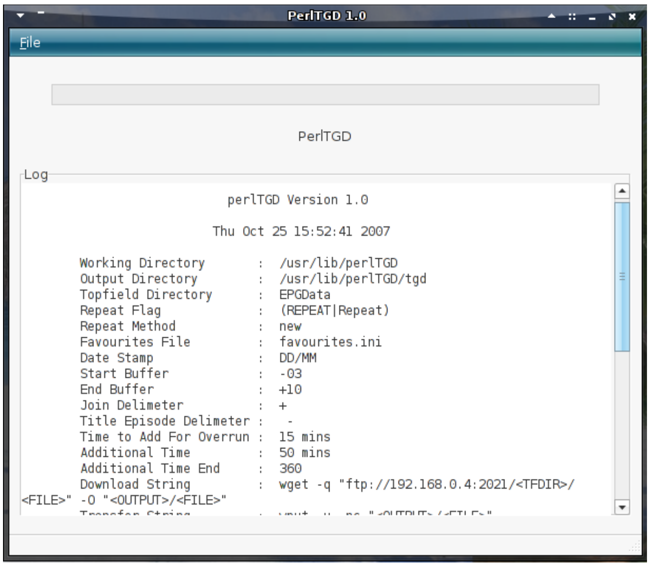
PerlTGD

Perl Topfield Guide Daemon

Manual

rev 0.2



```
                    perlTGD Version 1.0

                  Thu Oct 25 15:52:41 2007

        Working Directory       :  /usr/lib/perlTGD
        Output Directory        :  /usr/lib/perlTGD/tgd
        Topfield Directory      :  EPGData
        Repeat Flag             :  (REPEAT|Repeat)
        Repeat Method           :  new
        Favourites File         :  favourites.ini
        Date Stamp              :  DD/MM
        Start Buffer            :  -03
        End Buffer              :  +10
        Join Delimeter          :  +
        Title Episode Delimeter :   -
        Time to Add For Overrun :  15 mins
        Additional Time         :  50 mins
        Additional Time End     :  360
        Download String         :  wget -q "ftp://192.168.0.4:2021/<TFDIR>/
<FILE>" -O "<OUTPUT>/<FILE>"
```

* GTK2 version shown

Table Of Contents

To Do

- Improve manual and add more regular expression and favourite creation examples.

Releases

1.0 Beta D – Fixed bug with conflict resolution.
1.0 Beta C - Added mei2tgd to package. Add -onlytimers switch to allow for using broadcasted EPG as source to search. Use eit2mei TAP on TAPWorld to create the Freeview.mei file.
1.0 Beta B – Added capturing of output to download and makeguide components.
1.0 Beta A – Initial Release

Disclaimer

By using this package of perl scripts you are doing so at your own risk. I will not be held responsible for any problems that may arise from the use of these tools provided. Missed timers, Removal of current timers or anything this package does that results in you missing your favourite shows is your own fault as you have been warned.

Out of courtesy, ask permisson when modifiy any part of this package for purpose of redistributing.

Like others that have developed applications both TAPs and for the PC, I have done so in my spare time. I have no intention of spending hours and hours fixing and helping others, but saying that I will do my best.

This package will always be “free, as in beer”.

All this is released under GPL V2 and by using you are agreeing to adhere to the licencing terms.

PerlTGD

Introduction

PerlTGD is an all-in-one solution to updating the EPG and setting of timers on the Topfield range of PVR's for Unix based PC's and devices (nslu2 and Iceboxes). This applications handles the downloading*, producing of guides*, searching for favourites shows/keywords creating timers and uploading of tgd documents to the Topfield PVR*.

* By external system calls

What are the key features of perlTGD?

- Ability to download guide.*
- Ability to produce TGD documents.*
- Searches for favourites/keywords in TGD documents.
- Can increase duration of shows that are themselves or are after shows which run over there assigned timeslot.
- Can adjust duration of shows between a certain time frame i.e. Early morning to combat networks early morning nonesense.
- Checks for timer conflicts.
- Drops timers of lower priority.
- Can adjust timers when conflicting.^
- Joins sequential timers together.
- Automatically names timers with Show – Episode instead of Show DD/MM if show has an episode name and not joined to any other recording.
- Can append any datestamp using DD, MM, YY and YYYY to recordings.
- Can append timers that will not be found by searching.
- Removes changed timers from PVR.'
- Transfers all TGD documents to PVR.*
- Can reboot unit.*

* Uses external system calls
^ Use with caution as may contain significant bugs
' Can be disabled

What are the other scripts included in the package for?

