

mandoc: becoming the main BSD manual toolbox

BSDCan 2015, June 13, Ottawa

Ingo Schwarze <schwarze@openbsd.org>



Cynthia Livingston's OTTB "Bedifferent" (c) 2013 C. Livingston (with permission)

Brief history of UNIX documentation

- The key point: All documentation in one place and one format. Easy to find, uniform and easy to read and write. Be correct, complete, concise.
- 1964: RUNOFF/roff markup syntax by Jerome H. Saltzer, MIT. Unobtrusive, diff(1)-friendly, easy to hand-edit, simple tools, high quality output.
- 1971: Basic manual structure by Ken Thompson and Dennis Ritchie for the AT&T Version 1 UNIX manuals, Bell Labs.
- 1979: man(7) physical markup language for AT&T Version 7 UNIX.
- 1989: **mdoc(7) semantic markup by Cynthia Livingston** for 4.3BSD-Reno. Powerful, self-contained, portable.
- 1989: GNU troff by James Clarke.
- 2001: mdoc(7) rewrite by Werner Lemberg and Ruslan Ermilov for groff-1.17.
- 2008: mandoc(1) started by Kristaps Dzonsons.
- 2010: mandoc(1) is the only documentation formatter in the OpenBSD base system.
- 2014: mandoc(1) used by default in OpenBSD, FreeBSD, NetBSD, illumos.



What is the mandoc toolbox?

User perspective: man(1), the manual viewer

One comprehensive tool!

Normal operation always proceeds in three steps:

1. **Find** one or more manuals in the file system or using a database
by manual name — **man(1)** —
or by search query — **apropos(1)** = `man -k`
The result of this step can be printed out with `man -w`.
2. **Format** the manual page.
This step can be accessed individually with **mandoc(1)** = `man -l`.
3. **Display** the formatted text.
On a terminal device, use a pager unless `-c` is given.

Progress since last year

- In 2014, `man(1)` was still out of scope; now we have a unified user interface.
- Much simplified `man.conf(5)` configuration file.

Auxiliary components in the toolbox

- makewhatis(8) database generation and maintenance
- mandoc(1) -Tlint syntax checker
- mandoc(1) -Thtml polyglot HTML5 generator
- mandoc(1) -Tpdf and -Tps formatters
- mandoc(1) -Tman format converter
- mandoc(1) -Ttree parse tree debugger
- soelim(1) file inclusion resolver
- ...



Moorside Cottage, Mackenzie King Estate, Gatineau Park © 2009 Mac Armstrong @flickr (CC)

Which are the topics of this talk?

1. Unified user interface: `man(1)` and `man.conf(5)`.
2. More powerful search on the web: `man.cgi(8)`.
3. Better formatting of equations: `eqn(7)`.
4. No more options required for UTF-8 support.
5. Eradicating bugs with `afl(1)`, the American Fuzzy Lop.
6. Detect use of unsupported features: `-Wunsupp`.
7. Help LibreSSL documentation with `pod2mdoc(1)`.
8. Status in various operating systems and future directions.



Black Lake near King Mountain, Gatineau Park, Quebec © 2012 Lezumbalaberenjena@flickr (CC)

Replacing the manual viewer

- OpenBSD no longer uses the traditional BSD man(1) manual viewer, but an implementation of man(1) integrated in mandoc.
- Traditional setup:
man(1) finds the files in the file system and forks and executes the formatter (groff, mandoc) and the pager (more, less).
- New setup:
man(1) integrated in mandoc finds the files in the database or the file system and runs the formatter without forking, then forks and executes the pager.

- First portable release 1.13.2 on December 13, 2014
- Switched on in OpenBSD on December 14, 2014



Paroisse Sainte Cécile de Masham, Quebec
© 2013 Lezumbalaberenjena@flickr (CC)

Steps to replace the manual viewer

Preliminaries (Kristaps Dzonsons)

- Concept of a manual page search database (April 2, 2011)
- Create it: Implementation of makewhatis(8) (release 1.11.2, May 12, 2011)
- Use it: Implementation of apropos(1) (release 1.12.0, October 8, 2011)
- Port to SQLite3 (June 8, 2012)

Main steps (Ingo Schwarze)

- Polishing the SQLite3 implementation (t2k13 hackathon, June 2013)
- Support compressed manuals (makewhatis(8): March 26, man(1): September 3, .so: November 26, 2014)
- Switch makewhatis(8)/apropos(1) to mandoc in OpenBSD (April 18, 2014)
- First portable release with SQLite3 (1.13.1, August 10, 2014)
- Unified user interface and main program (August 16-26, 2014)
- Options man(1) -l and -h (August 30, September 3, 2014)
- File system lookup (January 13, 2015)
- portable: man(1) even without database support (February 3, 2015)

A better manual viewer

Current advantages

- Unified interfaces:
man can use: -W -T -O -I from mandoc(1)
apropos can use: -w -h -a from man(1)
- Allow much simpler man.conf(5) format → next slides
- database priority now overrides section priority
- Use additional names from .Dt/.TH and NAME .Nm (250 new entries in OpenBSD, compared to 10200 existing ones)
- One less userland program to maintain (that had rather old code)



Bonnechere Museum, Eganville
© 2011 Doug Kerr @flickr (CC)

Possible future advantages

- Possibility to get rid of multiple ln(1) links to the same manual
- Possibility to implement an interactive chooser (-i)

A better manual viewer (2)

Prices to pay

Database lookup is slightly slower than file system lookup:
Ten milliseconds additional delay for displaying a manual on my notebook.

