NAME

make_shownotes - Make show notes for the Hacker Public Radio Community News show

VERSION

This documentation refers to **make_shownotes** version 0.4.5

USAGE

```
    make_shownotes [-help] [-documentation|-man] [-config=FILE]
        [-from=DATE] [-[no]comments] [-[no]silent] [-[no]mailnotes]
        [-lastrecording=DATETIME]
        [-full-html=FILE] [-html=FILE] [-json=FILE]
        [-debug=N]
```

OPTIONS

-help

Displays a brief help message describing the usage of the program, and then exits.

-documentation or **-man**

Displays the entirety of the documentation (using a pager), and then exits.

To generate a PDF version use:

```
    pod2pdf make_shownotes --out=make_shownotes.pdf
```

-config=FILE

The script uses a configuration file to hold the various parameters it needs to run. This option allows an alternative configuration file to be used. This file defines many settings including the location of the database.

See the CONFIGURATION AND ENVIRONMENT section below for the file format.

If the option is omitted the default file is used: **.make_shownotes.cfg**, which is expected to be in the same directory as the script itself.

-from=DATE

This option is used to indicate the month for which the shownotes are to be generated. The script is able to parse a variety of date formats, but it is recommended that ISO8601 **YYYY-MM-DD** format be used (for example 2014-06-30).

The day part of the date must be present but is ignored and only the month and year parts are used (to internally denote the first day of the month).

If this option is omitted the current month is used. Of course, this may cause problems if the notes are to generated for an earlier (or later) month, which is why this option exists.

-[no]comments

This option controls whether the comments pertaining to the selected month are included in the output. If the option is omitted then no comments are included (**-nocomments**).

Output file options: **-html=FILE**, **-full-html=FILE**, **-json=FILE**

There are three output file types that can be generated by the script. At least one must be present:

HTML fragment (**-html=FILE**)

This file will contain the HTML to be added to the HPR database. The page for the show, when it is released, will be a full web page, with standard header and footer, and the contents will come from this HTML fragment in the database.

Action will be needed in addition to the script to add this file to the database, but how this is done is outside the scope of this documentation.

Standalone HTML (**-full-html=FILE**)

The file created in this case will contain a full, stand-alone HTML page. It is intended to be circulated to the co-hosts recording the episode to make it easier to access various information sources during the recording.

In the file the comments relating to past shows will show the full text, and there will be indications of comments that were read in the last recording, and any that were missed.

In order to highlight comments read, and those missed in the previous recording the script needs to know the date and time of the recording. This information should be in a date cache file referenced in the configuration file (usually **recording_dates.dat**). This file is updated when the monthly mail message is generated (see **make_email**). If, for any reason, this has not happened, the information can be provided with the **-lastrecording=DATETIME** option (alternatively written as **-lr=DATETIME**). See below for more information.

JSON details (**-json=FILE**)

This file will contain JSON data which is intended to be used to upload the episode to the database. How this is done is outside the scope of this document. The format used is very close to that used in the workflow which is used to upload episodes submitted through the upload forms.

In all cases the output file name may contain the characters '**%s**'. This denotes the point at which the year and month in the format **YYYY-MM** are inserted. For example if the script is being run for July 2014 the option:

```
-html=shownotes_%s.html
```

This will cause the generation of the file:

```
shownotes_2014-07.html
```

-lastrecording=DATETIME or **-lr=DATETIME**

As mentioned for **-full-html=FILE**, and later in the *MARKING COMMENTS* section, the script needs the date of the last recording when marking comments. This can be extracted from the file referenced in the configuration data using the setting **cache**. By default the name of this file is **recording_dates.dat**, and its contents are managed when the script **make_email** is run and by this script.

If for any reason the date and time of the last recording is missing, these values can be defined with this option, and these values will be written to the cache file (or modified, if necessary).

The format can be an ISO 8601 date followed by a 24-hour time, such as '2020-01-25 15:00:00'. If the time is omitted it defaults to the value of *starttime* in the configuration file.