These other scripts have been written to easily manage various features of perlTGD. Below is a brief description of what they are

PerlTGD Favourites Manager (perltgdfav.pl) is a GUI tool to help in the creation and maintance of your favourites list.

PerlTGD Append Manager (perltgdappend.pl) is a GUI tool to manage your list of timers to append to the timer file for shows that will not be picked up by searching.

PerlTGD Overrun Manager (perltgdoverrun.pl) is a GUI tool for managing the list of shows which run over there assigned timeslot.

PerlTGD Settings Manager (perltgdsettings.pl) is a GUI tool for creating or changing the settings to be used by perlTGD n.b. Still under construction but contains basic feature set so has been included.

Does it work with Ice TV?

Yes. To use Ice I recommend that you download the xmltv2tgd script that i wrote (Download from http://jim16.110mb.com/index.html). If you don't download and use this script you will need to find some method of generating TGD documents from your Ice XML document as this package searches TGD formatted strings to match your favourite shows/key words. **UPDATE: Package now includes xmltv2tgd and mei2tgd for FTA guide**

Are there any Windows GUI tools to set favourites etc... ?

Not currently. I don't run Windows so have not programmed any interfaces for the various compnents of the package. If you are a Unix based user you have the GTK2 scripts that i have written but I have not intention on porting these to an operating system which I don't use. If you wish to volunteer your time and skills I am happy for anyone to develop GUI tools for windows users. PM me on the topfield forums if you wish to develop such GUI tools.

- Eventually I would like to see a web interface on the device setup so you can manage the various sections of the package thru your web browser but haven't the time to do so for the forseable future.

Is there a TAP to control my favourites?

Yes. See the topfield australia forums for a guide Tap called JimsGuide, It contains a favourites manager design for perlTGD and contains many other useful features.

System Requirements

Topfield

Epg_Uploader >= 3.15rls2 by tonymy01. Recommend Latest 3.23 as of 04/11/2007

Download from : http://tonyspage.abock.de/

PC/Mac

Base Install

Unix/Linux based system

Perl > 5.8 * Only tested with this version but may work with earlier

ftpd-topfield/puppy * Some program to communicate with pvr via terminal

For GUI Version and Tools

libgtk2-perl

libgtk2-ex-simple-list-perl

* Mac – I'm currently unsure on the avaliability of GTK2 on Mac so the GUI's might not be avaliable.

Slug and Icebox2

This application works on a Slug/Icebox with perl is installed. Please see the installation instructions for slug/icebox to guide you through the installation.

Windows

Sorry but due to your systems having plentiful numbers of programs including the well established TEDs Suite this application is not avaliable.

Installation

PC

- This is a suggestive install procedure and you are free to install where and how you want.

- Feel free to try install-linux.sh script to install perltgd. Ensure to run as root or under sudo. Currently all files are installed but no shortcuts are made. If you wish to uninstall from this call perltgduninstall and all files and folders installed will be removed. Tested under fresh install of Ubuntu 7.10 (Gutsy) and all required GTK2 packages are installed by default. Will need to manually install wget/wput and ftpd-topfield or puppy. If reported errors relate to GTK2 then install the required GTK2 packages.

1. Extract the archive

2. Move the following files to /usr/bin/ and drop the .pl extension

   perltgd.pl
   perltgdcli.pl (Copy of perltgd for the command line. Can be omitted if desired)
   perltgdfav.pl
   perltgdappend.pl
   perltgdoverrun.pl
   perltgdsettings.pl

   * Ensure these files have executable status - sudo chmod +x perltgd* etc...

3. Make the following directories

   /usr/topfield/perltgd
   /usr/topfield/perltgd/tgd

   Copy the contents of the perlTGD folder in this zip file to /usr/topfield/perltgd

   * Ensure you and anyone else who will use this program has the required premissions to access these directories.

