: holds every program ever broadcasted and some of it's detailed info. Every program that is mentioned in table guidedata needs to be in the programs-table.
8. tbl_altprogramtitles, contains alternative titles to programs. To prevent double entries in the tbl_programs table (i.e. 'Dawson's Creek' vs. 'Dawson Creek' vs. 'Dawsons Creek')
9. tbl_guidedata, contains which program is on what channel. Stores data with a timespan of a week in the past to two weeks ahead (where available). Refers to programs in the programs-table, but some values can be overridden here (things like duration, episode or description may vary for instance)
10. tbl_titlecorrections, contains wrong titles of programs, together with their correction.

The `tbl_rel_stationgenre`, `tbl_rel_proggenre` and `tbl_rel_trimgenre` link tables are used to create an 'm to n' relationship between the different tables and the genre table. A station, program and trimtitle can have more than one genre. They need to have at least one broad-genre. More are optional. Specific genres are optional, but are used by your TiVo to create a profile for better recording suggestions. The more genres, the better the profiling.

9 What is stored in these tables?

This chapter will dig into each table mentioned in the ERD in figure 1. The datatypes are represented in PostgreSQL-format. A separate SQL-file for creating these tables is available on the DutchTiVo-website (www.nltivo.net).

9.1 Table `tbl_stations`

Attribute	Datatype	Mandatory	Description
<code>sta_stationid</code>	SERIAL	Yes, PK	Unique id, number, used by TiVo. Autoincrement. Offset (x000000) is added by the slice-generator for The Netherlands (see <code>dtg.conf</code>).
<code>sta_stationname</code>	VARCHAR(32)	Yes	The full name of the station (i.e. 'CNN International')
<code>sta_callsign</code>	VARCHAR(7)	Yes	The short-name which will be displayed In your TiVo (i.e. 'CNN')
<code>sta_description</code>	TEXT	No	Description of the channel (profile). Optional.
<code>sta_guidesource</code>	VARCHAR(16)	No	Mentions which parser is used for the guide-data to be provided (i.e. 'wolf' or 'none')
<code>sta_guidesourceid</code>	VARCHAR(64)	No	Helper for the parser under 'guidesource'. (i.e. the helper for 'NED1' is 'nederland1.omroep.nl' for the wolf-parser)
<code>sta_parse</code>	BOOLEAN, DEFAULT false	Yes	Used by the feeder to determine whether this station needs to be parsed from the XMLTV-source
<code>sta_whenchanged</code>	TIMESTAMP WITHOUT TIMEZONE DEFAULT false	Yes	Used by the slice-script to determine whether or not new head-end slices must be generated.

9.2 Table `tbl_providers`

Attribute	Datatype	Mandatory	Description
<code>prv_providerid</code>	SERIAL	Yes, PK	Unique id, autoincrement. Not used by TiVo
<code>prv_provider</code>	VARCHAR(64)	Yes	Name of the provider (i.e. 'CAIW Naaldwijk')
<code>prv_provider_short</code>	VARCHAR(10)	Yes	Short name of the provider (i.e. 'CAIW')
<code>prv_country</code>	VARCHAR(16) DEFAULT 'NL'	Yes	Country of the provider. Two letters (i.e. 'NL')
<code>prv_province</code>	CHAR(2)	Yes	Set this to your province, 2 letters only, ZH, NH.
<code>prv_city</code>	VARCHAR(32) DEFAULT 'Unknown'	Yes	City of the provider(i.e. 'Naaldwijk')
<code>prv_zipcode</code>	CHAR(5)	Yes	Important. The <i>provider's</i> 5-digit zipcode (i.e. '02671'). Used by TiVo to select lineups. Unique.
<code>prv_areacode</code>	CHAR(3)	Yes	Three-digit areacode (CHAR, because some areacodes start with 0)
<code>prv_whenchanged</code>	TIMESTAMP WITHOUT TIMEZONE DEFAULT now()	Yes	Used by the slice-script to determine whether or not new head-end slices must be generated.