After installing new manuals, `makewhatis(8)` is required for full database support. However, `pkg_add(1)` does that automatically and `man(1)` works even without it.



Opeongo Road, historic colonization road
© 2008 P199@wikimedia (CC)

Very little downsides, if any!

Undesirable features of the old man.conf(5) format

- Shell globbing for directory tree names
- Different directory trees for man(1) and apropos(1)
- Different directory tree search order for man(1) and apropos(1)
- Custom file names for mandoc.db(5) files in individual directory trees
- Custom subdirectory names not of the form man+section or cat+section
- Default search: Additional directories outside all sections
- Default search: Exclusion of certain sections in certain directory trees
- Section-specific search: Additional directories outside any configured trees
- Section-specific search: Exclusion of some directories configured by default
- Different search order for different sections
- Different decompression filters for different sections
- Custom file name suffixes not of the form *.section, with globbing
- Custom file name suffix for preformatted pages instead of .0, with globbing
- Configuration file versioning

So complicated that everybody hated it and almost nobody used it

Current features of the new man.conf(5) format

- **manpath** *directory* — configure a tree automatically consistent across man(1), apropos(1), makewhatis(8)
- **output** *options* — for example:
 - for the terminal: output width 104
 - for HTML: output style /man.css
 - for ps/pdf: output paper a4

Very easy to use, needs almost no learning



Wilno, Renfrew County, Ontario © 2009 P199@wikimedia (CC)

man.conf(5) features to be implemented on demand

- **alias** *aliasname dirname* — make a tree accessible with -s
- **sections** *name name name ...* — section names and search order
- **filter** *suffix command* — custom filters, in particular for decompression
- **pager** *command* — configure a default pager
- **input** *option value* — for example input os name

So far, there is no noticeable demand -> KISS.



Madawaska River at Whitney © 2006 P199@wikimedia (PD)

The new man.cgi(8) for manual pages on the web

- OpenBSD no longer uses the traditional man.cgi Perl script by Wolfram Schneider (wosch@FreeBSD) but a man.cgi(8) implementation included in the mandoc toolkit.
- Traditional setup: The man.cgi Perl script forks and executes the system's man(1) or apropos(1) commands, parses the output, and converts it to HTML.
- Traditionally, the spawned man(1) in turn forked and executed groff, then later — about since 2012 — mandoc, but not in HTML mode.
- New setup: man.cgi(8) is a C program linked against the mandoc parsers, the mandoc.db(5) database client code, and the mandoc HTML formatter.



Common Loons © 2009 Gary J. Wege @USFWS (CC)

Steps to replace the web manual viewer

Preliminaries

- Everything needed for man(1)
- Berkeley-db based man.cgi(8) by Kristaps Dzonsons (developed since Nov 9, 2011; first release 1.12.1, March 23, 2012)
- New makewhatis(8)/apropos(1) based on SQLite3 in OpenBSD since April 18, 2014



Cache Lake, Algonquin Park
© 2005 Ptrbnsn@wikimedia (CC)

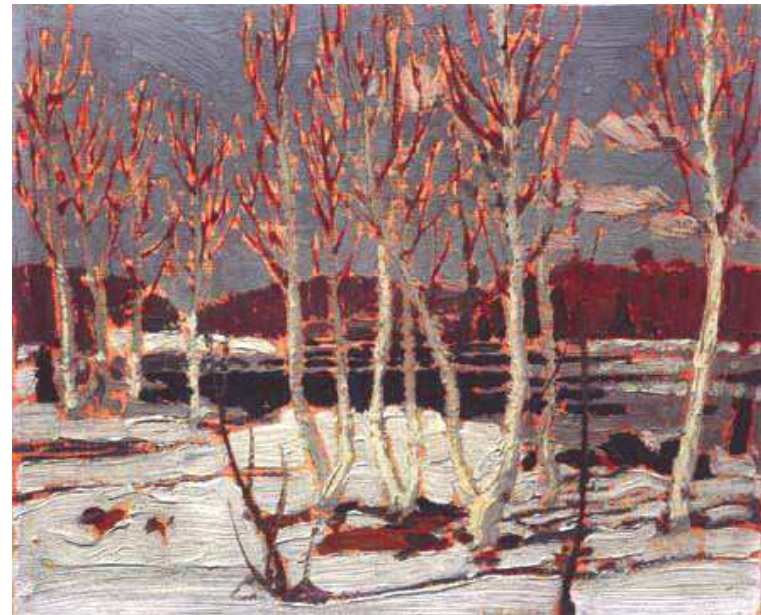
Main steps

- OpenBSD g2k14 general hackathon in Ljubljana, July 8-14
- Convert to SQLite3, polish, then switch over on July 12
- A some additional functional fixes on July 13-25
- Security audit by Sebastien Marie on July 18-23
- First portable release: 1.13.1 on August 10
- Release with HTML5 output: 1.13.2 on December 13
- All by Ingo Schwarze, 2014

