

# Hedgehog Tools Guide 2.1.0rc1

The Hedgehog tools are a set of scripts that are used to perform various tasks in terms of creating and maintaining the database and hedgehog files on disk. They can be run directly from `/libexec/hedgehog` or for convenience a wrapper script (`/bin/hedgehogctl`) is provided:

```
>hedgehogctl help
```

Run a Hedgehog command

```
Usage: hedgehogctl [help|list] [COMMAND] [command_options]
```

The available `hedgehogctl` commands are described below

- **Data Manager**
  - **Database**
    - `database_init`
    - `database_manage_partitions`
    - `database_process_rssac_data`
    - `database_rm_old_rssac_data`
    - `database_update_geoip`
    - `database_update_nodes`
    - `database_update_tlds_from_zone`
  - **Datafiles**
    - `datafiles_create_summary`
    - `datafiles_rm_empty_xml_dirs`
    - `datafiles_tar_old_xml_files`
- **Web front-end**
  - **Plotcache**
    - `plotcache_generate_cached_plots`
    - `plotcache_rm_cached_plots`
  - **RSSAC**
    - `rssac_generate_reports`
- **Other tools**
  - `conf_read`

## Data Manager

These scripts should be run as the `DB_OWNER` user (*hedgehog* by default).

### Database

#### `database_init`

- The script should be run after the database has been created (see the `<prefix>/libexec/database_create` helper script)
- This script populates the database with the plot and server/node data

Create and populate the database tables with plot options and nodes.

This calls the `hedeghog_database_update_nodes.sh` script which will populate the DB with the specified nodes and create a directory structure for them in the working data directory.

Usage: `database_init options`

Supported options:

- m Start month of oldest data to be imported  
(default is current month, format YYYY-MM)
- h Show this help

## database\_manage\_partitions

- This script creates new tables in the database to hold the imported data.

**It must be run at least once a month to create the tables for next month or the import will fail. It is recommended to configure a cron job as described in the Cron jobs section of the Installation guide.**

- Otherwise, the user will only have to run this script when new servers are added to an existing system.

Create a new set of partitioned data tables in the database to hold imported data.

The partitions are created per month of data.

Usage: `database_manage_partitions options`

Supported options:

- m Month to create partitions for  
(default is next month, format YYYY-MM)
- h Show this help

## database\_process\_rssac\_data

- This script should be run to process the data ready for the `rssac_process_data` script.
- The importer inserts data from the XML files into the 'unique\_source\_raw' table. For each node, this script creates entries in the 'unique\_source\_summary' table based on the raw data and then (optionally) deletes the original raw data from the 'unique\_source\_raw' table.

```
Process RSSAC data for a specified 24 hour period
```

```
Usage: database_process_rssac_data options
```

```
Supported options:
```

```
-d Date on which to process RSSAC data  
  (default is 7 days ago, format: YYYY-MM-DD)  
-f Force overwrite of existing summary data  
-D Also delete the raw unique_source data from the database to save  
space  
  (default: no delete)  
-h Show this help
```

### database\_rm\_old\_rssac\_data

- For some systems, the unique source raw data can be unmanageably large. This script can be used to truncate the database tables to recover the disk space, for example it can be run once a month to remove all the raw data for the previous month

```
Delete all the raw unique_source data in the database for a given  
month.  
All data older than this is also deleted.  
This script is intended to be run monthly to free disk space on systems  
where the raw unique sources data becomes unmanagably large.
```

```
Usage: database_rm_old_rssac_data options
```

```
Supported options:
```

```
-m Month from which to delete all raw unique source data  
  (default 2014-07: format YYYY-MM)  
  (default is 2 months ago when in the first half of the current  
month and  
  1 month ago when in the second half of the current month)  
-h Show this help
```

### database\_update\_geoup

- This script downloads a reference database of locations for use by the GEO maps. It should be periodically run to update with new location information.

```
Update the GeopIP list in the database. This reads input from /tmp  
Usage: database_update_geoup options  
Supported options:  
-h Show this help.
```

### database\_update\_nodes

- Run this script if new nodes or servers are added to the system

Add/update the servers and nodes in the database. This reads input from file

called nodes.csv in the /usr/local/etc/hedgehog directory.