9.3 Table `tbl_channels`

Attribute	Datatype	Mandatory	Description
<code>cha_channelid</code>	SERIAL	Yes, PK	Unique id, autoincrement. Not used by TiVo

cha_lin_lineupid	INT	Yes	Reference from attribute with the same name from table 'lineups'.
cha_sta_stationid	INT	Yes	Reference from attribute with the same name from table 'stations'
cha_tivo_choice	SMALLINT	Yes	Preset channel used by TiVo (i.e. '4' for 'RTL4')
cha_frequency	NUMERIC(5,2)	No	The frequency in MHz. up to three digits and two decimals.
cha_time_start	INT	No	The time on which the channel starts (when more than one station uses the same channel, you will have to provide two channels. Otherwise sometimes your TiVo will not be able to record your show. This is due to guidedata which is provided on a per-channel basis). In seconds from 00:00 that day. When not applicable: NULL.
cha_time_end	INT	No	The time on which the channel stops (see time_start). In seconds from 00:00 that day. When not applicable: NULL.
cha_whenchanged	TIMESTAMP WITHOUT TIMEZONE DEFAULT now()	Yes	Used by the slice-script to determine whether or not new head-end slices must be generated.

9.4 Table tbl_lineups

Attribute	Datatype	Mandatory	Description
lin_lineupid	SERIAL	Yes, PK	Unique lineup-id.
lin_prv_providerid	INT	Yes	Reference to the providerid in the table providers.
lin_lineupname	VARCHAR(24)	Yes	The name that shows up when selecting a lineup on your TiVo (i.e. 'CAIW Naaldwijk kabel').
lin_lineuptype	SMALLINT	Yes	Use '1' for cable or '17' for external receiver.
lin_tier	SMALLINT DEFAULT 1	Yes	1=Basic, 2=ExtendedBasic, 3=Premium, 4=PayPerView, 5=Music. 1 is safe pick.
lin_whenchanged	TIMESTAMP WITHOUT TIMEZONE DEFAULT now()	Yes	When was this record last changed?

9.5 Table tbl_programs

Attribute	Datatype	Mandatory	Description
prg_programid	SERIAL	Yes, PK	Unique id. Integer, auto-increment. Used to create TiVo-program and episode id's (add a basenumber (see dtg.conf in Perl-scripts) to this number to get it).
prg_title	VARCHAR(64)	Yes	The textual program-title (i.e. 'Dawson's Creek')
prg_matchpartial	BOOLEAN DEFAULT false	Yes	Is the title in prg_title a partial title?
prg_description	TEXT	No	Textual description of the program, arbitrarily long, single line
prg_sta_stationid	INTEGER	No	On which station is this program shown? (reference to sta_stationid in tbl_stations)
prg_year	SMALLINT	No	Year the program was created (4 digits)
prg_director	VARCHAR(64)	No	Director of the program
prg_advisory	VARCHAR(21)	No	Comma separated list of numbers between 1 to 10, representing: Language, GraphicLanguage, Nudity, BriefNudity, GraphicViolence, Violence, MildViolence, StrongSexual, Rape, AdultSituations.
prg_mpaarating	SMALLINT	No	Number between 1 to 6, representing: Y7 (Youth >= 7 years), Y (Youth), G (general audience), PG (parental guidance suggested), 14 (>= 14 years), M (mature). See also this link .
prg_starrating	SMALLINT	No	Starrating. Number between 1 to 7, representing:

			one or *, one point five or *+, two or **, two point five or **+, three or ***, three point five or ***+, four or ****.
prg_isepisodic	BOOLEAN	No	Is this program episodic (yes='1', no='0')
prg_episodenr	SMALLINT	No	Episodenummer of the program's season.
prg_duration	INT	Yes	Standard duration of this program in seconds
prg_timesshown	INT	No	The amount of times broadcasted since firstshown.
prg_firstshown	TIMESTAMP WITHOUT TIMEZONE	No	When was this program first broadcasted?
prg_lastshown	TIMESTAMP WITHOUT TIMEZONE	No	When was this program last broadcasted?
prg_status	CHAR(1) DEFAULT 'N'	Yes	1 character ENUM. "N" is new, "C" is checked, "E" is enriched.
prg_whochanged	VARCHAR(16)	Yes	Who changed the record last
prg_whenchanged	TIMESTAMP WITHOUT TIMEZONE DEFAULT now()	Yes	When was the record last changed (updated when the program was found scheduled)