4. For the GUI version and tools install the following packages libgtk2-perl, libgtk2-ex-simple-list-perl (These are the Ubuntu package names so may differ for your distro, OS)

5. Next test that all the dependencies are installed properly by launching the main program

In a console type perltgd&

If no error messages are displayed in the console and a GUI is created then all the requirements are met and you only need to create some shortcuts and you are done. If no GUI is created but no error message is displayed then GTK2 is not installed. Perltgd will only run in commandline form until GTK2 is installed.

6. Create shortcuts with the command to be executed as the following.

PerlTGD

```
perltgd -w=/usr/topfield/perltgd -a
```

* See PerlTGD usage to determine switches

Favourites Manager

```
perltgdfav -file=/usr/topfield/perltgd/favourites.ini -channels=<YOUR CHANNELS HERE>
```

Format - > Name**:**Name**:**Name**:**Name**:**Name
These must match the channel names in the tgd documents.

e.g. -channels=ABC:SBS:Ten:Seven:NINE:ABC2 etc...
-channels=2:3:10:7:9:21 etc.. for LCN numbers as channel name

* Can be launched without any switches but no file will be loaded automatically and channels dialog will contain no channel names.

Append Manager

```
perltgdappend -file=/usr/topfield/perltgd/append.timers
```

Overrun Manager

```
perltgdoverrun -file=/usr/topfield/perltgd/overrun.shows
```

Settings Manager

```
perltgdsettings -file=/usr/topfield/perltgd/perltgd.settings
```

* Include is a default settings file to get you started. Should only need a little tweaking to have it working as you require.

Gnome Users

```
sudo gedit /usr/share/applications/perltgdfav.desktop
```

Enter the following in the file replacing the -channels= with those appropriate to you

```
[Desktop Entry]
Version=1.0
Encoding=UTF-8
Name=PerlTGD Favourites Manager
Comment=Favourite Manager for PerlTGD
Exec=perltgdfav -file=/usr/topfield/perltgd/favourites.ini -channels=ABC:SBS:SC10:Prime:WIN:ABC2
StartupNotify=true
Terminal=false
Type=Application
Categories=Application;Topfield;
```

Create perltgdsettings.desktop and perltgd.desktop files changing the Exec lines as appropriate in the same directory.

Now the base install should be complete and you should use the shortcuts you created in step 6 to launch the various programs in the package.

Modify your perltgd.settings file and add favourites and you have a complete installation.

Installation

Nslu2/Icebox *Currently under construction. Use as rough guide.

Install Prerequisites

Install either perl to run the application

```
ipkg install perl
```

Install either puppy or ftpd-topfield for transfering of files

```
ipkg install puppy
ipkg install ftpd-topfield
```

* Probably already have installation on unit

* May want to install wget and wput – ipkg install wget-ssl/wput, if using ftpd-topfield

Install perltgd

Create the directory for perltgd

```
mkdir -p /opt/usr/local/perltgd
```

Create the directory for output

```
mkdir -p /opt/usr/local/perltgd/tgd
```

Transfer files in perlTGDslug folder to /opt/usr/local/perltgd (perltgd.settings, append.timers, overrun.shows and favourites.ini) and perltgdcli.pl. Ensure perltgdcli.pl has executable permission e.g. chmod +x perltgdcli.pl

Edit the perltgd.settings file for the transfer methods required i.e. puppy, wget/wput with ftpd-topfield. Also alter the tfdir value to be the location on the PVR to place the EPG files. (See Settings section of this manual)

If using XML file as source for EPG copy the file xmltv2tgd.pl to /opt/usr/local/perltgd and ensure it has executable permissions e.g. chmod +x xmltv2tgd.pl. Setup the makeguide setting in the settings file to convert the xml file to tgd documents

IceTV e.g.

```
makeguide::/opt/bin/perl /opt/usr/local/perltgd/xmltv2tgd.pl
-f=iceguide.xml -gmt=11:00 -o=/opt/usr/local/perltgd/tgd
```

* Read xmltv2tgd readme for switch details.
* Please ensure a full path to the utility is used

Setup the downloading of the EPG from the internet.

