
Structured text tools

The following is a list of text-based file formats and command-line tools for manipulating each.

Contents

- awk-like
 - awk
 - POSIX commands
 - SQL-based tools
 - Other tools
- CSV
- HTML
- JSON
- TOML
- XML
- YAML
- Configuration files
 - /etc/hosts
 - INI
 - Multiple formats
- Log files
- Multiformat tools
- Templating for structured text
- Extra: interactive TUIs
- Extra: CLIs for single-file databases
- License
- Disclosure

awk-like

Tools that work with lines of fields separated by delimiters but do not necessarily support CSV field quoting.

awk

AWK/awk is a programming language and a POSIX-standard command-line tool. (You will sometimes see “awk” used for the tool and “AWK” for the language. This document follows this convention. GNU Awk uses “Awk”.) If you run Linux, macOS, or a BSD, you almost certainly have it installed. See below for Windows.

- If you already know how to program, the `nawk` man page is a great way to learn AWK quickly. What you learn from it will apply to other implementations on different platforms. Read it first if you feel overwhelmed by the sheer size of the GNU Awk manual.
- [awk.info](#) archive — an extensive resource on Awk.
- “AWK Vs NAWK Vs GAWK” — a comparison of features present in different implementations.
- `busybox-w32` includes a full implementation of POSIX awk and other tools like [sed](#) in a single Windows executable.
- `frawk` is a Rust implementation of a language partially compatible with AWK that supports parallelism and CSV input and output.
- GNU Awk 5 binaries for Windows by EZWinPorts.
- GoAWK is a cross-platform implementation of awk with added support for CSV. The project provides binaries for many platforms, including Windows.

POSIX commands

- [comm](#) — Select the lines common to two sorted files or the lines contained in only one of them. (Manual: `man 1 comm` on your system, GNU, FreeBSD.)
- [cut](#) — Select portions of each line in one or more files. (Manual: `man 1 cut`, GNU, FreeBSD.)
- [grep](#) — Select the lines that match or do not match a pattern from one or more files. (Manual: `man 1 grep`, GNU, FreeBSD.)
- [join](#) — Take two files sorted by a common field and join their lines on the value of that field. Lines with values that do not appear in the other file are discarded. (Manual: `man 1 join`, GNU, FreeBSD.)
- [paste](#) — Combine several consecutive lines in a text file into one. (Manual: `man 1 paste`, GNU, FreeBSD.)
- [sort](#) — Sort lines by key fields. (Manual: `man 1 sort`, GNU, FreeBSD.)
- [uniq](#) — Find or remove repeated lines. (Manual: `man 1 uniq`, GNU, FreeBSD.)

Other tools

- `csvquote` — Transform CSV to and from a format processable with awk-like tools.

-
- GNU datamash — Perform statistical operations on text input.
 - Hawk — Transform text from the command-line using Haskell expressions.
 - pyp — Transform input (as text lines or as a whole) using Python code with automatic module imports. Can generate a Python script equivalent to its invocation. In Python 3.11 or later supports TOML through tomllib.
 - rq — Convert between Apache Avro, CBOR, CSV, JSON, MessagePack, Protocol Buffers, TOML, YAML, and awk-style plain text.
 - vnlog — Process labelled tabular ASCII data using normal UNIX tools. Can plot data with gnuplot.

CSV

CSV, TSV, and other delimiter-separated value formats. Tools belong on this list if they support field quoting.

- csv-nix-tools — List *nix system information such as environment variables, files, processes, network connections, users as CSV. Manipulate and pretty-print CSV. Execute CSV rows as commands.
- csv2html — Convert CSV to HTML tables.
- csv2md — Convert CSV to Markdown tables.
- csvfaker — Generate CSV files with fake data. Supports different types of fake data in different locales: names, cities, jobs, email addresses, and others.
- csvfix — A multitool. Compare, filter, normalize, split, and validate CSV files. Reorder, remove, split, and merge fields. Convert data between fixed-width, multi-line, XML, and DSV format. Generate SQL statements. (Unofficial mirror.)
- csvkit — csvkit is a suite of command-line tools for converting to and working with CSV: convert, clean, cut, grep, join, sort, stack, format, render, query, analyze, etc.
- csvquote — Transform CSV to and from a format processable with awk-like tools.
- csvtk — Search, sample, cut, join, transpose, and sort CSV/TSV files. Rename columns. Replace fields and generate new fields from existing fields. Plot data as vector or raster histograms and box, line, and scatter plots. Convert CSV to Markdown. Convert XLSX to CSV. Split XLSX sheets.
- CSVtoTable — Convert CSV to a searchable and sortable HTML table.
- dasel — Query and update data structures from the command line. Comparable to jq/yq but supports CSV, JSON, TOML, YAML, and XML. Static binaries available for releases.
- eBay's TSV utilities — Filtering, statistics, sampling, joins and other operations on TSV files. High performance, especially good for large datasets. Written in D.
- emuto — CLI tool similar to jq. Create and manipulate CSV, TSV, and JSON. Can be compiled to JavaScript.