9.6 Table tbl_altprogramtitles

Attribute	Datatype	Mandatory	Description
apt_apnid	SERIAL	Yes, PK	Unique id. Integer, auto-increment. Not used by TiVo.
apt_prg_programid	INT	Yes	Referees to prg_programid from table tbl_programs.
apt_altprgtitle	VARCHAR(64)	Yes	Alternative name (i.e. 'Dawsons Creek')

9.7 Table tbl_trimtitles

Attribute	Datatype	Mandatory	Description
trm_trimtitleid	SERIAL	Yes, PK	Unique id. Integer, auto-increment. Not used by TiVo.
trm_trimtitle	VARCHAR (64)	Yes	The part of the title which needs to match (i.e. serie:, afl: etc.)
trm_isepisode	BOOLEAN	No	Is this trimtitle episodic (leave empty for unknown, '0' for no, '1' for yes).

9.8 Table tbl_guidedata

Attribute	Datatype	Mandatory	Description
gui_guideitemid	SERIAL	Yes, PK	Unique number for every broadcasting. Used by TiVo. Read from numbers file (see script documentation) which number to use next.
gui_prg_programid	INT	Yes	Reference from the attribute with the same name from the table 'programs'
gui_sta_stationid	INT	Yes	Reference from the attribute with the same name from the table 'stations'
gui_starttime	INT	Yes	Time when the program starts, seconds since 00:00 on the date. Both the Time and Daynum fields are local time in the timezone from where the data came from.
gui_duration	INT	Yes	Duration of the broadcasting in seconds.
gui_daynum	INT	Yes	Date of the program as the number of days since 1970.
gui_title	VARCHAR(64)	Yes	Textual program-title (i.e. 'Dawson's Creek')
gui_episodenr	SMALLINT	No	The episodenummer (if available)
gui_episodename	VARCHAR(64)	No	Textual episode name, arbitrarily long, single line
gui_description	TEXT	No	Textual description of the program, arbitrarily

			long, single line
gui_year	YEAR	No	Year of the movie, four digits
gui_isepisode	SMALLINT	No	1 or 0, indicating if program is episodic or is not episodic.
gui_actors	VARCHAR(255)	No	Comma separated list of actor names
gui_director	VARCHAR(64)	No	Director's name
gui_premierenr	SMALLINT	No	Number between 1 to 5, representing: Season Finale, Series Finale, Premiere, Season Premiere, Series Premiere.
gui_livenr	SMALLINT	No	Number between 1 to 3, representing: Delay, Tape, Live.
gui_bits	INT	No	Decimal integer representing these bits: CloseCaption: 0x1 Stereo: 0x2 Subtitle: 0x4 JoinedInProgress: 0x8 CableInTheClassroom: 0x10 Sap: 0x20 Blackout: 0x40 Intercast: 0x80 ThreeD: 0x100 Repeat: 0x200 Letterbox: 0x400 SexRating: 0x10000 ViolenceRating: 0x20000 LanguageRating: 0x40000 DialogRating: 0x80000 FvRating: 0x100000
gui_advisory	VARCHAR(21)	No	Comma separated list of numbers between 1 to 10, representing: Language, GraphicLanguage, Nudity, BriefNudity, GraphicViolence, Violence, MildViolence, StrongSexual, Rape, AdultSituations.
gui_mpaarating	SMALLINT	No	Number between 1 to 6, representing: Y7 (Youth >= 7 years), Y (Youth), G (general audience), PG (parental guidance suggested), 14 (>= 14 years), M (mature). See also this link .
gui_starrating	SMALLINT	No	Starrating. Number between 1 to 7, representing: one or *, one point five or *+, two or **, two point five or **+, three or ***, three point five or ***, four or ****.
gui_status	CHAR(1) DEFAULT 'N'	Yes	1 character ENUM. "N" is new, "C" is checked, "E" is enriched.
gui_whochanged	VARCHAR(16)	Yes	Who changed the record last
gui_whenchanged	TIMESTAMP WITHOUT TIMEZONE DEFAULT now()	Yes	When was the record last changed (updated when the program was found scheduled)

