

XMLTV Project



Birmingham Linux User Group
17th November 2011

Nick Morrott

Plan for talk

- XMLTV Project overview
- Sources of TV listings
- XMLTV basics
- Grabber internals
- Filtering data
- Managing lineups

Part 1

XMLTV Project Overview

What is XMLTV?

The XMLTV Project - a collection of Perl modules, grabbers and utilities to obtain, manipulate and search TV listings;

XMLTV.pm - creates XMLTV TV listings;

xmltv.dtd - an XML format describing TV listings;

Project History

Initial release made in 2000

Moved to sourceforge.net in 2001

Current release 0.5.61 (as of 11/2011)

Project Structure

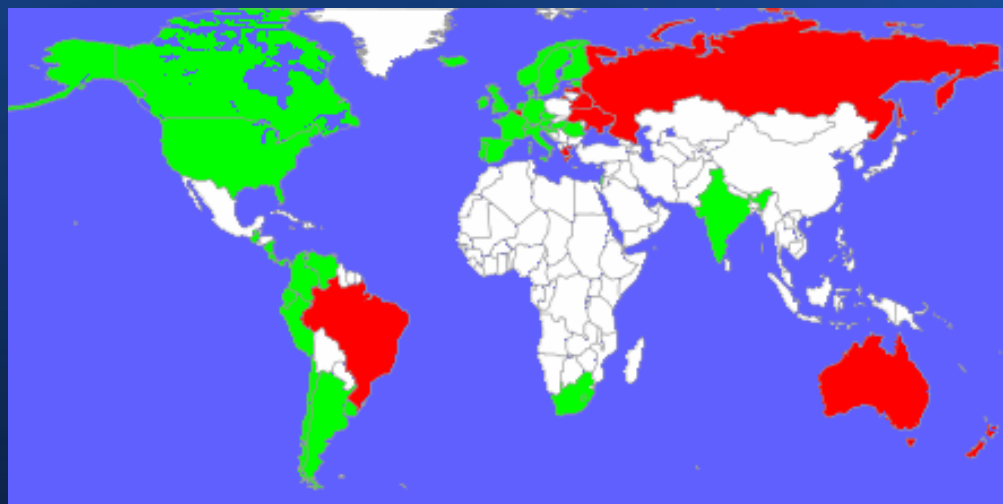
Few developers handling core modules/releases

25 grabbers serving 20+ countries maintained independently from core modules

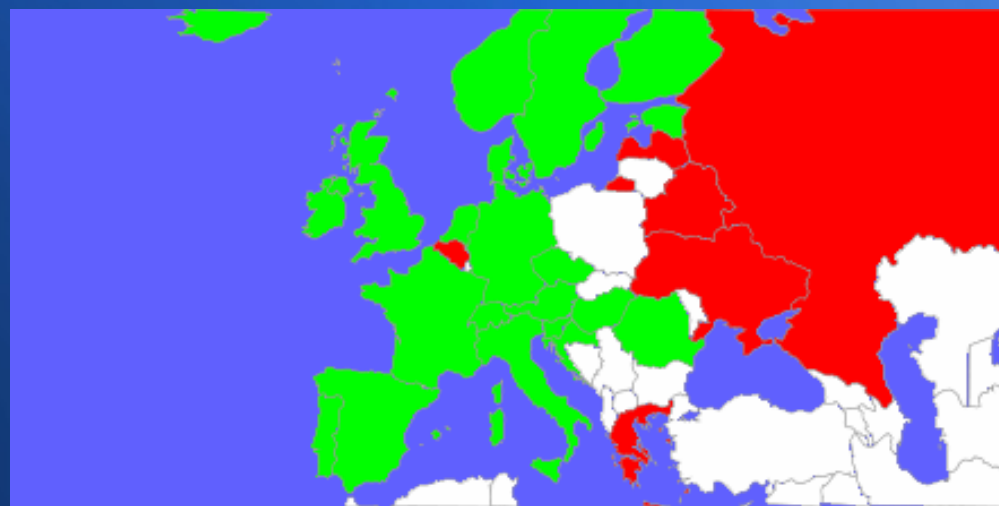
GPL v2 licensed

Releases typically made 2-3 times/year

Global Coverage



World



Europe

Note that for those countries with no **official XMLTV grabber** coverage, **unsupported 3rd-party grabbers** may provide listings

Personal Involvement

Started contributing in 2005

Maintaining UK Radio Times grabber since 2007

Rewrote French grabber after source site updated in early 2010

Nearing completion on channel lineups support

Who uses XMLTV Data?