The script will update the cache file with the date and time used in this option if the relevant entry is missing. Also, if an entry is present but the values are different from those provided with the option, the relevant entry will be updated.

Note that the **DATETIME** value must contain the date of the last recording. This will be checked, and written to the cache file prefixed by a "key" consisting of the first day of the month *BEFORE* the month being reviewed.

For example, when generating the notes for August 2025 the following command will be needed if there is no last recording date (for July 2025) in the cache:

```
    ./make_shownotes -from=2025-08-01
-full-html=full_shownotes_%s.html \
        -mail -comments -lr="2025-08-01 15:00:00"
```

Here we need the last recording date for the show reviewing HPR shows in July 2025. The date and time for this recording was in early August (Friday before the first Monday of September 2025-09-01), as shown. This combination will result in the addition of the following line to the cache file:

```
    2025-07-01,2025-08-01 15:00:00
```

As mentioned, the addition of such date and time information to the cache will normally be performed by **make_email**, which performs the date computations itself, unlike this script. This feature in this script is an alternative for special cases.

-[no]silent

This option controls whether the script reports details of its progress to STDERR. If the option is omitted the report is generated (**-nosilent**).

The script reports: the month it is working on, the name of the requested output files and details of the process of generating these files.

-[no]mailnotes

If desired, the show notes may include a section linking to recent discussions on the HPR Mailman mailing list.

The current template (defined in the configuration file by the variable **main_template**, **shownote_template12.tpl**) simply contains a section like the following:

```
    [%- IF mailnotes == 1 -%]
    <h2>Mailing List discussions</h2>
    <p>
    Policy decisions surrounding HPR are taken by the community
as a whole.
    This discussion takes place on the <a href="[% mailinglist
%]"
    target="_blank">Mailing List</a> which is open to all HPR
listeners and
    contributors. The discussions are open and available on the
HPR server
    under <a href="[% mailbase %]">Mailman</a>.
    </p>
    <p>The threaded discussions this month can be found
here:</p>
    <a href="[% mailthreads %]" target="_blank">[% mailthreads
%]</a>
    [%- END %]
```

The *TT2* variables such as **mailinglist** and **mailthreads** are defined earlier in the template.

-debug=N

Enables debugging mode when N > 0 (zero is the default, no debugging output). The levels are:

Values are:

1 TBA

2 Reports the following (as well as the data for level 1):

Details of the last recording data (and time)

3 Reports the following (as well as the data for level 2):

The generation of comment indexes needed in the comment lists. These are computed after the query has been run.

4 See the **DESCRIPTION** section for an explanation of the data structures mentioned here.

Reports the following (as well as the data for level 3):

A dump of the '%past' hash which contains details of comments on past shows.

A dump of the '%current' hash which contains details of comments on this month's shows.

A dump of the '@missed_comments' array containing comments that arrived after the last recording.

A list of the duplicated episode numbers in '@missed_episodes'

Another dump of '%past' after it has been cleaned up. Also the count of comments to past shows and the comment count.

MARKING COMMENTS

Explaining the marking of comments in the full HTML file:

This is only relevant when generating the full stand-alone HTML ("handout") for circulation to the volunteers recording the Community News episode. In this output the comments sent in to past shows (those before the review month) include their full texts in order to make reading them easier.

Normally comments (to shows in the reviewed month) are read as the shows themselves are reviewed, so the full texts are not needed in this handout.

In addition to this, there is a possibility that certain comments relating to past shows were already discussed last month, because they were made before that show was recorded. We don't want to read them again during this show, so a means of marking them is needed.

As well as this, some comments may have been missed in the last month because the recording was made before the end of the review month. On some months of the year the recording is made during the month itself because the first Monday of the next month is in the first few days of the next month. For example, in March 2019 the date of recording is the 30th, and the show is released on April 1st. Between the recording and the release of the show there is time during which more comments could be submitted.