Create a script called downloadepg.sh in /opt/usr/local/perltgd/

```
#!/bin/sh
wget <YOUR SOURCE ADDRESS>
```

Set the settings file to

```
grabber::./downloadepg.sh
```

* If file is zipped then add -> unzip xmltv.zip, to the downloadepg.sh or use the inbuilt download and extract of xmltv2tgd.
* Please ensure a full path to the utility is used

Test that perltgd runs correctly

```
/opt/bin/perl /opt/usr/local/perltgd/perltgdcli.pl -w=
/opt/usr/local/perltgd
```

If no errors are reported then perltgd is successfully installed.

Next you may want to setup your favourites so edit favourites.ini and add you required terms (Read Favourites section of this manual for the format).
Also look at adding show names to overrun.shows and add any appending timers you require to append.timers.

See automation next to have it run automatically everyday.

Automation

Create the following file with your required switches (See perltgd usage).

/opt/usr/local/perltgd/perltgd-run.sh

```
#!/bin/sh
perltgdcli.pl -w=/opt/usr/local/perltgd -a
```

Make script executable

```
chmod +x /opt/usr/local/perltgd/perltgd-run.sh
```

Final step is to setup the scheduled cronjob to run automatically. (Assumes you already have cron setup)

```
crontab -e
```

Add entry for perltgd to run automatically.

```
0 7 * * * /opt/usr/local/perltgd/perltgd-run.sh >> /dev/null
```

- 0 represents the minutes of the hour to run
- 7 represents the hour i.e. 7 am

Change these values as required to run when you want

* You may need to setup a wakeup timer for your PVR so it is awake when this script runs.

Files Created/Used

append.timers

File containing a list of fixed timers to be set. That is timers you wish to be set every week that will not be picked up by the search e.g. Tattslotto on saturday etc..

File Format

Channel **<TAB>** Day of Week **<TAB>** Time (HH:MM) **<TAB>** Duration **<TAB>** Name

* Where dow is 0 for Sunday 6 for Saturday

e.g. Prime 6 20:25 45 Tattslotto

overrun.shows

File containing a list of show names that run over there assigned timeslot. It is used to increase the duration of the shows on the current channel from the point which this show is detected. i.e. Protect timers that run after Big Brother or other shows that never run on time.

File Format

Show Name <New Line>

* Please note: Show name must match exactly the name in the .tgd documents

perltgd.log

File containing the output log of the runs of the PerlTGD application. Every run is appended to the file. If you wish for this file to not have each run appended add “rm perltgd.log” at the end of the script used to launch program.

perltgd.settings

File containing the settings to be used by PerlTGD. Use the program perltgdsettings to set values. If GTK2 is not avaliable just use a text editor like notepad or gedit.

shows.repeat

File containing a list of shows that have been broadcast and there date to be used to detect repeat episodes. Shows are only added to this when they are found and a timer has been created. **WARNING: May contain bugs.**

File Format

Show Name – Episode Name <TAB> Date (YYYYMMDD)

favourites.ini

File containing all favourites to be searched for. Use the included favourite manager program to create favourites. Read the favourites section for more details.

Using PerlTGD

Switches

-a | -all

Run full cycle of downloading, producing guides, searching and transfering

-at | -append

Append timers in the append.timers file

-d | -download

Download the guide

-mg | -maketgd

Produce .tgd documents

-s | -search

Perform search

-o | -output

Output search results to 99999999.tgd timer file

-t | -transfer

Transfer documents to PVR

-r | -reboot

Reboot PVR if not currently active

-ncc

Don't check for connection of PVR. Use for when not connecting the unit directly.

-w

Define working directory i.e. the location of the settings and other critical files.

e.g -w=/usr/topfield/perlTGD

-nogtk

Define application not to use GTK2 and run in command line state

-onlytimers

Only transfer timer file to the PVR.

-noremove

Disable removing of timers from the PVR. **WARNING: May result in missed timers. Only use if you know what you are doing.** Use this if you wish for perlTGD to not remove manually flagged recording on PVR. Also results in perlTGD not removing timers it sets that have changed. When I get some free time I will code perlTGD to track the timers it sets but it's not high on my priority list.

Usage Examples

* Replace perltgd with perltgdcli where appropriate if using on Icebox or Nslu2.