PVR applications (MythTV, Freevo...)

Listings viewers (FreeGuide, OnTV...)

Scripts/tools filtering XMLTV data directly

Part 2

Sources of TV listings

Sources of Listings Data

Pre-formatted XMLTV data (tv_grab_sw_swedb)

Machine-readable data (tv_grab_uk_rt)

Screen-scraping listings site (tv_grab_fr)

EIT broadcast data (tv_grab_it_dvb, via Linux::DVB)

tv_grab_uk_rt (Radio Times)

Richest source of data for UK users

Uses Radio Times XMLTV data service

Listings for >450 channels

Location-aware setup (postcode/TV service)

Significant “data cleansing” to improve listings

tv_grab_uk_rt - advantages

14 days of listings for all channels

Consistent and rich data

Support for Sky/Virgin/UPC Ireland pay channels

Data is “free as in beer” for **home use**

tv_grab_uk_rt - disadvantages

No radio channels

Data generated once per day (~07:45)

Can be cumbersome to configure

New channels → reconfigure XMLTV manually

Alternatives to XMLTV

i) EIT (“over-the-air”) listings:

- supported in several PVR apps
- broadcast on Freeview and Freesat
- 8 days of listings with frequent updates

ii) Digiguide (\$\$\$) / BBC Backstage (BBC only)

Over-the-air listings

Uses the Event Information Table (EIT) which is transmitted in the transport stream on each of the 6 UK post-DSO Freeview muxes (5 SD/1HD) and also on Freesat muxes

Can be interrogated using the dvbsnoop tool, but supported natively in MythTV, etc

Over-the-air listings: BBC DRM

With the launch of BBC One HD on Freeview HD in post-DSO areas, the BBC want to apply DRM to the listings data and require manufacturers to obtain a suitable license to use it

The listings data is to be encrypted via the use of Huffman tables, which are already available in MythTV/vdr and used for the Freesat EPG.

Over-the-air listings: BBC DRM

- Video/audio streams will NOT be encrypted - once again a DRM scheme does nothing to combat piracy but actively frustrate end-users and restrict choice of receiver availability
- Requested by large US content providers, who cannot restrict listings in this way in the US

MythTV users are unlikely to be affected

Part 3

XMLTV basics

Installing XMLTV

If installing from distro-supplied packages, just use your favourite package manager (XMLTV is a prereq for many PVR apps):

- `aptitude install xmltv`
- `yum install xmltv`

Building XMLTV

XMLTV binaries available for most distros

but

Typical build process from cvs/tarball:

```
$ perl Makefile.PL PREFIX=/usr/local/
```

```
$ make
```

```
$ make test
```

```
# make install
```

Configuring a grabber

Select desired channels:

```
$ tv_grab_uk_rt --configure (defaults to ~/.xmltv/)
```

Grab the data (daily via cron):

```
$ tv_grab_uk_rt --output listings.xml
```

Grabber Capabilities

```
$ tv_grab_uk_rt --capabilities
```

- baseline (quiet, output, days, offset)
- manualconfig
- tkconfig
- apiconfig
- cache
- preferredmethod
- lineups (a work in progress...)

apiconfig – XML-based config

Supported by some grabbers

Stage-based configuration using XML

Allows for easier configuration

Not really implemented in end-user apps though...

XMLTV Utilities

tv_grab_combiner

- run multiple grabbers and combine listings

tv_grep

- extract programmes/channels from an XMLTV file

tv_cat

- concatenate several XMLTV files together

tv_find_grabbers

- find all installed XMLTV grabbers (core and 3rd party)

(and tv_sort / tv_split / tv_imdb / tv_to_latex...)

