

Header And Logo



Peripheral Links

[Donate to FreeBSD](#)

Search

Search

Site Navigation

- [Home](#)
- [About](#)
 - [Introduction](#)
 - [Features](#)
 - [Advocacy](#)
 - [Marketing](#)
- [Get FreeBSD](#)
 - [Release Information](#)
 - [Release Engineering](#)
- [Documentation](#)
 - [FAQ](#)
 - [Handbook](#)
 - [Porter's Handbook](#)

- [Developer's Handbook](#)
 - [Manual Pages](#)
 - [Documentation Project Primer](#)
 - [All Books and Articles](#)
- [Community](#)
 - [Mailing Lists](#)
 - [Forums](#)
 - [User Groups](#)
 - [Events](#)
- [Developers](#)
 - [Project Ideas](#)
 - [GIT Repository](#)
- [Support](#)
 - [Vendors](#)
 - [Security Information](#)
 - [Bug Reports](#)
 - [Submit Bug-report](#)
- [Foundation](#)
 - [Donate](#)

FreeBSD Manual Pages

sort

man

apropos

All Sections ▾

FreeBSD 13.2-RELEASE and Ports ▾

All Architectures ▾

html ▾

a block-sorting file compressor, v1.0.8 bzip2recover - recovers data from damaged bzip2 files

qsort(3), qsort_b(3), qsort_r(3), heapsort(3), heapsort_b(3), mergesort(3), mergesort_b(3)

sort functions

radixsort(3), sradixsort(3)

radix sort

scandir(3), scandirat(3), scandir_b(3), alphasort(3), versionsort(3)

scan a directory

sort(1)

sort or merge records (lines) of text and binary files

tsort(1)

topological sort of a directed graph

DEFINE_STACK_OF(3), DEFINE_STACK_OF_CONST(3), DEFINE_SPECIAL_STACK_OF(3), DEFINE_SPECIAL_STACK_OF_CONST(3), sk_TYPE_num(3), sk_TYPE_value(3), sk_TYPE_new(3), sk_TYPE_new_null(3), sk_TYPE_reserve(3), sk_TYPE_free(3), sk_TYPE_zero(3), sk_TYPE_delete(3), sk_TYPE_delete_ptr(3), sk_TYPE_push(3), sk_TYPE_unshift(3), sk_TYPE_pop(3), sk_TYPE_shift(3), sk_TYPE_pop_free(3), sk_TYPE_insert(3), sk_TYPE_set(3), sk_TYPE_find(3), sk_TYPE_find_ex(3), sk_TYPE_sort(3), sk_TYPE_is_sorted(3), sk_TYPE_dup(3), sk_TYPE_deep_copy(3), sk_TYPE_set_cmp_func(3), sk_TYPE_new_reserve(3)

stack container

X509_CRL_get0_by_serial(3), X509_CRL_get0_by_cert(3), X509_CRL_get_REVOKED(3), X509_REVOKED_get0_serialNumber(3), X509_REVOKED_get0_revocationDate(3), X509_REVOKED_set_serialNumber(3), X509_REVOKED_set_revocationDate(3), X509_CRL_add0_revoked(3), X509_CRL_sort(3)

CRL revoked entry utility functions

BSeasy_A(3)

Given A in a standard numerical format, construct the sparse A that we need for BlockSolve. This routine is particularly useful for matrices created in Fortran. The rows on a processor must be contiguous in the global numbering. If they are not, then other BlockSolve routines must be called to construct the matrix for BlockSolve. Arrays are indexed starting with 0. The values in a row must be sorted according to column number

Beep(3), Beep CDKallocStrings CDKcountStrings CDKfreeChtypes CDKfreeStrings CDKsplitString baseName ceilCDK chlen chstrncpy chtype2Char cleanChar cleanChtype cmpStrChstr copyChar copyCharList copyChtype copyChtypeList dirName floorCDK freeChar freeCharList freeChtype freeChtypeList intlen lenCharList lenChtypeList mode2Char searchList sortList cdk_misc(3)

Curses Development Kit Miscellaneous Routines

Consortium(7), XConsortium(7)

X Consortium information

DEFINE_STACK_OF(3), DEFINE_STACK_OF_CONST(3), DEFINE_SPECIAL_STACK_OF(3), DEFINE_SPECIAL_STACK_OF_CONST(3), sk_TYPE_num(3), sk_TYPE_value(3), sk_TYPE_new(3), sk_TYPE_new_null(3), sk_TYPE_reserve(3), sk_TYPE_free(3), sk_TYPE_zero(3), sk_TYPE_delete(3), sk_TYPE_delete_ptr(3), sk_TYPE_push(3), sk_TYPE_unshift(3), sk_TYPE_pop(3), sk_TYPE_shift(3), sk_TYPE_pop_free(3), sk_TYPE_insert(3),

[sk_TYPE_set\(3\)](#), [sk_TYPE_find\(3\)](#), [sk_TYPE_find_ex\(3\)](#), [sk_TYPE_sort\(3\)](#), [sk_TYPE_is_sorted\(3\)](#), [sk_TYPE_dup\(3\)](#), [sk_TYPE_deep_copy\(3\)](#),
[sk_TYPE_set_cmp_func\(3\)](#), [sk_TYPE_new_reserve\(3\)](#)

stack container

[DEFINE_STACK_OF\(3ossl\)](#), [DEFINE_STACK_OF_CONST\(3ossl\)](#), [DEFINE_SPECIAL_STACK_OF\(3ossl\)](#), [DEFINE_SPECIAL_STACK_OF_CONST\(3ossl\)](#),
[sk_TYPE_num\(3ossl\)](#), [sk_TYPE_value\(3ossl\)](#), [sk_TYPE_new\(3ossl\)](#), [sk_TYPE_new_null\(3ossl\)](#), [sk_TYPE_reserve\(3ossl\)](#), [sk_TYPE_free\(3ossl\)](#),
[sk_TYPE_zero\(3ossl\)](#), [sk_TYPE_delete\(3ossl\)](#), [sk_TYPE_delete_ptr\(3ossl\)](#), [sk_TYPE_push\(3ossl\)](#), [sk_TYPE_unshift\(3ossl\)](#), [sk_TYPE_pop\(3ossl\)](#),
[sk_TYPE_shift\(3ossl\)](#), [sk_TYPE_pop_free\(3ossl\)](#), [sk_TYPE_insert\(3ossl\)](#), [sk_TYPE_set\(3ossl\)](#), [sk_TYPE_find\(3ossl\)](#), [sk_TYPE_find_ex\(3ossl\)](#),
[sk_TYPE_find_all\(3ossl\)](#), [sk_TYPE_sort\(3ossl\)](#), [sk_TYPE_is_sorted\(3ossl\)](#), [sk_TYPE_dup\(3ossl\)](#), [sk_TYPE_deep_copy\(3ossl\)](#), [sk_TYPE_set_cmp_func\(3ossl\)](#),
[sk_TYPE_new_reserve\(3ossl\)](#), [OPENSSL_sk_deep_copy\(3ossl\)](#), [OPENSSL_sk_delete\(3ossl\)](#), [OPENSSL_sk_delete_ptr\(3ossl\)](#), [OPENSSL_sk_dup\(3ossl\)](#),
[OPENSSL_sk_find\(3ossl\)](#), [OPENSSL_sk_find_ex\(3ossl\)](#), [OPENSSL_sk_find_all\(3ossl\)](#), [OPENSSL_sk_free\(3ossl\)](#), [OPENSSL_sk_insert\(3ossl\)](#),
[OPENSSL_sk_is_sorted\(3ossl\)](#), [OPENSSL_sk_new\(3ossl\)](#), [OPENSSL_sk_new_null\(3ossl\)](#), [OPENSSL_sk_new_reserve\(3ossl\)](#), [OPENSSL_sk_num\(3ossl\)](#),
[OPENSSL_sk_pop\(3ossl\)](#), [OPENSSL_sk_pop_free\(3ossl\)](#), [OPENSSL_sk_push\(3ossl\)](#), [OPENSSL_sk_reserve\(3ossl\)](#), [OPENSSL_sk_set\(3ossl\)](#),
[OPENSSL_sk_set_cmp_func\(3ossl\)](#), [OPENSSL_sk_shift\(3ossl\)](#), [OPENSSL_sk_sort\(3ossl\)](#), [OPENSSL_sk_unshift\(3ossl\)](#), [OPENSSL_sk_value\(3ossl\)](#),
[OPENSSL_sk_zero\(3ossl\)](#)

stack container

[FcFontSetSort\(3\)](#)