9.9 Table tbl_genres

Attribute	Datatype	Mandatory	Description
gen_genreid	SMALLINT	Yes, PK	Unique id, static values used by TiVo.
gen_category	VARCHAR(24)	Yes	The human-readable category (i.e. Movie, DocumentaryGroup, etc.)
gen_type	VARCHAR(8) DEFAULT 'broad'	Yes	ENUM. 'broad' or 'specific'

9.10 Table tbl_rel_stationgenre

Attribute	Datatype	Mandatory	Description
stg_sta_stationid	INT	Yes, combined PK	Reference to sta_stationid from the table 'tbl_stations'

stg_gen_genreid	SMALLINT	Yes, combined PK	Reference to gen_genreid from the table 'tbl_genres'
-----------------	----------	------------------------	--

9.11 Table *tbl_rel_proggenre*

Attribute	Datatype	Mandatory	Description
pge_prg_programid	INT	Yes, combined PK	Reference to prg_programid from the table 'tbl_programs'
pge_gen_genreid	SMALLINT	Yes, combined PK	Reference to gen_genreid from the table 'tbl_genres'

9.12 Table *tbl_rel_trimgenre*

Attribute	Datatype	Mandatory	Description
trg_trm_trimtitleid	INT	Yes, combined PK	Reference to trm_trimtitleid from the table 'tbl_trimtitles'
trg_gen_genreid	SMALLINT	Yes, combined PK	Reference to gen_genreid from the table 'tbl_genres'

10 Know that we have these tables, how do you feed the info into this wonderful database?

Good question. We need data. We can parse it using one of the many available parsers. These parsers grab data from popular tv-guide websites. We can also use the existing XMLTV-file for the Netherlands. When we have the XMLTV-files, we need a (XMLTV-)parser which reads the data and feeds it into the database. When that has been done, we can 'enrich' the data by creating a script to check for genre-information, and to check the guide-data against the IMDB (www.imdb.com) for movie-information (actors, directors, ratings etc.)

11 And how do you generate the slice-files?

We have another program which reads which providers are present in the database and generates the according headend- and guide-slices per provider for us.

12 The scripts

DutchTiVo-guide is build around a central database. Around this database the following programs/scripts exist:

- Feeder (*dtgfeeder*)
- Data-enricher (*dtgenricher*)
- Slice-generator (*dtgslicer*)
- Integrity checker (*dtgintegrity*)
- Maintainer (*dtgmaintainer*)

A description of the scripts is given in the following paragraphs.

12.1 Feeder

The feeder has two functions:

1. Checks for new programs against the database and fills the database with new found programs.
2. Feeds the database with guide-data.

The feeder also performs advanced recognition on each program. It determines genres and other extra info from the data in the XML-file. The feeder is written in Perl and uses XMLTV-files as source. There is some error-checking done in here, to prevent pollution of the database by keeping wrong or overlapping data out of the database.

12.2 Data-enricher

The data-enricher enriches the guide-data in the database by adding stuff like genres, ratings, actors, directors etc. from various (web-)sources like yahoo and the imdb (internet movie database).

The data-enricher is written in Perl. Currently the script supports starratings, director and year. More can be implemented.

12.3 Slice-generator

The slice-generator generates slice-files from various tables in the database. The slice-generator dynamically generates head-end and guidedata-slices for every provider in the database. The slice-generator also generates the fake guidedata for the NoNag-channel (this guidedata is not stored in the database). Support for logo-slices and message-slices can be expected in future versions of the slice generator.

The slice-generator is written in Perl. Currently it does some basic integrity-checking. The script is enhanced as soon as this becomes necessary.

12.4 Integrity checker

The integrity-checker checks the database for inconsistencies in the different tables (programs without genres, problems with referential tables, overlapping programs, non-existing references) and for other known errors and reports them.

The integrity-checker is written in Perl.

12.5 Maintainer

The maintainer presents a web-interface which makes it possible to maintain (add/remove/edit) all the important data in the database. This reflects station, program, genre, trimtitle, provider and line-up data.

The maintainer is going to be written in Ruby. This interface is currently in development.

13 More information

If you are hungry for more information, please see our Dutch TiVo community website at: www.nltivo.net.

This website contains all the information that is needed to get a TiVo running in The Netherlands. You will also find things like a roadmap, faq's, howto's, problems and solutions, explanations and a comparison to alternative solutions on the website.