Perform everything

```
perltgd -w=/usr/topfield/perlTGD -a
```

Search and Upload

```
perltgd -w=/usr/topfield/perlTGD -s -o -t
```

Reboot unit

```
perltgd -w=/usr/topfield/perlTGD -r
```

Settings (perltgd.settings)

Example settings file

```
tfdir::EPGData
favfile::favourites.ini
datestamp::DD-MM
startbuffer::-03
endbuffer::+10
join::+
showtitle:: -
grabber::downloadGuide
makeguide::makeTGD
addtime::40
addtimestart::0
addtimeend::360
overrun::15
writelog::yes
transfer::wput -u -nc "<OUTPUT>/<FILE>" "ftp://192.168.0.1:2021/<TFDIR>/<FILE>"
download::wget -q "ftp://192.168.0.1:2021/<TFDIR>/<FILE>" -O "<OUTPUT>/<FILE>"
reboot::puppy -c reboot
outputdir::/usr/topfield/perlTGD/tgd
repeatflag::(REPEAT|Rpt)
repeatmethod::new
lcnname::2-ABC:3-SBS:5-SC10:6-Prime:8-WIN:21-ABC2
ukfavfile::wget -q "ftp://192.168.0.1:2021/ProgramFiles/Auto Start/<FILE>" -O
"<OUTPUT>/<FILE>"
ignoretimers::(^ABC$)
```

Descriptions

tfdir

Directory on Topfield where EPG files are stored. * This must be able to be substituted in the transfer and download string

favfile

Filename of the favourites file to be used.

datestamp

Date stamp to be used for shows which don't have Name – Episode or are joined

Use DD, MM, YY, YYYY combinations and these fields will be automatically populated

e.g. YYYY-MM-DD, DD/MM, DD:MM:YY etc...

startbuffer

Buffer to add to the start of all recordings

Use +/-DD as the format

e.g. +03, -03, -10 etc..

endbuffer

Buffer to add to the end of all recordings

Use +/-DD as the format

e.g. +03, -03, -10 etc..

join

Delimeter to be used for joining of sequential timers

e.g +. /, & etc...

* All spaces will be removed and a space will automatically be placed before and after the defined character/s

showtitle

Delimeter to be used for joining the Show Name and Episode title if avaliable

e.g. -, + etc..

* No trimming is applied to this so if a space is required add it before and after

grabber

System call to the harvester or whatever you use to get the EPG data. Must work when called from a console.

makeguide

System call to produce the .tgd documents. Must work when called from a console.

addtime

This is the amount of time in minutes to be added to timers found that are between the range defined in the addtimestart and addtimeend.

* I use this for shows in the early morning which can often be way out e.g. 30 mins, 1 hour late

addtimestart

Start of the range in minutes to add the addtime value to the duration of those shows

addtimeend

End of the range in minutes to add the addtime value to the duration of those shows

overrun

Number of minutes to add to recordings which either are themselves or are after shows which over run there allocated timeslots.

writelog

Whether to write the perltgd.log file

e.g. yes or no, 1 or 0

transfer

System call to transfer files to the pvr. This is called for every file to be transfered so must be based on individual file transfering.

The following markups are avaliable and the values will be automatically populated

<CURRENT> - Current working directory
<OUTPUT> - Output directory
<FILE> - File to be transferred
<TFDIR> - Directory on the pvr

e.g. `wput -u -nc "<OUTPUT>/<FILE>" "ftp://192.168.0.1:2021/<TFDIR>/<FILE>"`

n.b. ftpd-topfield is running on port 2021

download

System call to download files from the pvr. This is called for every file to be downloaded so must be based on individual file downloading.

The following markups are avaliable and the values will be automatically populated

<CURRENT> - Current working directory
<OUTPUT> - Output directory

<FILE> - File to be transferred
<TFDIR> - Directory on the pvr

e.g. wget -q "ftp://192.168.0.1:2021/<TFDIR>/<FILE>" -O "<OUTPUT>/<FILE>"

n.b. ftpd-topfield is running on port 2021

reboot

System call to reboot the PVR.

e.g. puppy -c reboot

outputdir

Directory where .tgd documents are stored

e.g. /usr/topfield/perlTGD/tgd

repeatflag

RegExp match for each .tgd event to determine whether it is a repeat

repeatmethod

Which determination method is to be used to detect repeats of shows.