Add to a font set

[FcFontSetSortDestroy\(3\)](#)

DEPRECATED destroy a font set

[FcFontSort\(3\)](#)

Return list of matching fonts

[HPL_logsort\(3\)](#)

Sort the processes in logarithmic order

[OBJ_NAME_add\(3\)](#), [OBJ_NAME_remove\(3\)](#), [OBJ_NAME_get\(3\)](#), [OBJ_NAME_new_index\(3\)](#), [OBJ_NAME_init\(3\)](#), [OBJ_NAME_cleanup\(3\)](#),
[OBJ_NAME_do_all\(3\)](#), [OBJ_NAME_do_all_sorted\(3\)](#)

global associative array

[Smokeping_sorters_Loss\(3\)](#), [Smokeping::sorters::Loss\(3\)](#)

Order the target charts by loss

[Smokeping_sorters_Max\(3\)](#), [Smokeping::sorters::Max\(3\)](#)

Order the target charts by Max RTT

[Smokeping_sorters_Median\(3\)](#), [Smokeping::sorters::Median\(3\)](#)

Order the target charts by Median RTT

[Smokeping_sorters_StdDev\(3\)](#), [Smokeping::sorters::StdDev\(3\)](#)

Order the target charts by StdDev

[Smokeping_sorters_base\(3\)](#), [Smokeping::sorters::base\(3\)](#)

Base Class for implementing SmokePing Sorters

[Sort\(3o\)](#)

Sorting and merging lists

[X509_CRL_get0_by_serial\(3\)](#), [X509_CRL_get0_by_cert\(3\)](#), [X509_CRL_get_REVOKED\(3\)](#), [X509_REVOKED_get0_serialNumber\(3\)](#),
[X509_REVOKED_get0_revocationDate\(3\)](#), [X509_REVOKED_set_serialNumber\(3\)](#), [X509_REVOKED_set_revocationDate\(3\)](#), [X509_CRL_add0_revoked\(3\)](#),
[X509_CRL_sort\(3\)](#)

CRL revoked entry utility functions

[X509_CRL_get0_by_serial\(3ossl\)](#), [X509_CRL_get0_by_cert\(3ossl\)](#), [X509_CRL_get_REVOKED\(3ossl\)](#), [X509_REVOKED_get0_serialNumber\(3ossl\)](#),
[X509_REVOKED_get0_revocationDate\(3ossl\)](#), [X509_REVOKED_set_serialNumber\(3ossl\)](#), [X509_REVOKED_set_revocationDate\(3ossl\)](#),
[X509_CRL_add0_revoked\(3ossl\)](#), [X509_CRL_sort\(3ossl\)](#)

CRL revoked entry utility functions

[activateCDKAlphalist\(3\)](#), [activateCDKAlphalist](#) [destroyCDKAlphalist](#) [drawCDKAlphalist](#) [eraseCDKAlphalist](#) [getCDKAlphalistBox](#)
[getCDKAlphalistContents](#) [getCDKAlphalistCurrentItem](#) [getCDKAlphalistFillerChar](#) [getCDKAlphalistHighlight](#) [injectCDKAlphalist](#) [moveCDKAlphalist](#)
[newCDKAlphalist](#) [positionCDKAlphalist](#) [setCDKAlphalist](#) [setCDKAlphalistBackgroundAttrib](#) [setCDKAlphalistBackgroundColor](#) [setCDKAlphalistBox](#)
[setCDKAlphalistBoxAttribute](#) [setCDKAlphalistContents](#) [setCDKAlphalistCurrentItem](#) [setCDKAlphalistFillerChar](#) [setCDKAlphalistHighlight](#)
[setCDKAlphalistHorizontalChar](#) [setCDKAlphalistLLChar](#) [setCDKAlphalistLRChar](#) [setCDKAlphalistPostProcess](#) [setCDKAlphalistPreProcess](#)
[setCDKAlphalistULChar](#) [setCDKAlphalistURChar](#) [setCDKAlphalistVerticalChar](#) [cdk_alphalist\(3\)](#)

curses sorted list widget

[ares_set_sortlist\(3\)](#)

Initialize an ares_channel sortlist configuration

[astscript-sort-by-night\(1\)](#)

Sort input FITS files by night

[bogosort\(1\)](#)

sorts or doesn't sort files or standard input

[bzip2\(1\)](#), [bunzip2\(1\)](#)

a block-sorting file compressor, v1.0.8 [bzip2recover](#) - recovers data from damaged bzip2 files

[ceph-diff-sorted\(8\)](#)

compare two sorted files line by line

[cmp_set_offset\(3\)](#), [cmp_struct\(3\)](#), [cmp_char\(3\)](#), [cmp_short\(3\)](#), [cmp_int\(3\)](#), [cmp_long\(3\)](#), [cmp_float\(3\)](#), [cmp_double\(3\)](#), [cmp_long_double\(3\)](#),
[cmp_schar\(3\)](#), [cmp_uchar\(3\)](#), [cmp_ushort\(3\)](#), [cmp_uint\(3\)](#), [cmp_ulong\(3\)](#), [cmp_charptr\(3\)](#), [cmp_chararr\(3\)](#)

comparison functions for qsort, bsearch, and others

[cntlist\(5\)](#)

file listing number of times each tagged sense occurs in a semantic concordance, sorted most to least frequently tagged [cntlist.rev](#) - file listing

number of times each tagged sense occurs in a semantic concordance, sorted by sense key

[csort\(1\)](#)

sort lines of text files in the C locale

[ctwill\(1\)](#), [ctwill-refsort\(1\)](#), [ctwill-twinx\(1\)](#)

translate CWEB to TeX with mini-indexes

[datesort\(1\)](#)

Sort contents of FILE chronologically

[dnssort\(1\)](#)

sort DNS hostnames

[fpart\(1\)](#)

Sort and pack files into partitions

[ftimes-sortini\(1\)](#)

Sort dig output by hostname, name, and offset

[gcomm\(1\)](#), [comm\(1\)](#)

compare two sorted files line by line

[git-brv\(1\)](#)

List branches sorted by their last commit date

[gsort\(1\)](#), [sort\(1\)](#)

sort lines of text files

[gtsort\(1\)](#), [tsort\(1\)](#)

perform topological sort

[hdf_sort_obj\(3\)](#), [hdf_sort_obj\(3\)](#)

sort the children of an HDF node

[hstr\(1\)](#)

easily view, navigate, sort and use your command history with shell history suggest box

[ken\(1\)](#), [b1ff\(1\)](#), [censor\(1\)](#), [chef\(1\)](#), [cockney\(1\)](#), [eleet\(1\)](#), [fanboy\(1\)](#), [fudd\(1\)](#), [jethro\(1\)](#), [jibberish\(1\)](#), [jive\(1\)](#), [kenny\(1\)](#), [kraut\(1\)](#), [ky00te\(1\)](#), [nethackify\(1\)](#),

[newspeak\(1\)](#), [nyc\(1\)](#), [pirate\(1\)](#), [rasterman\(1\)](#), [scottish\(1\)](#), [spammer\(1\)](#), [scramble\(1\)](#), [studly\(1\)](#), [uniencode\(1\)](#), [upside-down\(1\)](#)

assorted text filters

[ldap_parse_sort_control\(3\)](#)

Decode the information returned from a search operation that used a server-side sort control

[ldap_sort_entries\(3\)](#), [ldap_sort_values\(3\)](#), [ldap_sort_strcasecmp\(3\)](#)

LDAP sorting routines (deprecated)

[ldns_rr_list_sort\(3\)](#)

sort a `ldns_rr_list`

[ldns_zone_sort\(3\)](#), [ldns_zone_glue_rr_list\(3\)](#)

sort a zone and get the glue records

[llvm-pdbutil\(1\)](#)