Such comments should be in the notes for March (and these can be regenerated to make sure this is so) but they will not have been read on the March recording. The **make_shownotes** script detects this problem and, if generating full notes will show a list of any eligible comments in a red highlighted box. This is so that the volunteers recording the show can ensure they read comments that have slipped through this loophole.

The script extracts the date of the last recording from the *cache* file (or it can be specified with the **-lastrecording=DATETIME** option, or its abbreviation **-lr=DATETIME**) and passes it to the template. The template can then compare this date with the dates of the relevant comments and take action to highlight those that don't want to be re-read or were missed last month. It is up to the template to do what is necessary to highlight them.

DESCRIPTION

Overview

This script generates notes for the next Hacker Public Radio *Community News* show. It does this by collecting various details of activity from the HPR database and passing them to a template. The default template is called **shownote_template.tpl** (defined in the configuration file) and this generates HTML, but any suitable textual format could be generated if required, by using a different template.

Data Gathering

Four types of information are collected by the script:

Host details

Details of new hosts who have released new shows in the selected month

Show details

Details of shows which have been released in the selected month

Comments

Comments which have been submitted to the HPR website in the selected month. These need to be related to shows in the current period or in the past. Comments made about shows which have not yet been released (but are visible on the website) are not included even though they are made in the current month.

Mailing list threads

A link to the current threads on the mailing list in the past month can be included. This is done by default but can be skipped if the **-nomailnotes** option is used.

Report Generation

The four components listed above are formatted in the following way by the default template.

New Hosts

These are formatted as a list of links to the **hostid** with the host's name.

Shows

These are formatted into an HTML table containing the show number, title and host name. The show title is a link to the show page on the HPR website. The host name is a link to the host page on the website.

Comments

These are formatted with <article> tags separated by horizontal lines. A <header> shows the author name and title and a <footer> displays a link to the show and the show's host and the show title is also included. Where relevant, the body of the article contains the comment text with line breaks.

Mailing list discussions

A link to the mail threads on the mailing list is included if the **-mailnotes** option is chosen or defaulted.

See the explanation of the **-mailnotes** option for more details.

Variable, Field and Hash names

If you wish to write your own template refer to the following lists for the names of items. Also refer to the default template **shownote_template.tpl** for the techniques used there. (Note that **shownote_template.tpl** is a soft link to the current default template, such as **shownote_template12.tpl**, the soft link name is currently used in the configuration file).

The hash and field names available to the template are as follows

Global variables

```
Variable Name           Details
-------------           -------
review_month            The month name of the report date
review_year             The year of the report date
```

```
 comment_count          The number of comments in total
 past_count             The number of comments on old shows
 ignore_count           The number of past comments to ignore
 missed_count           The number of comments missed last time
 skip_comments          Set when -comments is omitted
 mark_comments          Set when comments are being marked
 ctext                  Set when the comment texts in the 'Past
shows'
                        section are to be shown
 last_recording         The date the last recording was made
                        in Unixtime format
 last_month             The month prior to the month for which the
notes are
                        being generated (computed if comments are
being
                        marked) in 'YYYY-MM' format
 mailnotes              Set when a mail link is required
```

New Hosts

The name of the hash in the template is **hosts**. The hash might be empty if there are no new hosts in the month. See the default template for how to handle this.

```
 Field Name     Details
 ----------     -------
 host           Name of host
 hostid         Host id number
```

Show Details

The name of the hash in the template is **shows**. Note that there are more fields available than are used in the default template. Note also that certain field names are aliases to avoid clashes (e.g. *eps_hostid* and *ho_hostid*).

```
 Field Name     Details
 ----------     -------
 eps_id         Episode number
 date           Episode date
 title          Episode title
 length         Episode duration
 summary        Episode summary
 notes          Episode show notes
 eps_hostid     The numerical host id from the 'eps' table
 series         The series number from the 'eps' table
 explicit       The explicit marker for the show
 eps_license    The license for the show
 tags           The show's tags as a comma-delimited string
 version        ?Obsolete?
 eps_valid      The valid value from the 'eps' table
 ho_hostid      The host id number from the 'hosts' table
 ho_host        The host name
 email          The hosts's email address (protected)
 profile        The host's profile
 ho_license     The default license for the host
 ho_valid       The valid value from the 'hosts' table
```