Currently only new is avaliable. If you wish not to use this then leave blank.
* New is a method by which all timers set by this application get recorded to shows.repeat and is used to determine whether it is a repeat.

lcnname

String containing the channel names in the TGD documents and there associated LCN channel on your PVR. Used to map channel names to LCN numbers if not using LCN numbers as channel identifiers. If using LCN numbers leave blank as this is not required.
Seperated by : for multple entries. Each entry is of LCN-Name format.
e.g. 2-ABC:3-SBS etc...

ukfavfile

Download string for obtaining uk scheduler format favourite file from the various EPG TAPs that can create the search file. Feature used to remove the requirement to add favourites on the PC or small devices manually. Please note all favourites created on the PVR will be given a priority of 1 i.e. The lowest.

ignoretimers

This is a regular expression used to determine whether to ignore a set timer on the unit and not remove it.

Favourites (favourites.ini)

The favourites file is a list of shows/keywords which you would like to be automatically set. This file is to be placed in the working driectory (e.g. -w=blah).

Favourite Creation

Suggestion: Launch the perltgdfav program and use this to create your list of favourites. (*nix based only and requires GTK2 libraries)

perltgdfav accepts the following switches

-file=blah/favourites.ini
Path and name of the file to be automatically loaded upon start

-channels=ABC:SBS etc...
List of channel names that are used in the tgd documents or LCN numbers.

Format

Name**:**Name**:**Name**:**Name**:**Name

n.b. the '**:**' seperation

* If you fail to launch the program with the -channels= switch then it will still run but without the added help (Dialog with checked buttons) with the creation of the channel field.

Example Usage

```
perltgdfav -file=/usr/topfield/perlTGD/favourites.ini -channels=ABC:SBS:SC10:Prime:WIN:ABC2
```

Manual Creation

If perltgdfav is unavaliable for your platform or you wish to manually create your favourites file listed below is the file format.

* The included favourites.ini file can be used in any spreadsheet program e.g. Excel, OO Spreadsheet etc.. so if you have these programs installed it is recommended you edit with them ensuring all fields are text and when saving select tab seperated

format. To disable a favourite add # at the start and it will be ignored.

Search Term Format

Term <TAB> Channel <TAB> Day <TAB> Location <TAB> Earliest Start <TAB> Latest Start <TAB> Priority <TAB> Repeats <TAB> Adjustable

Description Of Fields

Term

Show name or a RegExp to search for. Remember to escape any regexp special characters like :'\ etc.. if part of text to search for.

e.g.

Australia(.*)Cricket
(AFL|Footy)(.*)(Your team|Someone elses tem)
House

etc...

Channel

All for any channel or a regexp match for the channel names in the .tgd documents e.g. (Prime|SC10). When using lcn numbers as channel names to avoid matching 50 (SC10 HD) for all 5 you will need to use ^5$ to ensure no matching of 50 occurs

Day

All for any, WeekEnd, WeekDay, Mon, Tue, Wed, Thu, Fri, Sat, Sun or 0 (Sunday) to 6 (Saturday) are avaliable e.g. (0|5|6) *Match on Fri, Sat and Sun

Location

Where in the .tgd string to search. Title, TitleExact, SubTitle, Description, All are the values avaliable.

n.b. Use TitleExact when attempting to match exactly the title e.g House to not match House of the Dead only House exactly.

Earliest Start

The time that the show must start on or after
Accepts the following

Any, -1 (Same and Any), HH:MM or 1201 (Time in minutes) e.g. 1140, 19:00, -1, Any

Latest Start

The time that the show must start before

Accepts the following

Any, -1 (Same and Any), HH:MM or 1201 (Time in minutes)

e.g. 1140, 19:00, -1, Any

Priority

Number from 1 to ? (Up to you). This is used to determine what show to drop when there are to many recordings on at the same time i.e. 3 timers at same time on different channels. The higher the priority the less likelyhood of show being dropped i.e. 10 will dropped shows with 9 or less if clashing.

Repeats

0 for record any occurance. 1 to only record new episodes. This works with the selected repeat determination method to determine whether the show has been on before.