XMLTV DTD

Developed by XMLTV Project, also used by 3rd party applications

Alternative to TV-Anywhere format

Simple: <channel> and <programme> elements, sub-elements cover attributes

Internally validated by XMLTV.pm

XMLTV Data Structure

List of four elements:

- i) character encoding used (string)
- ii) attributes of the root <tv> element (hash)
- iii) <channel> elements (hash)
- iv) <programme> elements (list)

XMLTV Data Structure (2)

Internal data structure will be something like:

```
[ 'UTF-8',  
  { 'source-info-name' => 'Ananova', 'generator-info-name' => 'XMLTV' },  
  { 'radio-4.bbc.co.uk' => { 'display-name' => [[ 'en', 'BBC Radio 4' ],  
                                          [ 'en', 'Radio 4' ],  
                                          [ undef, '4' ]],  
    'id' => 'radio-4.bbc.co.uk' },  
  ... },  
  [ { start => '200111121800', title => [ [ 'Simpsons', 'en' ] ],  
    channel => 'radio-4.bbc.co.uk' },  
  ... ]  
]
```

Part 4

Grabber Internals

Grabber Internals - Overview

Grabbers must allow for **configuration, listing channels and grabbing data**

Encouraged to use `ParseOptions()` from `XMLTV::Options` to simplify development

`ParseOptions()` provides direct access to runtime options and grabber configuration

ParseOptions()

Implements all required functionality **except configuration, listing channels and grabbing data**

```
my( $opt, $conf ) = ParseOptions( {  
    grabber_name => "tv_grab_uk_rt",  
    capabilities => [qw/baseline manualconfig apiconfig/],  
    stage_sub => \&config_stage,  
    listchannels_sub => \&list_channels,  
    version => 'v 1.301 2010/10/10 17:38:45',  
    description => "Radio Times (UK)",  
} );
```

Grabber Internals - Skeleton

```
#!/usr/bin/perl -w
```

```
=pod
```

```
Your documentation here...
```

```
=cut
```

```
use strict;  
use XMLTV::Options qw/ParseOptions/;
```

```
my( $opt, $conf ) = ParseOptions( {...} );
```

```
# Get the actual data and print it to stdout.
```

```
if( $is_success ) {  
    exit 0;  
}  
else {  
    exit 1;  
}
```

```
sub config_stage {...}
```

```
sub list_channels {...}
```


XMLTV.pm

Cornerstone module of the project

Handles all XMLTV data I/O

Uses specific handlers to validate content

Handlers include with-lang, episode-num, video, audio, rating, credits, scalar, length, icon

Reading XMLTV data

```
use XMLTV;
```

```
my $data = XMLTV::parsefile('tv.xml');
```

```
my ($encoding, $credits, $ch, $progs) = @$data;
```

Writing XMLTV data

```
use XMLTV;

my $w = new XMLTV::Writer(encoding => 'UTF-8');
$w->comment("Hello");
$w->start({ 'generator-info-name' => 'test-gen' });

# write a single channel
my %ch = (id => 'test-channel',
          'display-name' => [ [ 'Test', 'en' ] ]);
$w->write_channel(\%ch);

# write a single programme
my %prog = (channel => 'test-channel',
            start => '200203161500',
            title => [ [ 'News', 'en' ] ]);
$w->write_programme(\%prog);

$w->end();
```

Useful XMLTV modules

XMLTV::Supplement

- retrieve files such as channel lists from XMLTV server

XMLTV::DST

- handling for daylight savings timings

XMLTV::Get_nice

- inject random delays in successive HTTP retrievals

Useful core/3rd party modules

Encode
POSIX

LWP::UserAgent (and other LWP modules)

HTML::Entities

HTML::TreeBuilder

HTTP::Cache::Transparent

Date::Manip

HTML::TreeBuilder

```
use HTML::TreeBuilder;
use XMLTV::Get_nice qw(get_nice);

my $content = get_nice($url);
$content = decode_utf8($content);
my $tree = new HTML::TreeBuilder;
$tree->parse($content);
$tree->eof;

foreach my $cell ( $tree->look_down( "_tag", "td",
                                     "class",
"channel" ) ) {
    my $img = $cell->look_down( "_tag", "img" );
    my $chname = trim( $img->attr('alt') );
    ...
}

$tree->delete(); undef $tree;
```