-
- **fawk** — A Rust implementation of a language partially compatible with AWK that supports parallelism and CSV input and output. **fawk** is an awk-derived language with a CSV mode for input and for output.
 - **GoAWK** — A cross-platform implementation of awk with added support for CSV. The project provides binaries for many platforms, including Windows. **GoAWK** is an awk implementation that adds a CSV mode for input and for output.
 - **Graphstage** — Compare and merge tree-like structures semantically. Supports JSON, JSON5, XML, HTML, YAML, and CSV. Can be used as a Python library.
 - **jp (sgreben)** — Plot JSON and CSV data in the terminal. Supports different kinds of plots: bar charts, line charts, scatter plots, histograms, and heatmaps.
 - **Mario** — Manipulate and convert between CSV, JSON, YAML, TOML, and XML with Python code.
 - **MCMD (M-Command)** — Select, sample, cut, join, sort, reformat, and generate CSV files. Contains a large set of commands.
 - **Miller** — [sed](#), [awk](#), [cut](#), [join](#) and [sort](#) for name-indexed data such as CSV and tabular JSON.
 - **Nushell** — A command shell. Can natively load data from CSV, INI, JSON, TOML, TSV, XML, YAML, and other formats.
 - **pawk** — Process text with AWK-like patterns, but Python code.
 - **qsv** — Index, slice, analyze, split, and join CSV files. A fork of **xsv** that adds subcommands and features.
 - **ReadStat** — Convert statistics package datasets between SAS (SAS7BDAT, XPORT), SPSS (POR, SAV, ZSAV), and Stata (DTA). Convert those formats to CSV and XLSX. Can be used as a C library with bindings for Julia, Python, and R.
 - **rows** — A Python library with a CLI. Convert between a number of file formats for tabular data: CSV, XLS, XLSX, ODS, and others. Query the data (via SQLite). Combine tables. Generate schemas.
 - **rq** — Convert between Apache Avro, CBOR, CSV, JSON, MessagePack, Protocol Buffers, TOML, YAML, and awk-style plain text.
 - **scrubcsv** — Remove bad lines from a CSV file and normalize the rest. Written in Rust.
 - **Skeem** — Infer SQL DDL statements from tabular data. Supports CSV, JSON, JSON Lines, ODS, XLSX, and other formats.
 - **tab** — A non-Turing-complete statically typed programming language for data processing. An alternative to awk.
 - **teip** — Select fields, character ranges, or regular expression matches from standard input. Replace them with the output of a command.
 - **tv** — View delimited files in the terminal.
 - **xsv** — Index, slice, analyze, split, and join CSV files.
 - **zsv** — Slice, combine, reformat, flatten/unflatten CSV (TSV, DSV) files. Query them with SQL and **jq** filters. Convert between them, JSON, and SQLite 3. Also a C library.
-

SQL-based tools

See the big comparison list. It covers

- AlaSQL CLI
- csvq
- csvsql
- fsql
- Musoq
- q
- RBQL
- rows
- Sqawk (dbohdan)
- sqawk (tjunier)
- Squawk
- termsql
- trdsql
- textql

HTML

- Graphtag — Compare and merge tree-like structures semantically. Supports JSON, JSON5, XML, HTML, YAML, and CSV. Can be used as a Python library.
- hred — Query XML and HTML with a query language based on CSS selectors.
- html-xml-utils — A number of simple utilities (like [hxcopy](#), [hxpipe](#), [hxunent](#), [hxselect](#)) for manipulating HTML and XML files from W3C. Written in C, quite old-fashioned, but still relevant and maintained.
- htmlq — Query HTML with CSS selectors. Can remove elements in the output.
- pup — Query HTML pages with CSS selectors. Static binaries available for releases. Inspired by jq.
- Saxon — Query XML and HTML data with XPath. Documentation.
- Temme — Query HTML with CSS-like selectors to extract JSON. Temme extends CSS selectors with value capture patterns.
- tidy-html5 — Validate, fix, and reformat HTML(5), XHTML, and XML documents. Convert HTML to XHTML.
- tq — Query HTML with CSS selectors.
- Xidel — Query or modify XML and HTML pages with XPath, XQuery 3, and CSS selectors.