Polishing the web manual viewer

Even though all underlying functionality was already quite mature when i started, there were surprisingly many things to polish in many distinct components:

- Configuration: Let manpath.conf be a plain text list of the directories to use. Do almost all other configuration at compile time, avoid run-time configuration to reduce the attack surface.
- HTTP parser: make HTTP decode linear in time (tedu@)
- Query parser:
Various backward-compatibility features to not break existing links in static pages around the web. Support backslash-escaping of white space in the query expression (reported: Marcus Merighi, Sep 14)
- Page selection:
Distinguish between man(1) and apropos(1) mode. Even when there are multiple pages, show one of them.



April in Algonquin Park © 1917 Tom Thomson (PD)
Memorial Art Gallery, Owen Sound, Georgian Triangle

Polishing the web manual viewer (2)

- Start page: section and architecture dropdowns, the latter with a default of "All Architectures", and some additional polishing.
- HTTP output: HTTP headers must end lines with CRLF (tedu@)
- Query printer: include manpath=, omit empty parameters, preserve manpath and arch in .Xr links
- HTML output: use manroot as the default for .Os and start moving towards polyglot HTML5 (completed Sep 27)
- Error handling: make it clear, the code readable, and avoid information leakage
- Documentation: full rewrite of the man.cgi(8) manual
- Internals: namespace cleanups, in particular of global variables

Various features were developed for man.cgi(8) that can be useful elsewhere, too:

- The mandoc -Ios option to handle multiple OS versions in parallel by a single installation (already on May 24, 2012)
- The MANSEARCH_MAN internal mode in mansearch.c, reused for man(1)
- Sort result pages first by section number, then by name; that's also nice for command-line apropos(1).
- Unify console and CGI section titles (August 26)

Security issues of the web manual viewer

- Basic problem: turning untrusted input into harmful processing or output
- Sources of untrusted input: `PATH_INFO`, `QUERY_STRING`, manual page content
- Harmful processing or output: Information leakage, resource exhaustion, misleading or dangerous HTTP headers or HTML content
- Relevant modules: CGI steering program (`cgi.c`) and HTML formatter (`html.c`)

Auditing approaches

- Start with all untrusted input, trace forward, and check how it is used.
- Locate all output statements (`printf`, `puts`, `putchar...`), trace backward, and check where the data comes from.
- Locate all interfaces between the steering program and the HTML formatter, identify the data passed through the interface, and trace backward and forward to see where it comes from and what is done with it.

Almost all security issues were initially reported by Sebastien Marie, but i did the full audit described above to make sure nothing was missed. I also started an infrastructure for regression tests of HTML output (October 27).

Security issues found in the web manual viewer

- Invalid configuration → segfault:
Fix: When there is no MAN_DIR or manpath.conf, log and exit.
- Unvalidated PATH_INFO → access to unrelated files, information disclosure:
Fix: Reject absolute paths and ascension to the parent directory.
- Unvalidated QUERY_STRING manpath → info disclosure in error message:
Fix: Validate the manpath up front using a whitelist.
- Invalid characters in QUERY_STRING → XSS:
Fix: Restrict the character set of strings passed into html_alloc().
- Roff escape sequences and quotes in manuals → XSS:
Fix: HTML-encode quotes and rendered escape sequences.
- Choosing the right encoding is a tricky business...
Some output needs HTML encoding, some URI encoding, some both.
- Crafted expensive regular expressions → REDoS attacks:
No full fix possible. Mitigate by limiting CGI process execution time.



Maggie Lake
Algonquin Park

© 2011 Ryan Tir
@flickr (CC)

Equations in manual pages

Relevance of the eqn(7) language:

Used less than mdoc(7), man(7), and tbl(7); but: X.org!

Problem was: parsing works well, formatting was ugly...

Representing mathematical formula on the terminal is hard.

Kristaps mostly completed the parser in 2011.

(started Feb 6, released 1.10.10 Mar 20, mostly complete in 1.11.5 on July 24).

The parser is quite straightforward, just over 1000 lines.

Main loop:

- Parse a token.
- Generate a syntax tree node.
- If the token is binary, relink the preceding subtree.