Comment Details

Two hashes are created for comments. The hash named **past** contains comments made in the review month to shows before this month, and **current** contains comments to this month's shows. Note that these hashes are only populated if the **-comments** option is provided or defaulted. Both hashes have the same structure.

```
 Field Name             Details
 ----------             -------
 episode                Episode number
 identifier_url         Full show URL
 title                  Episode title
 date                   Episode date
 host                   Host name
 hostid                 Host id number
 timestamp              Comment timestamp in ISO8601 format
 comment_author_name    Name of the commenter
 comment_title          Title of comment
 comment_text           Text of the comment
 comment_timestamp_ut   Comment timestamp in Unixtime format
 in_range               Boolean (0/1) denoting whether the comment
was made
                        in the target month
 index                  The numerical index of the comment for a
given show
 ignore                 Boolean (0/1), set if the comment is to be
ignored
```

The purpose of the **in_range** value is to denote whether a comment was made in the target month. This is used in the script to split the comments into the **past** and **current** hashes. It is therefore of little use in the template, but is retained in case it might be useful. The **index** value can be used in the template to refer to the comment, make linking URLs etc. It is generated by the SQL.

Mailing List Link

The variable **mailnotes** contains 0 or 1 depending on whether the link and accompanying text are required.

Filters

A filter called **decode_entities** is available to the template. The reason for creating this was when the HTML of a comment is being listed as text (Unicode actually). Since comment text is stored in the database as HTML with entities when appropriate this is needed to prevent the plain text showing *&amp;* and the like verbatim. It is not currently used, but has been left in case it might be of use in future.

DIAGNOSTICS

Unable to find configuration file ...

The nominated configuration file in **-config=FILE** (or the default file) cannot be found.

Error: Unable to find AOB file ...

The AOB file referred to in the error message is missing. Th

Use -lastrecording=DATETIME only with -full-html=FILE

The **-lastrecording=DATETIME** option is only relevant when the full stand-alone HTML is being generated. This is a fatal error.

At least one of -html=FILE, -full-html=FILE and -json=FILE must be present

The script writes up to three output files, as explained above. At least one must be present otherwise there is no point in running it!

Error: Unable to find ... file ...

The type of mandatory file with the name referred to in the error message is missing.

The date and time of the last recording is not in the cache

Followed by:

```
Use option -lastrecording=DATETIME (or -lr=DATETIME) instead
Can't continue
```

This means that the date and time expected in the date cache cannot be found, so the script needs to be run again with the information presented in the option mentioned.

Unable to find database

The SQLite database referenced in the configuration file has not been found.

Trying to overwrite an existing show. Aborting

After a check on the database, the computed episode number matches a slot that has already been allocated. A check has been made for an old-style placeholder for Community News episodes, but no match was found, so the script has aborted in case an existing episode will be overwritten.

Error: show ... has a date in the past

The date computed for the Community News episode is in the past. Perhaps the wrong date or month was specified in an option?

Problem with constructing JSON

A request for JSON output has been made. The script is attempting to collect information about the host who is preparing the show (*HPR Volunteers*) but the database query has failed.

It is possible the configuration file entry *hostid* contains the wrong host id number.

Unable to find a reference show

The script is attempting to find the release date (and episode number) for the Community News show in the database. It does this by finding a reference episode and stepping forward, incrementing the episode number and date. The reference date is the earliest in the target month, but for some unexpected reason this has not been found.