-
- `xml2` — Convert XML and HTML to and from flat, greppable lists of “path=value” statements. Source code mirror.
 - `xpe` — Query HTML and XML with XPath expressions.

JSON

- `Cels` — Patch JSON, TOML, and YAML with patches in the same format with some special values. Can be used as a Python library.
- `clconf` — Merge multiple config files and extract values from them using path string. Supports JSON and YAML. Can be used as a Go library.
- `dasel` — Query and update data structures from the command line. Comparable to `jq/yq` but supports CSV, JSON, TOML, YAML, and XML. Static binaries available for releases.
- `emuto` — CLI tool similar to `jq`. Create and manipulate CSV, TSV, and JSON. Can be compiled to JavaScript.
- `fastgron` — Convert JSON to and from GRON, a flat, greppable list of `path=value` statements. Much faster than the original `gron` on large files.
- `ffs` — Mount JSON, TOML, and YAML as a Unix filesystem.
- `fx` — Run arbitrary JavaScript on JSON input. Standalone binaries available.
- `gojq` — A pure Go implementation of `jq`. Supports YAML input and output.
- `Graphstage` — Compare and merge tree-like structures semantically. Supports JSON, JSON5, XML, HTML, YAML, and CSV. Can be used as a Python library.
- `gron` — Convert JSON to and from GRON, a flat, greppable list of `path=value` statements.
- `jaq` — A Rust implementation of `jq` with minor changes to the language to make it more predictable.
- `JC` — Convert the output of standard command-line tools to JSON.
- `jello` — Query JSON and JSON Lines with Python code. Output the result in a line-based format suitable for creating Bash arrays. Generate a grep-able schema.
- `jet` — Convert between JSON, YAML, Clojure’s `edn`, and `Transit`. Transform them with Clojure code.
- `jfq` — Query and transform JSON with the JSONata language.
- `jj` — Query and modify values in JSON or JSON Lines with a key path.
- `jl` — Query and manipulate JSON using a tiny functional language.
- `jo` — Create JSON objects from the shell.
- `jp (jmespath)` — Query JSON with JMESPath.
- `jp (sgreben)` — Plot JSON and CSV data in the terminal. Supports different kinds of plots: bar charts, line charts, scatter plots, histograms, and heatmaps.
- `jplot` — Plot real-time JSON data in the terminal (works with terminals supporting graphic rendering).