PDB File forensics and diagnostics 0.0 (bu 2 %Synopsis (bu 2 %Description (bu 2 %Subcommands 2.0 (bu 2 %pretty 2.0 (bu 2 %Summary (bu 2 %Options 2.0 (bu 2 %Filtering and Sorting Options (bu 2 %Symbol Type Options (bu 2 %Other Options (bu 2 %dump 2.0 (bu 2 %Summary (bu 2 %Options 2.0 (bu 2 %MSF Container Options (bu 2 %Module & File Options (bu 2 %Symbol Options (bu 2 %Type Record Options (bu 2 %Miscellaneous Options (bu 2 %bytes 2.0 (bu 2 %Summary (bu 2 %Options 2.0 (bu 2 %MSF File Options (bu 2 %PDB Stream Options (bu 2 %DBI Stream Options (bu 2 %Module Options (bu 2 %Type Record Options (bu 2 %pdb2yaml 2.0 (bu 2 %Summary (bu 2 %Options (bu 2 %yaml2pdb 2.0 (bu 2 %Summary (bu 2 %Options (bu 2 %merge 2.0 (bu 2 %Summary (bu 2 %Options

[lmdbg-sort\(1\)](#)

sort stacktraces

[lsort\(1\)](#)

sort transcripts

[lsort.tcl85\(n\)](#), [lsort\(n\)](#)

Sort the elements of a list

[lsort.tcl86\(n\)](#), [lsort\(n\)](#)

Sort the elements of a list

[lsort.tcl87\(n\)](#), [lsort\(n\)](#)

Sort the elements of a list

[magicsort\(1\)](#)

Categorize files by their file(1) magic

[mailinspect\(1\)](#)

sort an mbox by category and pipe emails to a command

[memisort\(3\)](#)

insert new element into sorted array

[mlr\(1\)](#), [miller\(1\)](#)

like awk, sed, cut, join, and sort for name-indexed data such as CSV and tabular JSON

[msort\(1\)](#)

sort messages

[mtbl\(7\)](#)

immutable sorted string library

[mtbl_sorter\(3\)](#)

sort a sequence of unordered key-value pairs

[mysqlsla\(3\)](#)

Parse, filter, analyze and sort MySQL slow, general and binary logs

[oggz-sort\(1\)](#)

Sort the pages of an Ogg file in order of presentation time

[parsort\(1\)](#)

Sort (big files) in parallel

[pkg_sort\(1\)](#)

sort lines of package names in dependency order

[ptsort\(1\)](#)

Prioritized topological sort

[rasort\(1\)](#)

sort argus(8) data file

[results\(3\)](#), [insertinit\(3\)](#), [insertsort\(3\)](#), [get_results\(3\)](#), [set_results\(3\)](#), [save_median\(3\)](#), [save_minimum\(3\)](#)

the Imbench results subsystem

[rwsort\(1\)](#)

Sort SiLK Flow records on one or more fields

[samtools-markdup\(1\)](#), [samtools markdup\(1\)](#)

mark duplicate alignments in a coordinate sorted file EX in +\ \$1 nf ft CR . EE ft fi in

[samtools-merge\(1\)](#), [samtools merge\(1\)](#)

merges multiple sorted files into a single file EX in +\ \$1 nf ft CR . EE ft fi in

[samtools-sort\(1\)](#), [samtools sort\(1\)](#)

sorts SAM/BAM/CRAM files EX in +\ \$1 nf ft CR . EE ft fi in

[scene_gap\(3\)](#)

Number controlling the scene z-sorting algorithm behaviour. Allegro game programming library

[sk_new_null\(3\)](#), [sk_new\(3\)](#), [sk_set_cmp_func\(3\)](#), [sk_dup\(3\)](#), [sk_free\(3\)](#), [sk_pop_free\(3\)](#), [sk_num\(3\)](#), [sk_value\(3\)](#), [sk_find\(3\)](#), [sk_find_ex\(3\)](#), [sk_sort\(3\)](#),

[sk_is_sorted\(3\)](#), [sk_push\(3\)](#), [sk_unshift\(3\)](#), [sk_insert\(3\)](#), [sk_set\(3\)](#), [sk_pop\(3\)](#), [sk_shift\(3\)](#), [sk_delete\(3\)](#), [sk_delete_ptr\(3\)](#), [sk_zero\(3\)](#)

variable-sized arrays of void pointers, called OpenSSL stacks

[slapo-sssvlv\(5\)](#)

Server Side Sorting and Virtual List View overlay to slapd

[slapo-valsor\(5\)](#)

Value Sorting overlay to slapd

[soldout_array\(3\)](#), [arr_adjust\(3\)](#), [arr_free\(3\)](#), [arr_grow\(3\)](#), [arr_init\(3\)](#), [arr_insert\(3\)](#), [arr_item\(3\)](#), [arr_newitem\(3\)](#), [arr_remove\(3\)](#), [arr_sorted_find\(3\)](#),

[arr_sorted_find_i\(3\)](#), [parr_adjust\(3\)](#), [parr_free\(3\)](#), [parr_grow\(3\)](#), [parr_init\(3\)](#), [parr_insert\(3\)](#), [parr_pop\(3\)](#), [parr_push\(3\)](#), [parr_remove\(3\)](#),

[parr_sorted_find\(3\)](#), [parr_sorted_find_i\(3\)](#), [parr_top\(3\)](#)

array handling functions for soldout

[sort-pictures\(1\)](#)

sort pictures according to exif date

[sortbib\(1\)](#)

sort bibliographic database

[sortbyquote\(1\)](#)

sort C/C++ code by quoted regions

[sorter\(1\)](#)

Sort files in an image into categories based on file type

[sortlocon\(3\)](#)

sort the logical connectors of a figure by name man1/alc_origin.1

[sortlosig\(3\)](#)

sort the logical signals of a figure by name man1/alc_origin.1

[sortm\(1\)](#)

sort nmh messages

[sortmail\(1\)](#)

sort email messages by date/time

[tabsort\(1\)](#)

Sort a FITS binary table based on values in one column

[texindex\(1\)](#)

sort Texinfo index files

[vfs_dirsort\(8\)](#)

Sort directory contents

[vsearch\(1\)](#)

a versatile open-source tool for microbiome analysis, including chimera detection, clustering, dereplication and rereplication, extraction, FASTA/FASTQ/SFF file processing, masking, orienting, pairwise alignment, restriction site cutting, searching, shuffling, sorting, subsampling, and taxonomic classification of amplicon sequences for metagenomics, genomics, and population genetics

[winop\(n\)](#)

Perform assorted window and image operations

[ws\(1\)](#)

sort the output of w by terminal name

[Algorithm::Networksort\(3\)](#)

Create Sorting Networks

[Algorithm::Networksort::Best\(3\)](#)

Optimized Sorting Networks

[Alien::Build::Plugin::Prefer::SortVersions\(3\)](#)

Plugin to sort candidates by most recent first

[Apache::MP3::Playlist\(3\)](#)

Manage directories of MP3 files with sorting and persistent playlists

[Apache::MP3::Sorted\(3\)](#)

Generate sorted streamable directories of MP3 files

Array::PrintCols(3)

Print or format array elements in vertically sorted columns

Bigtop::Backend::Init::Std(3)

Bigtop backend which works sort of like h2xs

Bio::Ontology::OBOEngine(3)

An Ontology Engine for OBO style flat file format from the Gene Ontology Consortium

CDB_File::Generator(3)

generate massive sorted CDB files simply

Catalyst::TraitFor::Controller::DBIC::DoesPaging(3)

Helps you paginate, search, sort, and more easily using DBIx::Class

Cz::Sort(3)

Czech sort

DBIx::Class::Schema::Loader::Optional::Dependencies(3), \$class(3)