Invalid DATE or DATETIME '...' in parse_to_dc

It is likely that the date provided through the **-from=DATE** option is invalid. Use an ISO8601 date in the format *YYYY-MM-DD*.

... failed to open '...': ...

There was a problem opening the date cache file. Check the details in the configuration file. This file is expected to be located in the same directory as the script.

... failed to close '...': ...

There was a problem closing the date cache file.

Invalid Date::Calc date and time (...) in dc_to_dt

There was a problem processing a date (and time). A likely cause is an invalid date in one of options which requires and date or date and time. The script will report more than usual on this error to try and aid with debugging.

CONFIGURATION AND ENVIRONMENT

The script obtains the details and settings it required from a configuration file. The name of the file it expects is **.make_shownotes.cfg**, but this can be changed with the **-config=FILE** option.

The configuration file is expected to be in the directory holding the script. The script determines its location dynamically for this purpose.

The configuration file contains the following settings, which are explained at the end.

```
 #
 # .make_shownotes.cfg (2025-04-13)
 # Configuration file for make_shownotes version >= 4
 #
 <settings>
     # Format strings (using 'printf' formatting) for building certain
required strings
     title_template   = HPR Community News for %s %s
     summary_template = HPR Volunteers talk about shows released and
comments posted in %s %s

     # Repeat the following line with each of the desired tags to make an
     # array-like structure
     tags             = Community News

     # Host id for HPR Volunteers
     # Series id for HPR Community News series
     hostid = 159
     series_id = 47

     # Day the Community News show is released
     releaseday = Monday

     # Default day of the week for the recording
     recordingday = Friday

     # Recording times are UTC
     starttime = 16:00:00
     endtime = 17:00:00

     # cache of previous recording dates and times
     cache = recording_dates.dat

     # Template Toolkit templates
     # --------------------------

     # Main note template (actually a soft link)
     main_template = shownote_template.tpl

     # Used to make a stand-alone HTML file from the default HTML
     # fragment
     container_template = shownotes_container.tpl

 </settings>

 <database>
     # Assume a local file
     name = hpr.db
 </database>
```

Configuration file details

If any mandatory elements are omitted from the configuration file they are given default values in the script.

Section "settings"

title_template, **summary_template**

These are (*sprintf*) templates for these database fields, which are added to the full HTML and the JSON. They are filled in (using *sprintf*) in the script.

tags

Up to now, Community News episodes have used the tag **"Community News"**, so this is now used in the configuration file.

hostid, **series_id**

As documented in the file, these are the internal numbers from the database for the host and series associated with the show.

releaseday, **recordingday**, **starttime**, **endtime**

These values define default dates and times for the recording of the show and the date of release.

cache

This is the name of the file holding recording dates and times for Community News shows. The file is updated when **make_email** is run since this script defines the recording date and time. The script expects this file to be in the directory holding the script itself, though an absolute path could be used if needed.

main_template, **container_template**

These are *TT2* templates to be used by the script. The first is the template used to generate the two types of HTML, and the second is used to generate stand-alone HTML when the **-full-html=FILE** option is used. The script uses both templates when generating the full HTML.

The **container_template** contains a snapshot of an HPR HTML page with the body removed and the header and footer retained. The generated notes are added by the script, giving the HTML which can be handed out to the volunteers hosting the recording.

Section "database"

name

This contains the name of the SQLite database. This is a file which is expected to be in the same directory as the script.

DEPENDENCIES

Modules used:

```
Carp
Config::General
Cwd
DBI
Data::Dumper
Date::Calc
Date::Parse
DateTime::Duration
DateTime
File::Copy
Getopt::Long
```

```
HTML::Entities
JSON
Pod::Usage
Template::Filters
Template
```

BUGS AND LIMITATIONS

There are no known bugs in this module. Please report problems to Dave Morriss (Dave.Morriss@gmail.com) Patches are welcome.

AUTHOR

Dave Morriss (Dave.Morriss@gmail.com)

LICENCE AND COPYRIGHT