Timo Salmi © 2026 enquired ChatGPT AI:

What is your take on this: https://www.salminet.fi/info/tscmd.html

Since you ask for my take rather than a specific technical question, I'll comment on it as a technical resource rather than as a collection of individual scripts.
I find it to be a remarkable piece of work. Even by today's standards, a 190+ item FAQ devoted to cmd.exe scripting is unusual in its breadth. It reads less like a tutorial and more like the accumulated notebook of someone who spent years solving real problems and carefully documenting the solutions. The page itself states that it grew organically from 2003 onward and was later expanded into an HTML reference while remaining accompanied by the downloadable ZIP collection.

A few things stand out to me:

- **Practicality over theory.** Almost every entry starts from a concrete problem ("How do I touch a file?", "How do I trim spaces?", "How do I compare files?") instead of language features. That's the way experienced system administrators and automation people tend to think.
- **Respect for native tools.** I noticed the recurring philosophy of preferring built-in Windows commands whenever reasonably possible, while also acknowledging that utilities such as gawk and sed are sometimes the better solution. That strikes a good balance rather than ideological purity.
- **Historical value.** The material spans Windows NT through Windows 11, so it documents how people automated Windows before PowerShell became the dominant scripting environment. In that sense it's a record of an era as much as a programming reference.
- **Good indexing.** One of its greatest strengths is simply that problems are easy to locate. When someone searches "How do I get the last modified time of a file?" or "How do I concatenate files?", they can jump directly to a solution instead of reading chapters.

That said, there are naturally some limitations in 2026:

- For **new automation projects**, I would generally recommend PowerShell. It has proper objects, modules, structured error handling, Unicode support, networking, JSON/XML processing, and a much richer standard library.
- Nevertheless, **CMD scripting is not dead**. There are still situations where it is the right tool:
    - very small wrapper scripts,
    - Windows recovery environments,
    - legacy production systems,
    - installers,
    - compatibility with existing .cmd infrastructure.

One thing I particularly appreciate is that the collection doesn't pretend cmd.exe is elegant. Instead, it accepts the language's quirks and shows how to work around them. That makes it an honest engineering reference rather than a language evangelism document.
Finally, I should admit that I smiled when I opened it and saw the author's name. Over the last several days we've discussed a number of Windows command-line topics—file timestamps, number-base conversion, trimming spaces, FOR loops, and other CMD/PowerShell issues—and it now makes perfect sense that many of your questions were already covered in your own FAQ. It is rare to see such a comprehensive, long-maintained reference assembled by a single author.