Date::Manip

```
my $strDate = ParseDate( "20100301120000 +0000" );
my $strDelta = ParseDateDelta( "5minutes" );

my $date = DateCalc( $strDate, $strDelta );

my $unixDate = UnixDate( $date, "%Y%m%d%H%M %z" );

if ( Date_Cmp( $dateStart, $dateStop ) < 0 ) {
    print "Start date is earlier than stop date!";
}
```

Part 5

Filtering data

Filtering XMLTV data

- `tv_check` – takes a listings file and a list of target shows to check and creates text/HTML output of matching showings
- `tv_grep` – takes a full listings file and filters it against a provided Perl RE, producing new XMLTV output containing only matching programmes

Part 6

Managing lineups

Problem

When using XMLTV with a large number of configured channels (Virgin TV, Sky...) in an end-user application, lineup management becomes a pain:

- Channel can be added to EPG
- Channels can be removed to EPG
- Channels can move around the EPG
- Icons/channel names change over time

Lineup management strategies

- Manual editing of lineup configuration
- Providing lineup updates via a script
- “Intelligent” lineup management

Manually updating config

- An XMLTV grabber uses a simple configuration file (e.g. `tv_grab_uk_rt.conf`) to store configured channels and options
- A consuming application may have more complicated channel config (e.g. MythTV uses a MySQL database)
- Manually updating entries is therefore not simple and quite daunting to the casual user

Scripting updates

- Use of a shell/SQL script permits multiple channel configurations to be updated in one hit
- Script will typically contain raw SQL UPDATE statements and “frozen” XMLTV channel configs

Must be updated whenever channels are added/removed/renumbered from EPG

- Fragile if either the DB schema or XMLTV channel configs change – requires rewriting

Intelligent lineup management

- Goal of the “lineups” capability I am working on for XMLTV is to permit automagical updating of lineup changes
- Support for Freeview/Freesat/Sky/Virgin/UPC (at least) in the UK/Eire
- Requires “some” management of channel mappings, which can be managed within the XMLTV Project for grabbers using this feature

Intelligent lineup management

For `_rt` grabber, 3 data sources are currently used:

- `tv_grab_uk_rt` `channel_ids` file
- XMLTV ID:DVB SID mappings for Freeview/sat
- Platform EPG listings (Wikipedia)

Use of the WP source means that lineup changes are reflected very quickly

• XMLTV config changes

- A user's selected platform (e.g. @freeview) is stored in the XMLTV config, instead of a long list of individual channels
- When the grabber detects channels have been added/removed, the output XML data is updated automatically
- Will limit amount of editing of XMLTV config required

• XMLTV config changes

- New lineups capability will add --list-lineups and --lineup options to grabber
- --lineup option will produce XML lineup data (based on new XML Schema) that includes rich data for all channels available on platform and XMLTV/DTV settings where available
- End-user app can read this data and update channel configuration internally with no end-user input required

MythTV config changes

- Current development release of MythTV (0.25) includes new Services API permitting programmatic creation/update/deletion of channels/video sources and capture cards
- Working plan is to integrate support for using the XMLTV apiconfig and lineups capabilities to allow existing channels to be updated automagically and for entire new video sources to be created where no tuning data exists (e.g. Sky/Virgin)

Useful Links

Wiki

<http://www.xmltv.org>

Code (CVS/tarball)

<http://sourceforge.net/projects/xmltv/>

Mailing list

<http://lists.sourceforge.net/lists/listinfo/xmltv-users>

Useful Links

dvbsnoop

<http://dvbsnoop.sourceforge.net>

BBC DRM article

<http://www.guardian.co.uk/technology/2011/nov/14/bbc-hd-drm>

Useful Links

Freeview/Freesat update scripts for MythTV

http://www.mythtv.org/wiki/Uk_xmltv

mythtv_channel_import

https://github.com/ollyg/mythtv_channel_import

Thanks for listening!

Questions?

Nick Morrott
knowledgejunkie at gmail dot com