Optional module dependency specifications (for module authors) EOC ### SYNOPSIS HEADING ### push @chunks, <<"EOC"; =head1 SYNOPSIS Somewhere in your build-file (e.g. ExtUtils::MakeMaker's Makefile.PL): ... e\$EUMM_ARGS{CONFIGURE_REQUIRES} = { e\${e\$EUMM_ARGS{CONFIGURE_REQUIRES} || {} }, DBIx::Class::Schema::Loader => \$distver, }; ... my %DBIC_CONFIG_AND_ORACLE_DEPS = %{ eval { require \$class; \$class->req_list_for([qw(dbicdump_config rdbms_oracle)]); } || {} }; e\$EUMM_ARGS{PREREQ_PM} = { e%DBIC_CONFIG_AND_ORACLE_DEPS, e\${e\$EUMM_ARGS{PREREQ_PM} || {} }, }; ... ExtUtils::MakeMaker::WriteMakefile(e\$EUMM_ARGS); Note: The eval protection within the example is due to support for requirements during the configure build phase not being available on a sufficient portion of production installations of Perl. Robust support for such dependency requirements is available in the CPAN installer only since version 1.94_56 first made available for production with perl version 5.12. It is the belief of the current maintainer that support for requirements during the configure build phase will not be sufficiently ubiquitous until the year 2020 at the earliest, hence the extra care demonstrated above. It should also be noted that some 3rd party installers (e.g. cpanminus) do the right thing with configure requirements independent from the versions of perl and CPAN available. EOC ### DESCRIPTION HEADING ### push @chunks, <<'EOC'; =head1 DESCRIPTION Some of the less-frequently used features of DBIx::Class::Schema::Loader have external module dependencies on their own. In order not to burden the average user with modules they will never use, these optional dependencies are not included in the base Makefile.PL. Instead an exception with a descriptive message is thrown when a specific feature can't find one or several modules required for its operation. This module is the central holding place for the current list of such dependencies, for DBIx::Class::Schema::Loader core authors, and DBIx::Class::Schema::Loader extension authors alike. Dependencies are organized in groups where each group can list one or more required modules, with an optional minimum version (or 0 for any version). In addition groups prefixed with test_ can specify a set of environment variables, some (or all) of which are marked as required for the group to be considered by req_list_for Each group name (or a combination thereof) can be used in the public methods as described below. EOC ### REQUIREMENT GROUPLIST HEADING ### push @chunks, '=head1 CURRENT REQUIREMENT GROUPS'; my \$standalone_info; for my \$group (sort keys %\$dbic_reqs) { my \$info = \$standalone_info->{\$group} || = \$class->_groups_to_reqs(\$group); next unless (\$info->{modreqs_fully_documented} and (\$info->{augments} or \$info->{modreqs})); my \$p = \$dbic_reqs->{\$group}{pod}; push @chunks, ("=head2 \$p->{title}", "=head3 \$group", \$p->{desc}, =over,); if (keys %{\$

```
$info->{modreqs} || {} ) { push @chunks, map { "=item * $_" . ($info->{modreqs}{$_} ? " >= $info->{modreqs}{$_} : ) } ( sort keys %{ $info->{modreqs} } ) ; } else { push @chunks, =item * No standalone requirements, } push @chunks, =back; for my $ag ( sort keys %{ $info->{augments} } || {} ) { my $ag_info = $standalone_info->{$ag} || = $class->_groups_to_reqs($ag); my $newreqs = $class->modreq_list_for([ $group, $ag ]); for (keys %$newreqs) { delete $newreqs->{$_} if ( ( defined $info->{modreqs}{$_} and $info->{modreqs}{$_} == $newreqs->{$_} ) or ( defined $ag_info->{modreqs}{$_} and $ag_info->{modreqs}{$_} == $newreqs->{$_} ) ); } if (keys %$newreqs) { push @chunks, ( "Combined with L</$ag> additionally requires:", =over, ( map { "=item * $_" . ($newreqs->{$_} ? " >= $newreqs->{$_}" : ) } ( sort keys %$newreqs ) ), =back, ); } } } #@@ #@@ API DOCUMENTATION HEADING #@@ push @chunks, <<'EOC';
```

[Data::Unixish::sort\(3\)](#)

Sort items

[Data::Unixish::subsort\(3\)](#)

Sort items using Sort::Sub routine

[Devel::Cover::Report::Sort\(3\)](#)

backend for Devel::Cover

[DtSearchSortResults\(3\)](#)

Sort DtSearch results lists

[File::Sort\(3\)](#)

Sort a file or merge sort multiple files

[Gtk2::TreeModelSort\(3\)](#)

wrapper for GtkTreeModelSort

[Gtk2::TreeSortable\(3\)](#)

wrapper for GtkTreeSortable

[Heap\(3\)](#)

Perl extensions for keeping data partially sorted

[Heap::Binary\(3\)](#)

a binary heap to keep data partially sorted

[Heap::Binomial\(3\)](#)

a binomial heap to keep data partially sorted

[Heap::Fibonacci\(3\)](#)

a fibonacci heap to keep data partially sorted

[JQuery::TableSorter\(3\)](#)

The JQuery TableSorter

[Jifty::Web::Form::Field\(3\)](#)

Web input of some sort

[List::BinarySearch\(3\)](#)

Binary Search within a sorted array

[MP3::Find\(3\)](#)

Search and sort MP3 files based on their ID3 tags

[MP3::Tag::LastResort\(3\)](#)

Module for using other fields to fill autotag fields

[MP3::Tag::Utils\(3\)](#)

Assorted utilities for manipulating MP3 files via MP3::Tag

[Math::GSL::Heapsort\(3\)](#)

Functions for sorting data, both directly and indirectly (using an index)

[Math::GSL::Sort\(3\)](#)

Functions for sorting data

[Math::IntervalSearch\(3\)](#)

Search where an element lies in a list of sorted elements

[MooX::ConfigFromFile::Role::SortedByFilename\(3\)](#)

allows filename based sort algorithm for multiple config files

[Net::Amazon::Request::Sort\(3\)](#)

listing of sort types

[Net::Amazon::Validate::ItemSearch::ca::Sort\(3\)](#)

valid search indices for the ca locale and the Sort operation

[Net::Amazon::Validate::ItemSearch::de::Sort\(3\)](#)

valid search indices for the de locale and the Sort operation

[Net::Amazon::Validate::ItemSearch::fr::Sort\(3\)](#)

valid search indices for the fr locale and the Sort operation

[Net::Amazon::Validate::ItemSearch::jp::Sort\(3\)](#)

valid search indices for the jp locale and the Sort operation

[Net::Amazon::Validate::ItemSearch::uk::Sort\(3\)](#)

valid search indices for the uk locale and the Sort operation

[Net::Amazon::Validate::ItemSearch::us::Sort\(3\)](#)

valid search indices for the us locale and the Sort operation

[Net::Connection::Sort\(3\)](#)

Sorts array of Net::Connection objects

[Net::Connection::Sort::host_f\(3\)](#)

Sorts the connections via the foreign host

[Net::Connection::Sort::host_fl\(3\)](#)

Sorts the connections via the foreign host and then local host

[Net::Connection::Sort::host_l\(3\)](#)

Sorts the connections via the local host

[Net::Connection::Sort::host_lf\(3\)](#)

Sorts the connections via the local host and then foreign host

[Net::Connection::Sort::pid\(3\)](#)

Sorts the connections via the PID

[Net::Connection::Sort::port_f\(3\)](#)

Sorts the connections via the foreign port numerically

[Net::Connection::Sort::port_fa\(3\)](#)

Sorts the connections via the foreign port alphabetically

[Net::Connection::Sort::port_l\(3\)](#)

Sorts the connections via the local port numerically

[Net::Connection::Sort::port_la\(3\)](#)

Sorts the connections via the local port alphabetically

[Net::Connection::Sort::proto\(3\)](#)

Sorts the connections via the connection protocol

[Net::Connection::Sort::ptr_f\(3\)](#)

Sorts via the foreign PTR

[Net::Connection::Sort::ptr_l\(3\)](#)

Sorts via the foreign PTR

[Net::Connection::Sort::state\(3\)](#)

Sorts the connections via the connection state

[Net::Connection::Sort::uid\(3\)](#)

Sorts the connections via the UID

[Net::Connection::Sort::user\(3\)](#)