Archer Daniels Midland © 2013 Sean Marshall @flickr (CC)

Better formatting of equations

- **HTML output:** rewrite to generate MathML (Kristaps Dzonsons, Oct 10)
straightforward, 1:1 translation of the syntax tree, less than 200 lines of code
quite beautiful in a graphical browser
- **Terminal output:**
 - GNU eqn: moves elements up and down to the previous or next line and draws bars out of ASCII characters — results are unintelligible
 - mandoc output rewrite (Ingo Schwarze, Oct 12):
linear textual representation, showing fractions like $(a + b)/(c + d)$,
matrices like $((a_{11} \ a_{12})(b_{21} \ b_{22}))$, and roots like \sqrt{x}
 - certainly not pretty, but at least you can figure out what the formulas mean
 - merely 125 lines of very straightforward code
- I added support for in-line equations (Oct 16). That helps a lot to not lose mathematical symbols embedded in the middle of the text, in various manuals.
- mandoc eqn now much better than GNU eqn for both terminal and HTML
PostScript/PDF is still the domain of GNU eqn: mandoc just does the same as on the terminal.

All that was first released together with HTML5 output in 1.13.2 on Dec 13, 2014.

Multibyte character support

Non-english manuals:

rare — hard to maintain — worse than nothing when outdated...

But the tools must not hinder reading them!

That would only aggravate the problem.

Hence, early basic UTF-8 support (Kristaps Dzonsons, May 20-26, 2011):

Like groff, use a `preconv(1)` preprocessor: UTF-8 → roff escape sequences and `-Tutf8` and `-Tlocale` terminal output modes.

Improvements in 2014

- integrate `preconv(1)`:
direct UTF-8 input (Oct 25)
- default to `-Tlocale` (Dec 2)
(formerly `-Tascii`)
- No more options needed!
- released with 1.13.2 (Dec 13)
- rewrite UTF-8 parser (Dec 19)
- all by Ingo Schwarze



Wasaga Beach, L. Huron © 2013 Joe Mabel @wikimedia (CC)

American Fuzzy Lop inspecting mandoc

Fuzzer: Try to crash or hang a program by feeding it varying input.

<http://lcamtuf.coredump.cx/afl/>
"Compile-time instrumentation and genetic algorithms to automatically discover clean, interesting test cases that trigger new internal states."

Goal: Full functional coverage.

For mandoc, afl can be seeded with the extensive test suite.

For mandoc, full functional coverage requires running afl continuously for several days on modern PC hardware.



American Fuzzy Lop © 2008 Lithonius@wikimedia (PD)

Run by Jonathan Gray (jsg@) repeatedly since November 21, 2014.

45 issues grand total → Quantitative (but not statistical) analysis.

Root causes of bugs found with afl

- 15/45 logic errors arising from complexity
 - 9/15 excessive complexity in language design (badly nested blocks)
 - 6/15 excessive complexity in the implementation of macro rewinding
- 14/45 exceptions from invariants
 - 6/14 from supposedly generic invariants
 - 8/14 from macro-specific invariants
- 6/45 missing elementary input validation (too large; too small; zero; negative; invalid byte in binary data)
- 5/45 buffer overflows: string parser passes terminating `\0`
- 3/45 use after free of an allocated pointer
- 1/45 abuse of `assert()` for input validation
- 1/45 excessive power of `roff(7)` language: user input requests infinite recursion



Spiritcatcher by Ron Baird (1986) in Barrie
© 2009 Tudor Costache @wikimedia (CC)

Severity of bugs found with afl

- 2 write buffer overruns (both advancing a pointer past `\0` and returning it)
- 3 read buffer overruns (all advancing a pointer past `\0` without returning it)
- 2 use after free (both free a pointer, keep a copy, and use it)
- 2 unsigned integer underflows (by unchecked subtraction and unvalidated input)
- 2 cases of effectively infinite output (by not checking input for large numbers)
- 2 infinite loops (one by user input, one by complicated internal logic)
- 1 division by zero (by unvalidated user input)
- 11 NULL pointer accesses
- 20 assertion failures

Here: The easier to avoid, the more dangerous the failure.

Not guaranteed for other projects, but still, the following are likely to pay off:

1. Be particularly careful when passing around parse pointers.
2. Make sure you don't parse beyond `\0`.
3. Watch out for use after free.
4. Don't forget to validate any input.
5. Be careful with arithmetic operations (`/`, `*`, `-`, `+`, etc.)

Conclusions from bug-hunting with afl

- Largest numbers of bugs in most complex code → avoid complexity.
- Distribution across modules consistent with flat distribution → keep code small.
- Even mature open source projects have bugs.
Not unusual: about 0.5 to 3 serious bugs per 1000 lines of C code.
- Aggravating factors in this case: No formal language definitions, languages not designed according to any strict paradigm, languages evolved historically, parser requirements and design goals discovered piece-meal, again and again breaking existing logic and existing invariants.
- If you can afford to spend the effort, explicitly specify the invariants you intend to guarantee, audit all your code changing data structures to make sure these are actually maintained, and audit all your code using data structures to make sure no other assumptions are made. That may catch a substantial fraction of severe bugs even without fuzzing, about one third in the case of mandoc.