-
- `jq` — Create and manipulate JSON with a functional (as in “functional programming”) DSL. Can convert JSON to other formats.
 - `jql` — Create and manipulate JSON with a Lisp-syntax DSL.
 - `jshon` — Create and manipulate JSON using getopt-style command-line options.
 - `json` — Run arbitrary JavaScript on JSON input.
 - `json-patch` — Apply RFC 6902 JSON Patches to JSON. The CLI tool is secondary to a Go library that also creates and applies RFC 7386 JSON merge patches.
 - `json-table` — Convert nested JSON into CSV or TSV for processing in the shell.
 - `json.tool` — Validate and pretty-print JSON. This module is part of the standard library of Python 2/3 and is likely to be available wherever Python is installed. (Python 3 docs.)
 - `json2` — Convert JSON to and from flat, greppable lists of “path=value” statements. Modeled after `xml2`.
 - `jsonaxe` — Create and manipulate JSON with a Python-based DSL. Inspired by `jq`.
 - `jsonwatch` — Track changes in JSON data from the command line. Works like `watch -d`.
 - `jtbl` — Format JSON or JSON Lines as a plain-text table.
 - `jtc` — Create, manipulate, search, validate JSON with path expressions. Can be used as a C++14 library.
 - `lobar` — Process JSON and explore it interactively with a wrapper for `lodash.chain()`. An alternative to `jq` with JavaScript syntax.
 - `madato` — Convert ODS and XLSX spreadsheets to JSON, Markdown, and YAML.
 - `Mario` — Manipulate and convert between CSV, JSON, YAML, TOML, and XML with Python code.
 - `Nushell` — A command shell. Can natively load data from CSV, INI, JSON, TOML, TSV, XML, YAML, and other formats.
 - `pyp` — Transform input (as text lines or as a whole) using Python code with automatic module imports. Can generate a Python script equivalent to its invocation. In Python 3.11 or later supports TOML through `tomllib`.
 - `qpyson` — Query and manipulate JSON with Python.
 - `query-json` — A faster `jq` implementation written in Reason Native (OCaml).
 - `quicktype` — Infer the underlying model of the JSON and output as types for various programming languages or JSON Schema. CLI and Web UI.
 - `ramda-cli` — Manipulate JSON with the Ramda functional library, and either LiveScript or JavaScript syntax.
 - `RecordStream` — Create, manipulate, and output a stream of records, or JSON objects. Can retrieve records from an SQL database, MongoDB, Atom feeds, XML, and other sources.
 - `Remarshal` — Convert between CBOR, JSON, MessagePack, TOML, and YAML. Validate each of the formats. Pretty-print JSON, TOML, and YAML.
 - `rq` — Convert between Apache Avro, CBOR, CSV, JSON, MessagePack, Protocol Buffers, TOML, YAML, and awk-style plain text.
-

-
- Skeem — Infer SQL DDL statements from tabular data. Supports CSV, JSON, JSON Lines, ODS, XLSX, and other formats.
 - validjson — Validate or pretty-print JSON.
 - xml-to-json-fast — Convert XML to JSON. Can handle very large XML files.
 - xmljson — Convert multiple and large XML files to JSON. Written in Swift.
 - yaml-diff-patch — Patch YAML with RFC 6902 JSON Patches. Generate a JSON Patch from two JSON documents or a YAML and a JSON document. Preserves style. Can be used as a TypeScript library.
 - yamlpath — Query, modify, diff, merge, and validate YAML and JSON with YAML Paths. Also a Python library.

TOML

With a format converter like Remarshal you can use JSON tools to process TOML and YAML, but make sure you do not lose data in the conversion.

- Cels — Patch JSON, TOML, and YAML with patches in the same format with some special values. Can be used as a Python library.
- dasel — Query and update data structures from the command line. Comparable to jq/yq but supports CSV, JSON, TOML, YAML, and XML. Static binaries available for releases.
- ffs — Mount JSON, TOML, and YAML as a Unix filesystem.
- Mario — Manipulate and convert between CSV, JSON, YAML, TOML, and XML with Python code.
- Nushell — A command shell. Can natively load data from CSV, INI, JSON, TOML, TSV, XML, YAML, and other formats.
- pyp — Transform input (as text lines or as a whole) using Python code with automatic module imports. Can generate a Python script equivalent to its invocation. In Python 3.11 or later supports TOML through tomllib.
- Remarshal — Convert between CBOR, JSON, MessagePack, TOML, and YAML. Validate each of the formats. Pretty-print JSON, TOML, and YAML.
- rq — Convert between Apache Avro, CBOR, CSV, JSON, MessagePack, Protocol Buffers, TOML, YAML, and awk-style plain text.
- taplo-cli — Query, format, and validate (lint) TOML.
- validtoml — Validate TOML.
- yq (kislyuk) — jq wrapper for YAML, XML, and TOML.

XML

- csvfix — A multitool. Compare, filter, normalize, split, and validate CSV files. Reorder, remove,