Sorts the connections via the username

[Net::LDAP::Control::Sort\(3\)](#)

Server Side Sort (SSS) control object

[Net::LDAP::Control::SortResult\(3\)](#)

Server Side Sort (SSS) result control object

[PPIx::Regexp\(3\)](#)

Represent a regular expression of some sort

[Perl::Critic::Policy::BuiltinFunctions::ProhibitReverseSortBlock\(3\)](#)

Forbid \$b before \$a in sort blocks

[Perl::Critic::Policy::BuiltinFunctions::RequireSimpleSortBlock\(3\)](#)

Sort blocks should have a single statement

[Perl::Critic::Policy::Subroutines::ProhibitReturnSort\(3\)](#)

Behavior of "sort" is not defined if called in scalar context

[Pod::Abstract::Filter::sort\(3\)](#)

paf command to alphabetically sort sub-sections within a Pod section

[RDF::Query::Algebra::Sort\(3\)](#)

Algebra class for sorting

[RDF::Query::Plan::Sort\(3\)](#)

Executable query plan for Sorts

[Search::Xapian::DatabaseError\(3\)](#)

DatabaseError indicates some sort of database related error

[Search::Xapian::InternalError\(3\)](#)

InternalError indicates a runtime problem of some sort

[Search::Xapian::MultiValueSorter\(3\)](#)

allows sorting by a several values

[Senna::OptArg::Sort\(3\)](#)

Wrapper Around sen_sort_optarg

[Sort::ArbBiLex\(3\)](#)

make sort functions for arbitrary sort orders

[Sort::Array\(3\)](#)

This extended sorting algorithm allows you to a) sort an array by ANY field number, not only the first. b) find duplicates in your data-set and sort them out. The function is case-sensitive. Future versions might come without this limitation

[Sort::ArrayOfArrays\(3\)](#)

Perl extension for sorting an array of arrays

[Sort::Fields\(3\)](#)

Sort lines containing delimited fields

[Sort::Key\(3\)](#)

the fastest way to sort anything in Perl

[Sort::Key::DateTime\(3\)](#)

Perl extension for sorting objects by some DateTime key

[Sort::Key::IPv4\(3\)](#)

sort IP v4 addresses

[Sort::Key::Maker\(3\)](#)

multi-key sorter creator

[Sort::Key::Multi\(3\)](#)

simple multi-key sorts

[Sort::Key::Natural\(3\)](#)

fast natural sorting

[Sort::Key::Register\(3\)](#)

tell Sort::Key how to sort new data types

[Sort::Key::Top\(3\)](#)

select and sort top n elements

[Sort::Maker\(3\)](#)

A simple way to make efficient sort subs

[Sort::Naturally\(3\)](#)

-- sort lexically, but sort numeral parts numerically

[Sort::Sub\(3\)](#)

Collection of sort subroutines

[Sort::Sub::ascibetically\(3\)](#)

Sort ascibetically (string-wise)

[Sort::Sub::by_ascii_then_num\(3\)](#)

Sort non-numbers (sorted ascibetically) before numbers (sorted numerically)

[Sort::Sub::by_count\(3\)](#)

Sort by number of occurrences of pattern in string

[Sort::Sub::by_first_num_in_text\(3\)](#)

Sort by first number found in text or (if no number is found) ascibetically

[Sort::Sub::by_last_num_in_text\(3\)](#)

Sort by last number found in text or (if no number is found) ascibetically

[Sort::Sub::by_length\(3\)](#)

Sort by length of string

[Sort::Sub::by_num_in_text\(3\)](#)

Sort by first number found in text or (if no number is found) ascibetically

[Sort::Sub::by_num_then_ascii\(3\)](#)

Sort numbers (sorted numerically) before non-numbers (sorted ascibetically)

[Sort::Sub::by_perl_code\(3\)](#)

Sort by Perl code

[Sort::Sub::by_perl_function\(3\)](#)

Sort by Perl function

[Sort::Sub::by_perl_op\(3\)](#)

Sort by Perl operator

[Sort::Sub::by_rand\(3\)](#)

Sort randomly using Perl's rand()

[Sort::Sub::naturally\(3\)](#)

Sort naturally (by number or string parts)

[Sort::Sub::numerically\(3\)](#)

Sort numerically

[Sort::Sub::randomly\(3\)](#)

Sort randomly using Perl's rand()

[Sort::Tree\(3\)](#)

Organize list of objects into parent/child order

[Sort::Versions\(3\)](#)

a perl 5 module for sorting of revision-like numbers

[Sub::Sort\(3\)](#)

Dummy module to catch Sort::Sub typo

[Test::Sort::Sub\(3\)](#)

Test Sort::Sub::* subroutine

[Text::BibTeX::BibSort\(3\)](#)

generate sort keys for bibliographic entries

[Text::Pipe::List::Sort\(3\)](#)

Common text filter API

[Tie::Array::Sorted\(3\)](#)

An array which is kept sorted

[Tie::Array::Sorted::Lazy\(3\)](#)

An array which is kept sorted

[Tie::Hash::Sorted\(3\)](#)

Presents hashes in sorted order

[Tk::Columns\(3\)](#)

A multicolumn list widget with sortable & sizeable columns

[Tk::DockFrame\(3\)](#)

A multicolumn list widget with sortable & sizeable columns

[Tree::Simple::Visitor::Sort\(3\)](#)

A Visitor for sorting a Tree::Simple object hierarchy

[VCP::Filter::sort\(3\)](#)

Sort revs by field, order

[W3C::LogValidator::Basic\(3\)](#)

[W3C Log Validator] Sort Web server log entries by popularity (hits)

bk-sccslog(1), bk sccslog(1)

list deltas sorted by date across all files

bot_face_sort(nged)

Sorts the list of triangles that constitutes the BOT primitive to optimize it for raytracing with the specified number of triangles per piece

bzip(1), bunzip(1)

a block-sorting file compressor, v0.21

certbot(1)

certbot script documentation 0.0 3.5 C usage: certbot [SUBCOMMAND] [options] [-d DOMAIN] [-d DOMAIN] ... Certbot can obtain and install HTTPS/TLS/SSL certificates. By default, it will attempt to use a webserver both for obtaining and installing the certificate. The most common SUBCOMMANDS and flags are: obtain, install, and renew certificates: (default) run Obtain & install a certificate in your current webserver certonly Obtain or renew a certificate, but do not install it renew Renew all previously obtained certificates that are near expiry enhance Add security enhancements to your existing configuration -d DOMAINS Comma-separated list of domains to obtain a certificate for --apache Use the Apache plugin for authentication & installation --standalone Run a standalone webserver for authentication --nginx Use the Nginx plugin for authentication & installation --webroot Place files in a server(aqs webroot folder for authentication --manual Obtain certificates interactively, or using shell script hooks -n Run non-interactively --test-cert Obtain a test certificate from a staging server --dry-run Test "renew" or "certonly" without saving any certificates to disk manage certificates: certificates Display information about certificates you have from Certbot revoke Revoke a certificate (supply --cert-name or --cert-path) delete Delete a certificate (supply --cert-name) manage your account: register Create an ACME account unregister Deactivate an ACME account update_account Update an ACME account show_account Display account details --agree-tos Agree to the ACME server(aqs Subscriber Agreement -m EMAIL Email address for important account notifications options: -h, --help show this help message and exit -c CONFIG_FILE, --config CONFIG_FILE path to config file (default: /etc/letsencrypt/cli.ini and ~/.config/letsencrypt/cli.ini) -v, --verbose This flag can be used multiple times to incrementally increase the verbosity of output, e.g. -vvv. (default: 0) --max-log-backups MAX_LOG_BACKUPS Specifies the maximum number of backup logs that should be kept by Certbot(aqs built in log rotation. Setting this flag to 0 disables log rotation entirely, causing Certbot to always append to the same log file. (default: 1000) -n, --non-interactive, --noninteractive Run without ever asking for user input. This may require additional command line flags; the client will try to explain which ones are required if it finds one missing (default: False) --force-interactive Force Certbot to be interactive even if it detects it(aqs not being run in a terminal. This flag cannot be used with the renew subcommand. (default: False) -d DOMAIN, --domains DOMAIN, --domain DOMAIN Domain names to apply. For multiple domains you can use multiple -d flags or enter a comma separated list of domains as a parameter. The first domain provided will be the subject CN of the certificate, and all domains will be Subject Alternative Names on the certificate. The first domain will also be used in some software user interfaces and as the file paths for the certificate and related material unless otherwise specified or you already have a certificate with the same name. In the case of a name collision it will append a number like 0001 to the file path name. (default: Ask) --eab-kid EAB_KID Key Identifier for External Account Binding (default: None) --eab-hmac-key EAB_HMAC_KEY HMAC key for External Account Binding (default: None) --cert-name CERTNAME Certificate name to apply. This name is used by Certbot for housekeeping and in file paths; it doesn(aqt affect the content of the certificate itself. To see certificate names, run (aqcertbot certificates(aq. When creating a new certificate, specifies the new certificate(aqs name. (default: the first provided domain or the name of an existing certificate on your system for the same domains) --dry-run Perform a test run of the client,