Barrie across Kempenfelt Bay, Lake Simcoe, Ontario © 2010 Jliu@wikimedia (CC)

Three message levels

1. Warnings: Tell manual authors that the quality of the code can be improved.
2. Errors: Tell manual authors that the intent of the code is unclear.
→ Likely information loss or seriously bad formatting.
3. Unsupported: Tell porters that a manual page uses features not (yet) supported by mandoc and groff should better be used instead of mandoc.
4. The former fatal level no longer exists:
It meant that mandoc(1) was unable to produce any output at all.
That can no longer happen (since January 15, 2015).

Several major cleanups
of messages and levels in mandoc:

July 2009, May 2010,
August 2010, October 2010,
January 2011, March 2011,
July 2014, January 2015, ...



Toronto © 2008 Paul Bica @flickr (CC)

Manual pages in ports

By now, 95% of ports manuals just work with mandoc.

The OpenBSD way

- Mark those that don't work individually with `USE_GROFF` (about 200 remain).
- Preformat these manuals at port build time with `groff`, package the preformatted versions of the manuals.

Advantage:
Perfect manuals for every port.

Inconveniences:
Needs support in the
ports infrastructure
(written by Marc Espie@),
and manual maintenance
of the `USE_GROFF` variable.



Humber Bay Bridge, Toronto © 2010 Paul Bica @flickr (CC)

Manual pages in ports (2)

- The FreeBSD way:
 - Inside man(1), try mandoc -Tlint -Wunsupp and let mandoc itself decide whether it accepts the job. If so, use mandoc, otherwise groff.
 - Advantage: No manual checking of ports required.
 - Inconveniences: Double parsing at man(1) call time.
False negatives: mandoc doesn't realize it's unfit for the job
→ user gets incomplete or misformatted manual
False positives: mandoc feels shy even though its output is fine
→ time wasted for running groff
- The NetBSD way:
 - Just ignore the whole problem (if i understand correctly).
 - Seems to be good enough for them, i don't hear any complaints regarding those about 5% broken ports manuals in NetBSD.



Bloor Viaduct, Don Valley, Toronto
© 2013 Paul Bica @flickr (CC)

Possible future directions for ports manuals

- For the OpenBSD way: Improve low-level roff(7) support in mandoc(1) reducing the number of USE_GROFFs required.
 - Chances aren't bad, there are some big classes of problems it might be possible to solve, so we can certainly get below 200.
 - But, at some point, there will be diminishing returns...
 - naddy@ eventually wants to get rid of USE_GROFF altogether.
 - Not yet quite sure how it will be done.
- For the FreeBSD way: Improve -Wunsupp logic.
 - Detection can certainly be improved in some typical situations.
 - But we will run into diminishing returns, too; the number of ways to abuse low-level roff in manuals is boundless.
 - Not yet sure how good it can become.



Tommy Thompson Park
Toronto, Ontario

© 2008
Kiril Strax
@flickr (CC)

Getting started with pod2mdoc(1)

- `perlpod(1)`: maybe the third most important documentation format after `mdoc(7)` and `man(7)`; used by Perl and many other projects, for example OpenSSL
- usually converted to `man(7)` with `pod2man(1)` written in Perl
- downsides: no semantic searching, LibreSSL developers have to cope with a second (and less useful) documentation language, consumes time during the build, `pod2man(1)` `man(7)` output contains ugly low-level roff(7) and keeps getting worse, the `pod2man(1)` utility itself is hackish and ugly
- Hence we are working on `pod2mdoc(1)`, a conversion tool from `perlpod(1)` to `mdoc(7)` format.
- During g2k14 in Ljubljana (July 2014), Anthony Bentley started to convert the LibreSSL `libssl` manuals from `perlpod(1)` to `mdoc(7)` with `pod2mdoc(1)`, mostly based on the version Kristaps developed in March/April 2014, containing only minor fixes on my part (versions 0.0.12 on July 11 and 0.0.13 on July 19).
- With those versions of `pod2mdoc(1)`, considerable manual postprocessing was still required.
- The completed work was finally committed on October 12.
- One important benefit: An extensive list of bug reports and feature requests.

Improvements of pod2mdoc(1)

In February 2015, i started work on LibreSSL libcrypto, improving pod2mdoc(1) in parallel, using Anthony's list and my own observations.

- whitespace improvements
 - preserve blank lines inside literal displays (Feb 19)
 - re-wrap text lines (Feb 23)
 - new sentence, new line (Feb 14)
 - print whitespace before words, never after (Feb 13)
 - horizontal spacing fixes (Feb 20)
- handle all closing punctuation (Feb 19)
- escape literal double quotes at the start of macro arguments (Feb 19)



Theological Hall, Queen's University, Kingston
© 2011 Canteaus@wikimedia (CC)

Such polishing may seem minor but is actually important:

The goal is to commit the converted manuals.
So the generated code must be clean and maintainable.
Cleaning it up by hand is very tedious.