-
- split, and merge fields. Convert data between fixed-width, multi-line, XML, and DSV format. Generate SQL statements. (Unofficial mirror.)
- **dasel** — Query and update data structures from the command line. Comparable to `jq/yq` but supports CSV, JSON, TOML, YAML, and XML. Static binaries available for releases.
 - **Graphstage** — Compare and merge tree-like structures semantically. Supports JSON, JSON5, XML, HTML, YAML, and CSV. Can be used as a Python library.
 - **hred** — Query XML and HTML with a query language based on CSS selectors.
 - **html-xml-utils** — A number of simple utilities (like [hxcopy](#), [hxpipe](#), [hxunent](#), [hxselect](#)) for manipulating HTML and XML files from W3C. Written in C, quite old-fashioned, but still relevant and maintained.
 - **Mario** — Manipulate and convert between CSV, JSON, YAML, TOML, and XML with Python code.
 - **Nushell** — A command shell. Can natively load data from CSV, INI, JSON, TOML, TSV, XML, YAML, and other formats.
 - **Saxon** — Query XML and HTML data with XPath. Documentation.
 - **sml2** — Convert between XML and SML, a simplified XML representation.
 - **tidy-html5** — Validate, fix, and reformat HTML(5), XHTML, and XML documents. Convert HTML to XHTML.
 - **Xidel** — Query or modify XML and HTML pages with XPath, XQuery 3, and CSS selectors.
 - **xml-to-json-fast** — Convert XML to JSON. Can handle very large XML files.
 - **xml2** — Convert XML and HTML to and from flat, greppable lists of “path=value” statements. Source code mirror.
 - **xmljson** — Convert multiple and large XML files to JSON. Written in Swift.
 - **XMLLint** — Query (including XSLT), validate and reformat XML documents.
 - **XMLStarlet** — Query, modify, and validate XML documents.
 - **xpe** — Query HTML and XML with XPath expressions.
 - **xq** — `jq` wrapper for XML documents.
 - **xsltproc** — Transform XML documents using XSLT and EXSLT.
 - **yq (kislyuk)** — `jq` wrapper for YAML, XML, and TOML.

See also

- “Grep and Sed Equivalent for XML Command Line Processing” on Stack Overflow.

YAML

- **Cels** — Patch JSON, TOML, and YAML with patches in the same format with some special values. Can be used as a Python library.

-
- `clconf` — Merge multiple config files and extract values from them using path string. Supports JSON and YAML. Can be used as a Go library.
 - `dasel` — Query and update data structures from the command line. Comparable to `jq/yq` but supports CSV, JSON, TOML, YAML, and XML. Static binaries available for releases.
 - `dy` — Construct YAML from a directory tree.
 - `ffs` — Mount JSON, TOML, and YAML as a Unix filesystem.
 - `gojq` — A pure Go implementation of `jq`. Supports YAML input and output.
 - `Graphstage` — Compare and merge tree-like structures semantically. Supports JSON, JSON5, XML, HTML, YAML, and CSV. Can be used as a Python library.
 - `jet` — Convert between JSON, YAML, Clojure’s `edn`, and `Transit`. Transform them with Clojure code.
 - `madato` — Convert ODS and XLSX spreadsheets to JSON, Markdown, and YAML.
 - `Mario` — Manipulate and convert between CSV, JSON, YAML, TOML, and XML with Python code.
 - `Nushell` — A command shell. Can natively load data from CSV, INI, JSON, TOML, TSV, XML, YAML, and other formats.
 - `Remarshal` — Convert between CBOR, JSON, MessagePack, TOML, and YAML. Validate each of the formats. Pretty-print JSON, TOML, and YAML.
 - `rq` — Convert between Apache Avro, CBOR, CSV, JSON, MessagePack, Protocol Buffers, TOML, YAML, and `awk`-style plain text.
 - `shyaml` — Query YAML. Can output null-terminated strings for use in shell scripts.
 - `validyaml` — Validate or pretty-print YAML.
 - `yaml-diff-patch` — Patch YAML with RFC 6902 JSON Patches. Generate a JSON Patch from two JSON documents or a YAML and a JSON document. Preserves style. Can be used as a TypeScript library.
 - `yaml-tools` — A set of CLI tools to manipulate YAML files (merge, delete, etc...) with comment preservation, based on `ruamel.yaml`.
 - `yamlpath` — Query, modify, diff, merge, and validate YAML and JSON with YAML Paths. Also a Python library.
 - `yq (kislyuk)` — `jq` wrapper for YAML, XML, and TOML.
 - `yq (mikefarah)` — Query, modify, and merge YAML. Convert to and from JSON.

Configuration files

`/etc/hosts`

- `hostctl` — Add and remove entries in `/etc/hosts`. Disable (comment out) and enable (uncomment) entries. Idempotent. Preserves arbitrary comments above its section of the hosts file. Works with groups of entries called “profiles”.

-
- **hostess** — Add and remove entries in `/etc/hosts`. Disable (comment out) and enable (uncomment) entries. Check if a hostname exists. Reformat the hosts file. Convert the entries to JSON. Idempotent. Removes arbitrary comments.
 - **hosts** — Add and remove entries in `/etc/hosts`. Change a hostname's IP address. Idempotent. Preserves arbitrary comments. Can be used as a Tcl library.