obtaining test (invalid) certificates but not saving them to disk. This can currently only be used with the `(aqcertonly(aq and (aqrenew(aq subcommands. Note: Although --dry-run tries to avoid making any persistent changes on a system, it is not completely side-effect free: if used with webserver authenticator plugins like apache and nginx, it makes and then reverts temporary config changes in order to obtain test certificates, and reloads webserver to deploy and then roll back those changes. It also calls --pre-hook and --post-hook commands if they are defined because they may be necessary to accurately simulate renewal. --deploy- hook commands are not called. (default: False) --debug-challenges After setting up challenges, wait for user input before submitting to CA. When used in combination with the (ga-v(ga option, the challenge URLs or FQDNs and their expected return values are shown. (default: False) --preferred-chain PREFERRED_CHAIN Set the preferred certificate chain. If the CA offers multiple certificate chains, prefer the chain whose topmost certificate was issued from this Subject Common Name. If no match, the default offered chain will be used. (default: None) --preferred-challenges PREFERRED_CHALLENGES A sorted, comma delimited list of the preferred challenge to use during authorization with the most preferred challenge listed first (Eg, "dns" or "http,dns"). Not all plugins support all challenges. See https://certbot.eff.org/docs/using.html#plugins for details. ACME Challenges are versioned, but if you pick "http" rather than "http-01", Certbot will select the latest version automatically. (default: []) --issuance-timeout ISSUANCE_TIMEOUT This option specifies how long (in seconds) Certbot will wait for the server to issue a certificate. (default: 90) --user-agent USER_AGENT Set a custom user agent string for the client. User agent strings allow the CA to collect high level statistics about success rates by OS, plugin and use case, and to know when to deprecate support for past Python versions and flags. If you wish to hide this information from the Let's Encrypt server, set this to "". (default: Certbot/ACMEClient/2.0.0 (certbot; OS_NAME OS_VERSION) Authenticator/XXX Installer/YYY (SUBCOMMAND; flags: FLAGS) Py/major.minor.patchlevel). The flags encoded in the user agent are: --duplicate, --force-renew, --allow-subset-of-names, -n, and whether any hooks are set. --user-agent-comment USER_AGENT_COMMENT Add a comment to the default user agent string. May be used when repackaging Certbot or calling it from another tool to allow additional statistical data to be collected. Ignored if --user-agent is set. (Example: Foo-Wrapper/1.0) (default: None) automation: Flags for automating execution & other tweaks --keep-until-expiring, --keep, --reinstall If the requested certificate matches an existing certificate, always keep the existing one until it is due for renewal (for the (aqrun(aq subcommand this means reinstall the existing certificate). (default: Ask) --expand If an existing certificate is a strict subset of the requested names, always expand and replace it with the additional names. (default: Ask) --version show program(aq version number and exit --force-renewal, --renew-by-default If a certificate already exists for the requested domains, renew it now, regardless of whether it is near expiry. (Often --keep-until-expiring is more appropriate). Also implies --expand. (default: False) --renew-with-new-domains If a certificate already exists for the requested certificate name but does not match the requested domains, renew it now, regardless of whether it is near expiry. (default: False) --reuse-key When renewing, use the same private key as the existing certificate. (default: False) --no-reuse-key When renewing, do not use the same private key as the existing certificate. Not reusing private keys is the default behavior of Certbot. This option may be used to unset --reuse-key on an existing certificate. (default: False) --new-key When renewing or replacing a certificate, generate a new private key, even if --reuse-key is set on the existing certificate. Combining --new-key and --reuse-key will result in the private key being replaced and then reused in future renewals. (default: False) --allow-subset-of-names When performing domain validation, do not consider it a failure if authorizations can not be obtained for a strict subset of the requested domains. This may be useful for allowing renewals for multiple domains to succeed even if some domains no longer point at this system. This option cannot be used with --csr. (default: False) --agree-tos Agree to the ACME Subscriber Agreement (default: Ask) --duplicate Allow making a certificate lineage that duplicates an existing one (both can be renewed in parallel) (default: False) -q, --quiet Silence all output except errors. Useful for automation via cron. Implies --non-interactive. (default: False) security:`

Security parameters & server settings --rsa-key-size N Size of the RSA key. (default: 2048) --key-type {rsa,ecdsa} Type of generated private key. Only *ONE* per invocation can be provided at this time. (default: ecdsa) --elliptic-curve N The SECG elliptic curve name to use. Please see RFC 8446 for supported values. (default: secp256r1) --must-staple Adds the OCSP Must-Staple extension to the certificate. Autoconfigures OCSP Stapling for supported setups (Apache version >= 2.3.3). (default: False) --redirect Automatically redirect all HTTP traffic to HTTPS for the newly authenticated vhost. (default: redirect enabled for install and run, disabled for enhance) --no-redirect Do not automatically redirect all HTTP traffic to HTTPS for the newly authenticated vhost. (default: redirect enabled for install and run, disabled for enhance) --hsts Add the Strict-Transport-Security header to every HTTP response. Forcing browser to always use SSL for the domain. Defends against SSL Stripping. (default: None) --uir Add the "Content-Security-Policy: upgrade-insecure- requests" header to every HTTP response. Forcing the browser to use https:// for every http:// resource. (default: None) --staple-ocsp Enables OCSP Stapling. A valid OCSP response is stapled to the certificate that the server offers during TLS. (default: None) --strict-permissions Require that all configuration files are owned by the current user; only needed if your config is somewhere unsafe like /tmp/ (default: False) --auto-hsts Gradually increasing max-age value for HTTP Strict Transport Security security header (default: False) testing: The following flags are meant for testing and integration purposes only. --test-cert, --staging Use the staging server to obtain or revoke test (invalid) certificates; equivalent to --server https://acme-staging-v02.api.letsencrypt.org/directory (default: False) --debug Show tracebacks in case of errors (default: False) --no-verify-ssl Disable verification of the ACME server(aqs certificate. The root certificates trusted by Certbot can be overridden by setting the REQUESTS_CA_BUNDLE environment variable. (default: False) --http-01-port HTTP01_PORT Port used in the http-01 challenge. This only affects the port Certbot listens on. A conforming ACME server will still attempt to connect on port 80. (default: 80) --http-01-address HTTP01_ADDRESS The address the server listens to during http-01 challenge. (default:) --https-port HTTPS_PORT Port used to serve HTTPS. This affects which port Nginx will listen on after a LE certificate is installed. (default: 443) --break-my-certs Be willing to replace or renew valid certificates with invalid (testing/staging) certificates (default: False) paths: Flags for changing execution paths & servers --cert-path CERT_PATH Path to where certificate is saved (with certonly --csr), installed from, or revoked (default: None) --key-path KEY_PATH Path to private key for certificate installation or revocation (if account key is missing) (default: None) --fullchain-path FULLCHAIN_PATH Accompanying path to a full certificate chain (certificate plus chain). (default: None) --chain-path CHAIN_PATH Accompanying path to a certificate chain. (default: None) --config-dir CONFIG_DIR Configuration directory. (default: /etc/letsencrypt) --work-dir WORK_DIR Working directory. (default: /var/lib/letsencrypt) --logs-dir LOGS_DIR Logs directory. (default: /var/log /letsencrypt) --server SERVER ACME Directory Resource URI. (default: https://acme-v02.api.letsencrypt.org/directory) manage: Various subcommands and flags are available for managing your certificates: certificates List certificates managed by Certbot delete Clean up all files related to a certificate renew Renew all certificates (or one specified with --cert- name) revoke Revoke a certificate specified with --cert-path or --cert-name update_symlinks Recreate symlinks in your /etc/letsencrypt/live/ directory run: Options for obtaining & installing certificates certonly: Options for modifying how a certificate is obtained --csr CSR Path to a Certificate Signing Request (CSR) in DER or PEM format. Currently --csr only works with the (aqcertonly(aq subcommand. (default: None) renew: The (aqrenew(aq subcommand will attempt to renew any certificates previously obtained if they are close to expiry, and print a summary of the results. By default, (aqrenew(aq will reuse the plugins and options used to obtain or most recently renew each certificate. You can test whether future renewals will succeed with (ga--dry-run(ga. Individual certificates can be renewed with the (ga--cert-name(ga option. Hooks are available to run commands before and after renewal; see <https://certbot.eff.org/docs/using.html#renewal> for more information on these. --pre-hook PRE_HOOK Command to be run in a shell before obtaining any certificates. Intended primarily for renewal, where it can be used to temporarily shut down a webserver that might