New concepts in pod2mdoc(1)

- SYNOPSIS rendering for library manuals:
 - function prototypes -> .Ft/.Fo/.Fa/.Fc (Oct 22, version 0.1)
 - render preprocessor lines with .Fd (Feb 13)
- ohash-driven markup
 - function names with .Fn (Feb 13)
 - function args with .Fa (Feb 13 and 23)
 - type names with .Vt (Feb 13)
 - #defined names: .Dv or .Fn (Feb 19)
- foreign function names: .Xr (Feb 14)



Kingston City Hall © 2013 Mkooiman@wikimedia (CC)

All released with pod2mdoc-0.2 on May 19, 2015.

Similar logic is needed for legacy man(7) manuals.

Status in OpenBSD

Status last year, at BSDCan 2014



- Kristaps@ developed mandoc(1) since November 22, 2008.
- Schwarze@ maintaining it since June 14, 2009.
- Base system manuals built with mandoc(1) since April 3, 2010.
- Releases fully rely on mandoc(1) since OpenBSD 4.8, November 1, 2010.
- Groff disconnected from base build since October 18, 2010:
mandoc(1) is the only documentation formatter in base for almost five years.
- No stable releases contain groff since OpenBSD 4.9, May 1, 2011.
- Install manual sources, not preformatted manuals since June 23, 2011.
- SQLite-based code in the source tree since December 30, 2013.
- **makewhatis(8)/apropos(1) using mandoc** since April 18, 2014 (rel. with 5.6).

New since last year

- New man.cgi(8) online on www.openbsd.org since July 12, 2014.
- Unified interface for mandoc(1) and apropos(1) since August 26, 2014.
- **man(1) switched to the mandoc implementation** on December 14, 2014.
- All released with OpenBSD 5.7 on May 1, 2015.

Status in FreeBSD



Status last year, at BSDCan 2014

- An mdocml port by Ulrich Spörlein exists since March 9, 2009.
- Source code in the base repo and installed by default since October 19, 2012.
- **First released with FreeBSD 10.0 on January 20, 2014.**

New since last year

- All work done by Baptiste Daroussin (bapt@).
- **mandoc(1) is the default formatter** since November 23, 2014.
- Using -Wunsupp since March 2, 2015.
- **apropos(1)/makewhatis(8) default to the mandoc versions** since May 30, 2015
- Code is up to date using the portable stable release mandoc-1.13.3.
- To be released with FreeBSD 11 in the future.
- Still using the man(1) implementation written as a sh(1) script.

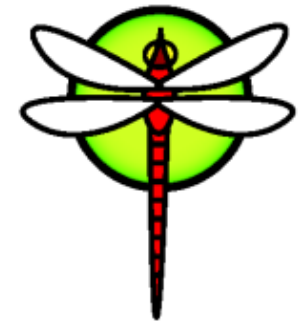
Status in NetBSD

- pkgsrc version well maintained by Thomas Klausner (wiz@) and up to date.
- Apart from that, no progress since last year. mandoc(1) version is outdated (1.12.3, released 2013, no reaction to the three newer versions of mandoc released after that).
- Installing source manuals.
- Using mandoc(1) as the run-time manual formatter since February 7, 2012.
- NetBSD-only apropos(1)/makemandb(8) featuring full-text search but no semantic search.



Status in DragonFly BSD

- No progress since last year. mandoc version in base is outdated (1.13.1, released in August 2014, no reaction to the two versions of mandoc released after that).
- mandoc(1) is installed, but not used.
- Semantic searching is not yet supported, not even as an option.



What happened in illumos?

- Mandoc is contained in the base system **and used by default for formatting manuals** since July 21, 2014 (Garrett D'Amore).
- The first non-BSD system to accomplish this, and the third system grand total.
- Have decided to gradually upgrade their man(7) pages to mdoc(7) format.
- Using the same old version as NetBSD (1.12.3, released in 2013).
- Still using their own apropos(1)/catman(8), no support for semantic searching.



Status in Linux

- Two Linux distributions have mandoc fully enabled by default:
- Alpine Linux: mandoc(1) since June 12, 2011;
man(1), apropos(1), makewhatis(8) since December 29, 2014
- Void Linux: mandoc(1) since September 23, 2014;
man(1), apropos(1), makewhatis(8) since March 14, 2015
- Both using the latest stable portable mandoc (1.13.3)
- Alpine Linux was the first non-BSD system ever to use the mandoc-based man(1), apropos(1), and makewhatis(8).
- Apart from OpenBSD, i know of no other operating systems that rely more heavily on mandoc right now than these two.
- Definitely, these prove that fully switching a Linux distribution to use the complete mandoc suite is feasible.
- Arch Linux has an official port, Slackware and Crux unofficial ones, no major Linuxes have followed so far, even though mandoc is known to work on Debian and Ubuntu, Fedora and RHEL and CentOS, OpenSuSE and SLES.



Little Blue Penguin © 2009
Flagstaffotos @wikimedia (CC)

Status in other operating systems

Minix 3

Source code in the base repo since June 26, 2010 (Ben Gras).

Completely apathetic, still using a version that is more than five years old.