INI

- **cfget**
 - **Platform:** Any with Python 2.6-2.7?
 - **License:** GPL-2.0-or-later
 - **Description:** Retrieve properties as shell script commands to set the corresponding variables (with `--dump exports`). Retrieve properties' values as plain text. Substitute values from an INI file in an Autoconf-style template. Supports plug-ins. Chokes on section names and keys with spaces.
- **confget**
 - **Platform:** Free/Net/OpenBSD, Linux, likely others
 - **License:** BSD-2-Clause
 - **Description:** Retrieve properties and sections as shell script commands to set the corresponding variables. Retrieve properties' values as plain text. Check for existence of properties. List sections. Find values that match a pattern. Read-only. Has a C, Python, and Rust implementation. The Rust implementation can be installed with `cargo install confget`.
- **crudini**
 - **Platform:** Any with Python 2.6–2.7 or 3.x
 - **License:** GPL-2.0
 - **Description:** Retrieve properties and sections as INI fragments or shell script commands to set the corresponding variables. Retrieve properties' values as plain text. Set properties. Remove properties and sections. Create empty sections. Merge INI files. Changes files in place.
- **inicom**
 - **Platform:** Windows, POSIX
 - **License:** Apache-2.0
 - **Description:** Compare INI (and also Windows .reg) files.

-
- **IniFile**
 - **Platform:** Windows (x86, x86-64), MS-DOS
 - **License:** Closed-source freeware
 - **Description:** Retrieve properties and sections as batch file commands to set the corresponding variables. Set properties. Remove properties and sections. Changes files in place.
 - **initool**
 - **Platform:** FreeBSD, Linux, Windows
 - **License:** MIT
 - **Description:** Retrieve properties and sections as INI fragments. Retrieve properties' values as plain text. Set properties. Check for existence of properties and sections. Remove properties and sections. Outputs the updated INI file.
 - **Nushell** ([from ini](#))
 - **Platform:** Free/Net/OpenBSD, Linux, macOS, Windows
 - **License:** MIT
 - **Description:** Query and transform data with the Nushell language.

Multiple formats

- **Augeas** — Query and modify a number of file formats. Not all of the formats are equally well supported by Augeas and for some only a limited subset of all valid files can be parsed.
- **Elektra** — Query and modify configuration files. Shares Augeas' limitations when it comes to application-specific configuration files (it uses the same lenses), but has better support for generic formats such as JSON and INI.

Log files

- **Inav** — Query and watch log files. Has batch and interactive mode. Supported formats include the Common Log Format, CUPS page_log, syslog, strace, and generic timestamped messages. Can perform SQL queries.
- **Squawk** — Query Apache and Nginx log files. See the SQL-based tool comparison.

Multiformat tools

Tools that support multiple input formats. Programs that convert between only two formats in both directions are excluded. We only count JSON support that is separate from YAML.