conflict with the standalone plugin. This will only be called if a certificate is actually to be obtained/renewed. When renewing several certificates that have identical pre-hooks, only the first will be executed. (default: None) --post-hook POST_HOOK Command to be run in a shell after attempting to obtain/renew certificates. Can be used to deploy renewed certificates, or to restart any servers that were stopped by --pre-hook. This is only run if an attempt was made to obtain/renew a certificate. If multiple renewed certificates have identical post-hooks, only one will be run. (default: None) --deploy-hook DEPLOY_HOOK Command to be run in a shell once for each successfully issued certificate. For this command, the shell variable \$RENEWED_LINEAGE will point to the config live subdirectory (for example, "/etc/letsencrypt/live/example.com") containing the new certificates and keys; the shell variable \$RENEWED_DOMAINS will contain a space-delimited list of renewed certificate domains (for example, "example.com www.example.com") (default: None) --disable-hook-validation Ordinarily the commands specified for --pre-hook/--post-hook/--deploy-hook will be checked for validity, to see if the programs being run are in the \$PATH, so that mistakes can be caught early, even when the hooks aren't being run just yet. The validation is rather simplistic and fails if you use more advanced shell constructs, so you can use this switch to disable it. (default: False) --no-directory-hooks Disable running executables found in Certbot's hook directories during renewal. (default: False) --disable-renew-updates Disable automatic updates to your server configuration that would otherwise be done by the selected installer plugin, and triggered when the user executes "certbot renew", regardless of if the certificate is renewed. This setting does not apply to important TLS configuration updates. (default: False) --no-autorenew Disable auto renewal of certificates. (default: False) certificates: List certificates managed by Certbot delete: Options for deleting a certificate revoke: Options for revocation of certificates --reason {unspecified,keycompromise,affiliationchanged,superseded,cessationofoperation} Specify reason for revoking certificate. (default: unspecified) --delete-after-revoke Delete certificates after revoking them, along with all previous and later versions of those certificates. (default: None) --no-delete-after-revoke Do not delete certificates after revoking them. This option should be used with caution because the (aq)renew(aq) subcommand will attempt to renew undeleted revoked certificates. (default: None) register: Options for account registration --register-unsafely-without-email Specifying this flag enables registering an account with no email address. This is strongly discouraged, because you will be unable to receive notice about impending expiration or revocation of your certificates or problems with your Certbot installation that will lead to failure to renew. (default: False) -m EMAIL, --email EMAIL Email used for registration and recovery contact. Use comma to register multiple emails, ex: u1@example.com,u2@example.com. (default: Ask). --eff-email Share your e-mail address with EFF (default: None) --no-eff-email Don't share your e-mail address with EFF (default: None) update_account: Options for account modification unregister: Options for account deactivation. --account ACCOUNT_ID Account ID to use (default: None) install: Options for modifying how a certificate is deployed rollback: Options for rolling back server configuration changes --checkpoints N Revert configuration N number of checkpoints. (default: 1) plugins: Options for the "plugins" subcommand --init Initialize plugins. (default: False) --prepare Initialize and prepare plugins. (default: False) --authenticators Limit to authenticator plugins only. (default: None) --installers Limit to installer plugins only. (default: None) update_symlinks: Recreates certificate and key symlinks in /etc/letsencrypt/live, if you changed them by hand or edited a renewal configuration file enhance: Helps to harden the TLS configuration by adding security enhancements to already existing configuration. show_account: Options useful for the "show_account" subcommand: plugins: Plugin Selection: Certbot client supports an extensible plugins architecture. See (aq)certbot plugins(aq) for a list of all installed plugins and their names. You can force a particular plugin by setting options provided below. Running --help <plugin_name> will list flags specific to that plugin. --configurator CONFIGURATOR Name of the plugin that is both an authenticator and an installer. Should not be used together with --authenticator or --installer. (default: Ask) -a AUTHENTICATOR, --authenticator AUTHENTICATOR Authenticator plugin name. (default: None) -i INSTALLER, --installer INSTALLER Installer

plugin name (also used to find domains). (default: None) --apache Obtain and install certificates using Apache (default: False) --nginx Obtain and install certificates using Nginx (default: False) --standalone Obtain certificates using a "standalone" webserver. (default: False) --manual Provide laborious manual instructions for obtaining a certificate (default: False) --webroot Obtain certificates by placing files in a webroot directory. (default: False) --dns-cloudflare Obtain certificates using a DNS TXT record (if you are using Cloudflare for DNS). (default: False) --dns-digitalocean Obtain certificates using a DNS TXT record (if you are using DigitalOcean for DNS). (default: False) --dns-dnsimple Obtain certificates using a DNS TXT record (if you are using DNSimple for DNS). (default: False) --dns-dnsmadeeasy Obtain certificates using a DNS TXT record (if you are using DNS Made Easy for DNS). (default: False) --dns-gehirn Obtain certificates using a DNS TXT record (if you are using Gehirn Infrastructure Service for DNS). (default: False) --dns-google Obtain certificates using a DNS TXT record (if you are using Google Cloud DNS). (default: False) --dns-linode Obtain certificates using a DNS TXT record (if you are using Linode for DNS). (default: False) --dns-luadns Obtain certificates using a DNS TXT record (if you are using LuaDNS for DNS). (default: False) --dns-nsone Obtain certificates using a DNS TXT record (if you are using NS1 for DNS). (default: False) --dns-ovh Obtain certificates using a DNS TXT record (if you are using OVH for DNS). (default: False) --dns-rfc2136 Obtain certificates using a DNS TXT record (if you are using BIND for DNS). (default: False) --dns-route53 Obtain certificates using a DNS TXT record (if you are using Route53 for DNS). (default: False) --dns-sakuracloud Obtain certificates using a DNS TXT record (if you are using Sakura Cloud for DNS). (default: False) apache: Apache Web Server plugin (Please note that the default values of the Apache plugin options change depending on the operating system Certbot is run on.) --apache-enmod APACHE_ENMOD Path to the Apache (aqa2enmod(aq binary (default: None) --apache-dismod APACHE_DISMOD Path to the Apache (aqa2dismod(aq binary (default: None) --apache-le-vhost-ext APACHE_LE_VHOST_EXT SSL vhost configuration extension (default: -le- ssl.conf) --apache-server-root APACHE_SERVER_ROOT Apache server root directory (default: /etc/apache2) --apache-vhost-root APACHE_VHOST_ROOT Apache server VirtualHost configuration root (default: None) --apache-logs-root APACHE_LOGS_ROOT Apache server logs directory (default: /var/log /apache2) --apache-challenge-location APACHE_CHALLENGE_LOCATION Directory path for challenge configuration (default: /etc/apache2) --apache-handle-modules APACHE_HANDLE_MODULES Let installer handle enabling required modules for you (Only Ubuntu/Debian currently) (default: False) --apache-handle-sites APACHE_HANDLE_SITES Let installer handle enabling sites for you (Only Ubuntu/Debian currently) (default: False) --apache-ctl APACHE_CTL Full path to Apache control script (default: apache2ctl) --apache-bin APACHE_BIN Full path to apache2/httpd binary (default: None) dns-cloudflare: Obtain certificates using a DNS TXT record (if you are using Cloudflare for DNS). --dns-cloudflare-propagation-seconds DNS_CLOUDFLARE_PROPAGATION_SECONDS The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 10) --dns-cloudflare-credentials DNS_CLOUDFLARE_CREDENTIALS Cloudflare credentials INI file. (default: None) dns-digitalocean: Obtain certificates using a DNS TXT record (if you are using DigitalOcean for DNS). --dns-digitalocean-propagation-seconds DNS_DIGITALOCEAN_PROPAGATION_SECONDS The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 10) --dns-digitalocean-credentials DNS_DIGITALOCEAN_CREDENTIALS DigitalOcean credentials INI file. (default: None) dns-dnsimple: Obtain certificates using a DNS TXT record (if you are using DNSimple for DNS). --dns-dnsimple-propagation-seconds DNS_DNSIMPLE_PROPAGATION_SECONDS The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 30) --dns-dnsimple-credentials DNS_DNSIMPLE_CREDENTIALS DNSimple credentials INI file. (default: None) dns-dnsmadeeasy: Obtain certificates using a DNS TXT record (if you are using DNS Made Easy for DNS). --dns-dnsmadeeasy-propagation-seconds DNS_DNSMADEEASY_PROPAGATION_SECONDS The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 60) --dns-dnsmadeeasy-credentials DNS_DNSMADEEASY_CREDENTIALS DNS