- An example nodes.csv file is installed if one does not exist.
- See the comments in that file for details of the format.
- No action is taken on servers/nodes that are in the database but are not in the input files.

Usage: /usr/local/bin/hedgehog\_database\_update\_nodes options

Supported options:

- d debug mode (set -x).
- h Show this help.

### database\_update\_tlds\_from\_zone

- Run this script to update the categorisation of TLDs in the database from the root zone held in the IANA database.

Update the TLD list in the database. This reads input from IANA ftp site from the directory.

No action is taken on TLDs that are in the database but are not in the input data.

Usage: database\_update\_tlds\_from\_zone options

Supported options:

- h Show this help.

## Datfiles

### datafiles\_create\_summary

- This is a useful utility script that can be used to get a high level picture of the processing of the XML files for all the nodes in the system. It can help identify backlogs in the processing system.

Generate a processing report based on the information in the log files

Usage: datafiles\_create\_summary options

Supported options:

- s Only report on the server with this name (default is all servers)
- n Only report on the node with this name (default is all nodes)
- c Output in csv format for import into spreadsheet application
- d Run in debug mode (set -x)
- h Show this help.

### datafiles\_rm\_empty\_xml\_dirs

Remove empty incoming xml directories after a certain period of time (default is all the processed directories older than 7 days ago)

Usage: `datafiles_rm_empty_xml_dirs options`

Supported options:

- d Date before which removing empty xml directories (default: 7 days ago, format YYYY-MM-DD)
- h Show this help

## **datafiles\_tar\_old\_xml\_files**

Pack old xml files already processed into a .tar package to store and archive.

The xml files are packed according to their date and node and server. The filename has the following format `done-PID-date.tar.bz2` (e.g. `done-14143-2015-05-01.tar.bz2`).

Usage: `datafiles_tar_old_xml_files options`

Supported options:

- h Show this help.

## **Web front-end**

These scripts should be run as the `DB_READ_USER` user (*www-data* by default).

## **Plotcache**

### **plotcache\_generate\_cached\_plots**

- You must add at least one option for which plot window to generate [-D]-W[-M].
- By default, the cached plots are generated daily for the day before between 00:00 and 23:59. It can be helpful to configure a daily cron job to use this script to create cached plots to make loading of the homepage and common plots faster.

If not relying on the default behaviour be careful to specify BOTH a start and end date for this script, otherwise the end date will default to today and this may result in many plots being generated.

Create cached plots for each chosen period (day/week/month) between Start and End range of time.

Usage: generate\_cached\_plots options

Supported options:

- s Start date of time range from which to create cached plots (default is 1 day before at 00:00, format YYYY-MM-DDThh:mm)
- e End date of time range from which to create cached plots (default is 1 day before at 23:59, format YYYY-MM-DDThh:mm)
- D Generate daily cached plots
- W Generate weekly cached plots
- M Generate monthly cached plots
- h Show this help

## plotcache\_rm\_cached\_plots

- The user should not normally need to run this script, but is available in case problem are encountered with cached plots, or it is desirable to flush the cache for some reason (for example the default interactive plot type is changed after graphs have been cached).
- This script is interactive if the -a option is not selected. You will be asked to select which cached plot types you'd like to remove.

Remove all or a set of cached plots from the directory structure.

Usage: plotcache\_rm\_cached\_plots options

Supported options:

- a Remove all cached plots
- h Show this help

## RSSAC

### rssac\_generate\_reports

- By default the script creates RSSAC reports for all configured servers for a date 7 days in the past
- It will require that the database\_process\_rssac\_data script will have already been run for the date in question.
- Note the unique\_source data for the report is taken from the summary table, not the raw table.

Create RSSAC reports for a specified 24 hour period

Usage: rssac\_generate\_reports options

Supported options:

- d Date on which to create RSSAC reports (default is 7 days ago, format: YYYY-MM-DD)
- h Show this help

## Other tools

### conf\_read

- The script parses the contents of the <prefix>/etc/hedgehog/hedgehog.yaml file into environment variables for use by the other

scripts. Users will not need to run this directly.

- The database parameters are to be set by the user (if default ones need to be modified) and the directory structure parameters are auto-generated and better left alone.