-
- Augeas — Query and modify a number of file formats. Not all of the formats are equally well supported by Augeas and for some only a limited subset of all valid files can be parsed.
 - Cels — Patch JSON, TOML, and YAML with patches in the same format with some special values. Can be used as a Python library.
 - clconf — Merge multiple config files and extract values from them using path string. Supports JSON and YAML. Can be used as a Go library.
 - csvfix — A multitool. Compare, filter, normalize, split, and validate CSV files. Reorder, remove, split, and merge fields. Convert data between fixed-width, multi-line, XML, and DSV format. Generate SQL statements. (Unofficial mirror.)
 - csvtk — Search, sample, cut, join, transpose, and sort CSV/TSV files. Rename columns. Replace fields and generate new fields from existing fields. Plot data as vector or raster histograms and box, line, and scatter plots. Convert CSV to Markdown. Convert XLSX to CSV. Split XLSX sheets.
 - dasel — Query and update data structures from the command line. Comparable to jq/yq but supports CSV, JSON, TOML, YAML, and XML. Static binaries available for releases.
 - Elektra — Query and modify configuration files. Shares Augeas' limitations when it comes to application-specific configuration files (it uses the same lenses), but has better support for generic formats such as JSON and INI.
 - emuto — CLI tool similar to jq. Create and manipulate CSV, TSV, and JSON. Can be compiled to JavaScript.
 - ffs — Mount JSON, TOML, and YAML as a Unix filesystem.
 - fawk — A Rust implementation of a language partially compatible with AWK that supports parallelism and CSV input and output. fawk is an awk-derived language with a CSV mode for input and for output.
 - GoAWK — A cross-platform implementation of awk with added support for CSV. The project provides binaries for many platforms, including Windows. GoAWK is an awk implementation that adds a CSV mode for input and for output.
 - gojq — A pure Go implementation of jq. Supports YAML input and output.
 - Graphstage — Compare and merge tree-like structures semantically. Supports JSON, JSON5, XML, HTML, YAML, and CSV. Can be used as a Python library.
 - hred — Query XML and HTML with a query language based on CSS selectors.
 - html-xml-utils — A number of simple utilities (like [hxcopy](#), [hxpipe](#), [hxunent](#), [hxselect](#)) for manipulating HTML and XML files from W3C. Written in C, quite old-fashioned, but still relevant and maintained.
 - jet — Convert between JSON, YAML, Clojure's edn, and Transit. Transform them with Clojure code.
 - jp (sgreben) — Plot JSON and CSV data in the terminal. Supports different kinds of plots: bar charts, line charts, scatter plots, histograms, and heatmaps.

-
- Inav — Query and watch log files. Has batch and interactive mode. Supported formats include the Common Log Format, CUPS page_log, syslog, strace, and generic timestamped messages. Can perform SQL queries.
 - madato — Convert ODS and XLSX spreadsheets to JSON, Markdown, and YAML.
 - Mario — Manipulate and convert between CSV, JSON, YAML, TOML, and XML with Python code.
 - Nushell — A command shell. Can natively load data from CSV, INI, JSON, TOML, TSV, XML, YAML, and other formats.
 - pyp — Transform input (as text lines or as a whole) using Python code with automatic module imports. Can generate a Python script equivalent to its invocation. In Python 3.11 or later supports TOML through tomlib.
 - RecordStream — Create, manipulate, and output a stream of records, or JSON objects. Can retrieve records from an SQL database, MongoDB, Atom feeds, XML, and other sources.
 - ReadStat — Convert statistics package datasets between SAS (SAS7BDAT, XPORT), SPSS (POR, SAV, ZSAV), and Stata (DTA). Convert those formats to CSV and XLSX. Can be used as a C library with bindings for Julia, Python, and R.
 - Remarshal — Convert between CBOR, JSON, MessagePack, TOML, and YAML. Validate each of the formats. Pretty-print JSON, TOML, and YAML.
 - rows — A Python library with a CLI. Convert between a number of file formats for tabular data: CSV, XLS, XLSX, ODS, and others. Query the data (via SQLite). Combine tables. Generate schemas.
 - rq — Convert between Apache Avro, CBOR, CSV, JSON, MessagePack, Protocol Buffers, TOML, YAML, and awk-style plain text.
 - Saxon — Query XML and HTML data with XPath. Documentation.
 - Skeem — Infer SQL DDL statements from tabular data. Supports CSV, JSON, JSON Lines, ODS, XLSX, and other formats.
 - tidy-html5 — Validate, fix, and reformat HTML(5), XHTML, and XML documents. Convert HTML to XHTML.
 - VisiData — Explore interactively data in TSV, CSV, XLS, XLSX, HDF5, JSON, and other formats. Introduction.
 - Xidel — Query or modify XML and HTML pages with XPath, XQuery 3, and CSS selectors.
 - xml2 — Convert XML and HTML to and from flat, greppable lists of “path=value” statements. Source code mirror.
 - xmljson — Convert multiple and large XML files to JSON. Written in Swift.
 - xpe — Query HTML and XML with XPath expressions.
 - yaml-diff-patch — Patch YAML with RFC 6902 JSON Patches. Generate a JSON Patch from two JSON documents or a YAML and a JSON document. Preserves style. Can be used as a TypeScript library.

-
- `yamllpath` — Query, modify, diff, merge, and validate YAML and JSON with YAML Paths. Also a Python library.
 - `yq` (`kislyuk`) — `jq` wrapper for YAML, XML, and TOML.
 - `zsv` — Slice, combine, reformat, flatten/unflatten CSV (TSV, DSV) files. Query them with SQL and `jq` filters. Convert between them, JSON, and SQLite 3. Also a C library.