Mac OS X

1.13.2 available on Homebrew

1.13.1 available on MacPorts

Microsoft Windows

1.13.1 package available built with MinGW

1.12.2 only an outdated package for Cygwin

IBM AIX

Outdated unofficial 1.12.0 port



Periodically updated information:

<http://mdocml.bsd.lv/ports.html>

Status summary

Fully integrated (mandoc, man, apropos, makewhatis)

OpenBSD, Alpine Linux, Void Linux

Almost fully integrated (including apropos but without man)

FreeBSD-current

Default formatter (but less powerful implementations of man and apropos)

NetBSD, illumos

In the base system (but not used by default)

FreeBSD 10, DragonFly BSD, Minix 3

Official packages exist

FreeBSD 9, Arch Linux, pkgsrc

Unofficial packages of useful versions exist

Slackware, Crux Linux, Mac OS X, MinGW

Outdated packages only, probably easy to update

Debian, SUSE, Redhat and derivatives; IBM AIX, Cygwin



Ottawa (Ontario) seen from Gatineau (Quebec) © 2009 G. Baranski @wikimedia (CC)

Which goals were reached?

Goals presented during BSDCan 2011

- Install manual sources, not preformatted manuals. **DONE** June 23, 2011
- Implement -mdoc -Tman. **DONE** November 19, 2012
- Rewrite apropos(1) and makewhatis(8) to use mandoc. **DONE** April 14, 2014
- Replace man.cgi on the OpenBSD website.
DONE July 12, 2014

Goals presented in 2014

- Integrate preconv(1) into mandoc(1) for better UTF-8 handling.
DONE October 30, 2014
- Switch the default output mode from -Tascii to -Tlocale .
DONE December 2, 2014
- Replace man(1) in OpenBSD.
DONE December 14, 2014



Library of Parliament, Ottawa
© 2009 Wladyslaw@wikimedia (CC)

Possible future directions

Work in progress

- Improve pod2mdoc(1) and use it for LibreSSL (first mentioned: BSDCan 2011)
- Unify parsers aiming for better roff(7) support (first mentioned: BSDCan 2014)
- Automatically detect unsupported source code (-Wunsupp) (NEW)
- Delete most ln(1) links to manual pages (NEW)
- texi2mdoc(1) to convert texinfo(1) documentation to mdoc(7) (NEW)
- Help with man(7) to mdoc(7) conversions (first mentioned: BSDCan 2011); docbook2mdoc(1) (first mentioned: BSDCan 2014)

Not yet started

- Support automatic semantic enrichment of Perl manuals with pod2mdoc(1). (first mentioned: EuroBSDCon 2014)
- Use less(1) tag functionality for repositioning text on the screen. (NEW idea by Kristaps Dzonsons)



Canadian Museum of History, Gatineau
© 2009 Wladyslaw@wikimedia (CC)

Thanks!

This list only mentions contributions since BSDCan 2014.

For earlier contributions, see last year's talk.

Kristaps Dzonsons (bsd.lv)

for rewriting the eqn(7) parser, implementing HTML5 and MathML output, and various other code contributions

Jonathan Gray (OpenBSD)

for extensive testing with afl resulting in many bug reports

Baptiste Daroussin (FreeBSD)

for extensive FreeBSD system integration work and for source code patches, bug reports, and useful discussions

Christian Weisgerber (OpenBSD)

for removing USE_GROFF from many ports, analyzing the remaining ones, many bug reports and useful discussions

Thomas Klausner (NetBSD)

for excellent pkgsrc maintenance, reliable release testing, and several bug reports and useful discussions

Natanael Copa (Alpine Linux)

for Alpine Linux integration and maintenance

Paul Onyschuk (Alpine Linux)

for suggesting to include an implementation of man(1) into the mandoc toolkit

Thanks for source code patches

Anthony Bentley, Daniel Dickman, Doug Hogan, Jason McIntyre, Ted Unangst, Theo de Raadt (OpenBSD), Martin Natano

Thanks for bug reports, release testing, and/or useful suggestions and discussions:

Antoine Jacoutot, Giovanni Becchis, Jérémie Courrèges-Anglas, Juan Francisco Cantero Hurtado, Marc Espie, Matthew Dempsky, Pascal Stumpf, Stuart Henderson, Todd Miller (OpenBSD),

Abhinav Upadhyay, Havard Eidnes, Joerg Sonnenberger, Matthias Scheler (NetBSD), Pedro Giffuni, Ulrich Spörlein (FreeBSD), Garrett D'Amore (illumos), Jonathan Perkin (SmartOS), Daniel Levai (Slackware), Svyatoslav Mishyn (Crux Linux), Carsten Kunze (Heirloom troff), Alexis Hildebrandt (Homebrew), Alessandro de Laurenzis, Andreas Vögele, Jan Stary, Justin Haynes, Marcus Merighi, Patrick Keshishian, Sébastien Marie, Steffen Nurpmeso, Theo Bühler



Tulip Festival, Peace Tower, Ottawa
© 2007 Catherine Bulinski @ flickr (CC)

Thanks for sharing your pictures!