Made Easy credentials INI file. (default: None) dns-gehirn: Obtain certificates using a DNS TXT record (if you are using Gehirn Infrastructure Service for DNS). --dns-gehirn-propagation-seconds DNS_GEHIRN_PROPAGATION_SECONDS The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 30) --dns-gehirn-credentials DNS_GEHIRN_CREDENTIALS Gehirn Infrastructure Service credentials file. (default: None) dns-google: Obtain certificates using a DNS TXT record (if you are using Google Cloud DNS for DNS). --dns-google-propagation-seconds DNS_GOOGLE_PROPAGATION_SECONDS The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 60) --dns-google-credentials DNS_GOOGLE_CREDENTIALS Path to Google Cloud DNS service account JSON file. (See <https://developers.google.com/identity/protocols/OAuth2ServiceAccount#creatinganaccount> for information about creating a service account and https://cloud.google.com/dns/access-control#permissions_and_roles for information about the required permissions.) (default: None) dns-linode: Obtain certificates using a DNS TXT record (if you are using Linode for DNS). --dns-linode-propagation-seconds DNS_LINODE_PROPAGATION_SECONDS The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 120) --dns-linode-credentials DNS_LINODE_CREDENTIALS Linode credentials INI file. (default: None) dns-luadns: Obtain certificates using a DNS TXT record (if you are using LuaDNS for DNS). --dns-luadns-propagation-seconds DNS_LUADNS_PROPAGATION_SECONDS The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 30) --dns-luadns-credentials DNS_LUADNS_CREDENTIALS LuaDNS credentials INI file. (default: None) dns-nsone: Obtain certificates using a DNS TXT record (if you are using NS1 for DNS). --dns-nsone-propagation-seconds DNS_NSONE_PROPAGATION_SECONDS The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 30) --dns-nsone-credentials DNS_NSONE_CREDENTIALS NS1 credentials file. (default: None) dns-ovh: Obtain certificates using a DNS TXT record (if you are using OVH for DNS). --dns-ovh-propagation-seconds DNS_OVH_PROPAGATION_SECONDS The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 120) --dns-ovh-credentials DNS_OVH_CREDENTIALS OVH credentials INI file. (default: None) dns-rfc2136: Obtain certificates using a DNS TXT record (if you are using BIND for DNS). --dns-rfc2136-propagation-seconds DNS_RFC2136_PROPAGATION_SECONDS The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 60) --dns-rfc2136-credentials DNS_RFC2136_CREDENTIALS RFC 2136 credentials INI file. (default: None) dns-route53: Obtain certificates using a DNS TXT record (if you are using AWS Route53 for DNS). --dns-route53-propagation-seconds DNS_ROUTE53_PROPAGATION_SECONDS The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 10) dns-sakuracloud: Obtain certificates using a DNS TXT record (if you are using Sakura Cloud for DNS). --dns-sakuracloud-propagation-seconds DNS_SAKURACLOUD_PROPAGATION_SECONDS The number of seconds to wait for DNS to propagate before asking the ACME server to verify the DNS record. (default: 90) --dns-sakuracloud-credentials DNS_SAKURACLOUD_CREDENTIALS Sakura Cloud credentials file. (default: None) manual: Authenticate through manual configuration or custom shell scripts. When using shell scripts, an authenticator script must be provided. The environment variables available to this script depend on the type of challenge. \$CERTBOT_DOMAIN will always contain the domain being authenticated. For HTTP-01 and DNS-01, \$CERTBOT_VALIDATION is the validation string, and \$CERTBOT_TOKEN is the filename of the resource requested when performing an HTTP-01 challenge. An additional cleanup script can also be provided and can use the additional variable \$CERTBOT_AUTH_OUTPUT which contains the stdout output from the auth script. For both authenticator and cleanup script, on HTTP-01 and DNS-01 challenges, \$CERTBOT_REMAINING_CHALLENGES will be equal to the number of challenges that remain after the current one, and \$CERTBOT_ALL_DOMAINS contains a comma-separated list of all domains that are challenged for the current certificate. --manual-auth-hook

MANUAL_AUTH_HOOK Path or command to execute for the authentication script (default: None) --manual-cleanup-hook
MANUAL_CLEANUP_HOOK Path or command to execute for the cleanup script (default: None) nginx: Nginx Web Server plugin --nginx-server-root NGINX_SERVER_ROOT Nginx server root directory. (default: /etc/nginx or /usr/local/etc/nginx) --nginx-ctl NGINX_CTL Path to the (aqnginx(aq binary, used for (aqconfigtest(aq and retrieving nginx version number. (default: nginx) --nginx-sleep-seconds
NGINX_SLEEP_SECONDS Number of seconds to wait for nginx configuration changes to apply when reloading. (default: 1) null: Null Installer standalone: Spin up a temporary webserver webroot: Place files in webroot directory --webroot-path WEBROOT_PATH, -w WEBROOT_PATH public_html / webroot path. This can be specified multiple times to handle different domains; each domain will have the webroot path that preceded it. For instance: (ga-w /var/www/example -d example.com -d www.example.com -w /var/www/thing -d thing.net -d m.thing.net(ga (default: Ask) --webroot-map WEBROOT_MAP JSON dictionary mapping domains to webroot paths; this implies -d for each entry. You may need to escape this from your shell. E.g.: --webroot-map (aq{"eg1.is,m.eg1.is":"/www/eg1/", "eg2.is":"/www/eg2"})(aq This option is merged with, but takes precedence over, -w / -d entries. At present, if you put webroot-map in a config file, it needs to be on a single line, like: webroot-map = {"example.com":"/var/www"}. (default: {}) P

comm(1)

select or reject lines common to two sorted files

cssort(1)

-- Czech sort

file_sorter(3)

File sorter

huge-merge.pl(1)

Merge the results of multiple huge-sort generated files into a single sorted file

huge-sort.pl(1)

Sort a --tokenlist of bigrams from huge-count.pl in alphabetical order

look(1)

find lines in a sorted list

read_mail_log(3), read_mail_log.pl(3)

-- Summarizes amount of mail routed through host, sorted by e-mail address

sort(1)

sort and/or merge files

sort(3)

perl pragma to control sort() behaviour

tv_sort(1)

Sort XMLTV listings files by date, and add stop times

ucsort(1)

sort input records using the UCA