Templating for structured text

Listed below are restricted programming language interpreters and templating tools that produce structured text output. They are generally intended to remove repetition in configuration files. They are distinct from unstructured templating tools like the `jinjja2` CLI program, which should not be added to this table.

- CUE
 - **Output format:** JSON
 - **Turing-complete:** No
 - **Syntax:** Extended JSON
 - **I/O:** ?
 - **Description:** A constraint language for JSON configuration data. Can generate and validates JSON.
- Dhall
 - **Output format:** JSON, YAML
 - **Turing-complete:** No
 - **Syntax:** Haskell-inspired
 - **I/O:** Limited to importing libraries from files and HTTP(S) URLs (with protection against leaking your data to the server)
 - **Description:** A statically-typed functional configuration language. Has a standard formatting tool.
- jk
 - **Output format:** JSON, YAML, plain text
 - **Turing-complete:** Yes
 - **Syntax:** JavaScript
 - **I/O:** Disk I/O
 - **Description:** Generate configuration files using JavaScript (V8 VM).
- Jsonnet

-
- **Output format:** JSON, INI, XML, YAML, plain text
 - **Turing-complete:** Yes
 - **Syntax:** Extended JSON
 - **I/O:** None
 - **Description:** A functional configuration language. Has a standard formatting tool.
 - Nickel
 - **Output format:** JSON, TOML, YAML
 - **Turing-complete:** Yes
 - **Syntax:** Inspired by ML and JSON
 - **I/O:** Limited input is to be implemented
 - **Description:** A gradually-typed functional configuration language with contracts.
 - Pkl
 - **Output format:** JSON, YAML, macOS property list, Java `.properties`
 - **Turing-complete:** Yes
 - **Syntax:** Swift-inspired
 - **I/O:** The CLI can read environment variables and files, [GET](#) HTTP(S) URLs. It can import modules from files and HTTP(S) URLs.
 - **Description:** A command-line tool, Java library, and build tool plugin. Can generate code for Go, Java, Kotlin, and Swift. “Pkl vs. Other Config Languages”.
 - rjsone
 - **Output format:** JSON, YAML
 - **Turing-complete:** No?
 - **Syntax:** Extended JSON
 - **I/O:** None
 - **Description:** A CLI tool for the JSON-e templating language.
 - ytt
 - **Output format:** YAML
 - **Turing-complete:** No
 - **Syntax:** YAML/Python hybrid
 - **I/O:** None?
 - **Description:** A templating tool for YAML built upon the Starlark configuration language.

See also

- A comparison table of Nickel and other configuration languages.

Extra: interactive TUIs

- `jid` — Explore JSON interactively with filtering queries like `jq`.
- `jiq` — Explore JSON interactively with `jq`. Requires `jq`.
- `lobar` — Process JSON and explore it interactively with a wrapper for `lodash.chain()`. An alternative to `jq` with JavaScript syntax.
- `sc-im` — A Vim-like spreadsheet calculator for CSV and TSV files.
- `VisiData` — Explore interactively data in TSV, CSV, XLS, XLSX, HDF5, JSON, and other formats. [Introduction](#).

Extra: CLIs for single-file databases

- `Firebird`
 - **Description:** Firebird is a FOSS database that can be used from a single file, like SQLite. “`isql` is a program that allows the user to issue arbitrary SQL commands”.
 - **File format:** Binary
- `Fsdb`
 - **Description:** A flat-file database for shell scripting.
 - **File format:** Text-based, TSV with a header or “key: value”
- `GNU Recutils`
 - **Description:** “[A] set of tools and libraries to access human-editable, plain text databases called recfiles.”
 - **File format:** Text-based, roughly “key: value”
- `SDB`
 - **Description:** “[A] simple string key/value database based on djb’s `cdb` disk storage and supports JSON and arrays introspection.”
 - **File format:** Binary
- `sqlite3(1)`
 - **Description:** “[A] simple command-line utility [...] that allows the user to manually enter and execute SQL statements against an SQLite database.”
 - **File format:** Binary

License

The contents of this document is licensed under the Creative Commons Attribution 4.0 International License. By contributing you agree to release your contribution under this license.

Disclosure

csv2html, hosts, Sqawk, jsonwatch, Remarshal, and initool are developed by the curator of this document.