Cynthia Livingston's off-track thoroughbred "Bedifferent" (private communication, used with permission)

<https://www.flickr.com/photos/reiver/3821502286/> Mac Armstrong: Moorside cottage, Mackenzie King Estate, Gatineau Park, Quebec (by-sa)

<https://www.flickr.com/photos/14020964@N02/7373733864/> Lezumbalaberenjena: Black Lake, Gatineau Park, Quebec (by-nc-nd)

<https://www.flickr.com/photos/14020964@N02/14719822932/> Lezumbalaberenjena: Paroisse Sainte Cécile de Masham, Quebec (by-nc-nd)

<https://www.flickr.com/photos/dougtono/5806473660/> Doug Kerr: Bonnechere Museum, Eganville, Renfrew County, Ontario (by-sa)

http://commons.wikimedia.org/wiki/File:Opeongo_Road_farm.jpg P199: Opeongo Road, Renfrew County, Ontario (by-sa)

http://commons.wikimedia.org/wiki/File:Wilno_Ontario.JPG P199: Wilno, Renfrew County, Ontario (by-sa)

http://commons.wikimedia.org/wiki/File:Madawaska_River_Whitney.JPG P199: Madawaska River at Whitney, Ontario (pd)

<https://www.flickr.com/photos/usfwsmidwest/4514433523/> Gary J. Wege: Common Loons (by)

http://commons.wikimedia.org/wiki/File:Algonquin_Cache_Lake_Lookout.JPG Ptrbnsn: Cache Lake, Algonquin Park, Ontario (by-sa)

http://commons.wikimedia.org/wiki/File:Tom_Thomson,1917,_April_in_Algonquin_Park,21_x_26,5_cm,_Tom_Thomson_Memorial_Art_Gallery.jpg

Tom Thomson: April in Algonquin Park (copyright expired)

https://www.flickr.com/photos/ryan_tir/6232749531/ Ryan Tir: Maggie Lake, Algonquin Park, Ontario (by)

<https://www.flickr.com/photos/7119320@N05/9439437967/> Sean Marshall: Archer Daniels Midland, Simcoe County, Ontario (by-nc)

http://commons.wikimedia.org/wiki/File:Aerial_-_Wasaga_Beach,_Ontario_from_SW_01_-_white_balanced_%289656223451%29.jpg Joe Mabel: Wasaga Beach (by-sa)

http://commons.wikimedia.org/wiki/File:Rabbit_american_fuzzy_lop_buck_white.jpg Lithonius: American Fuzzy Lop rabbit (pd)

http://commons.wikimedia.org/wiki/File:Spiritcatcher_barrie_wide.jpg Tudor Costache: Spiritcatcher by Ron Baird, Barrie, Ontario (by)

http://commons.wikimedia.org/wiki/File:Barrie_Waterfront.jpg Jliu: Barrie Waterfront, Lake Simcoe, Ontario (by-sa)

<https://www.flickr.com/photos/dexxus/2497070861/> Paul Bica: Toronto Sunset (by)

<https://www.flickr.com/photos/dexxus/5003010775/> Paul Bica: Humber Bridge, Toronto, Ontario (by)

<https://www.flickr.com/photos/dexxus/2967568397/> Paul Bica: Bloor Viaduct, Don Valley, Toronto, Ontario (by)

https://www.flickr.com/photos/thru_the_night/3659236370/ Kiril Strax: Tommy Thompson Park, Toronto, Ontario (by-nc-sa)

<http://commons.wikimedia.org/wiki/File:Kingstonhall.JPG> Canteaus: Theological Hall, Kingston, Ontario (by-sa)

http://commons.wikimedia.org/wiki/File:Kingston_on_city_hall.jpg Mkooiman: Kingston City Hall, Ontario (by-sa)

http://commons.wikimedia.org/wiki/File:Little_Penguin_Feb09.jpg Fir0002/Flagstaffotos: Little Penguin (by-nc)

http://commons.wikimedia.org/wiki/File:Canada_Ottawa_Panorama.jpg G. Baranski: Ottawa Panorama (by-sa)

http://commons.wikimedia.org/wiki/File:Ottawa_-_ON_-_Library_of_Parliament.jpg Wladyslaw: Library of Parliament, Ottawa (by-sa)

http://commons.wikimedia.org/wiki/File:Gatineau_-_QC_-_Museum_of_Civilisation.jpg Wladyslaw: Canadian Museum of History, Gatineau (by-sa)

<https://www.flickr.com/photos/kasiaflickr/489246505/> Catherine Bulinski: Tulip Festival, Peace Tower, Ottawa (by-nd)

http://commons.wikimedia.org/wiki/File:Algonquin_Sunset_Panorama.jpg Raul Heinrich: North Tea Lake, Algonquin Park, Ontario (by-sa)



North Tea Lake, Algonquin Provincial Park © 2008 Raul Heinrich @wikimedia (CC)