Adjustable

0 for no adjustment. 1 for time can be adjusted to fit in with clashing timers. **WARNING:** This is experimental atm and i have no time to work out any bugs that may be found when using this.

Simple Favourite Creation

The most simple way to create a favourite is to use the name of the show as the key word. Other simple methods are to specify the location, time frame and channel name the term must meet. Below are some very simple examples to get you started. If you have any problems creating a favourite feel free to pm me for help and I will provide the required favourite string and a explanation of what I have done.

Examples

House on wednesday.

Would be

House<TAB>Ten<TAB>Wed<TAB>TitleExact<TAB>Any<TAB>Any<TAB>5<TAB>0<TAB>0

Explanation

The term is house and it must be on channel Ten. The show must be on wednesday and the term must exactly be house. This favourite has a priority of 5 (1 is lowest) and to record any occurance but don't adjust duration to avoid clashing.

Only new episodes of House on wednesday

Would be

House<TAB>Ten<TAB>Wed<TAB>TitleExact<TAB>Any<TAB>Any<TAB>5<TAB>1<TAB>0

Explanation

Same as the first example but the second last field is 1 instead of 0. This means that the show will be checked whether it is a repeat and if so will report that we found it but its a repeat so will not record. Uses the specified repeat method and repeat string to determine.

Only House on wednesday nights during a particular time frame.

Would be

House<TAB>Ten<TAB>Wed<TAB>TitleExact<TAB>20:00<TAB>22:00<TAB>5<TAB>0<TAB>0

Explanation

Same as the first example but Any has been replaced by times the show must start on or between. In this example the show must start between 8 and 10 pm at night for it to be a match. This is a great way of having the day time repeats of some shows skipped.

Set timers of movies/shows that a particular actor/actress is in

Would be

```
Actor/ActressName<TAB>All<TAB>All<TAB>Description<TAB>Any<TAB>Any<TAB>2<TAB>0<TAB>0
```

Explanation

The search term is the name or names of the actor/actress. Because the stars is usually located in the description we have set the location to this and set the channel, day and time frame to any and all.

Complex Favourite Creation

Firstly a quick guide to regular expressions

* - Represents 0 or more characters.
+ - Represents 1 or more characters.
(Term|Term|Term) - | is used as an or
^ - Is used to indicate that the string must start with the folowing characters/string/term
$ - Is used to indicate that the string must end in the term
[] - Is used to define a character class
. - Represents any character
\s – Represents space
\d – Represents number

See http://www.cs.tut.fi/~jkorpela/perl/regexp.html, http://www.troubleshooters.com/codecorn/littperl/perlreg.htm and http://perldoc.perl.org/perlretut.html for more information about perl regular expressions.

AND equivalent – Term(.+)Term
OR equivalent – (Term|Term)
NOT equivalent* – This(?!(.*)Not This)

> ** Not is complicated in regular expression when it comes to this but not this or not this but this. Generally it is easier to use the other things such as location, channel and time frame to negate shows being picked up. The display regular expression dealing with this and not this works but I am currently unable to work out not this but this.*

Examples

> *When using complicated matches remember that these only search in the location specified i.e. Title, Subtitle etc... If you wish for multiple areas specify location as all.*

Matching all australia cricket games

Regexp: Australia(.+)Cricket

Explanation

Australia and Cricket are 2 text phrases to look for. Because we don't know and don't really care whats between Australia and Cricket we just say that as long as there is 1 or more characters and both terms are found then its a match.

Matching football games by code and teams

Regexp: `(AFL|Footy)(.+)(Collingwood|Essendon)`

Explanation

Similar to the above example except we have now include 2 or statements. AFL or Footy and Collingwood or Essendon in any combination must be found to be a match

Matching multiple terms

Regexp: `(AFL|Rugby|Soccer)`

Explanation

We are simply looking for a match of any of the strings contained in the brackets.

Match a term only when another term isn't present

Regexp: `Ten(.+)News(?!(.+)Sports)`

Explanation

We are searching for Ten then 1 or more characters before News. The complicated bit here is (?!(.+)Sports). This means that News must not be followed by sports. The (.+) is needed as we don't know how many characters inbetween News and Sports nor do we care.