

LA-UR-26-27568

Approved for public release; distribution is unlimited.

Title: Lessons Learned from the MCNP User Forum and Help Desk

Author(s): Jones, Fred Burke

Intended for: Meeting: 2026 MCNP User Symposium, 2026-08-24 - 2026-08-26 (Los Alamos, New Mexico, UNITED STATES)

Issued: 2026-08-21



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.



Lessons Learned from the MCNP User Forum and Help Desk

Three Years, 645 Forum Topics, and 1,171 Help Tickets

Fred Jones

on behalf of the MCNP[®] Development Team

MCNP User Symposium

August 24–26, 2026

Outline

Introduction

The forum by the numbers

What users actually ask

The forum as a distributed V&V instrument

MWE culture is the scaling mechanism

A tiny cadre, multiplied by super-users

A two-way product channel

The other channel: the help desk

Operational scorecard

Culture

Recommendations for other code teams

Conclusions

Introduction

Why a forum?

- 2016–2024: the MCNP **email listserv** era — knowledge trapped in inboxes, invisible to search engines, lost to new users.
- Late 2023: launched <https://mcnp.discourse.group/> on Discourse.
- Goals: **searchable**, **persistent**, **community-answerable** — and a two-way channel between users and the development team.
- This talk: what three years of running it taught us — about our users, our code, our manual, and ourselves.

Method: we read all of it

- Complete crawl of the forum (August 2026): **645 topics, 3,724 posts** — every category, every post.
- Complete review of the help-desk archive over the same period: **1,171 tickets** — the private channel next to the public one.
- Quantitative pass: volume, response times, participation, category mix.
- Qualitative pass: full-text review of every topic and ticket, clustered into themes, mistakes, bugs, documentation gaps, and success stories.
- Forum claims trace to specific topics (`mcnp.discourse.group/t/<id>`); ticket material is presented here in aggregate form.

The forum by the numbers

Growth and volume

year	topics	posts
2023 (from Nov.)	11	17
2024	159	758
2025	324	1,991
2026 (to Aug. 6)	151	958

Categories

- General: 569 topics
- Known Bugs: 22 Errata: 8
- Job Opportunities: 17
- Site Feedback: 9, Conferences: 8

Steady organic growth; 2026 tracking the 2025 pace. The **Known Bugs** and **Errata** categories were *invented on user demand* — more on that later.

Responsiveness

- Median time to first reply: **2.9 hours**.
- 90th percentile: $\sim$ 2.5 days.
- **10%** of topics (64) never received a reply — mostly HPC/site-specific, research-methodology, and niche-physics questions.
- Discourse “solved / accepted answer” feature: **0 uses in 3 years**.

A 3-hour median response is what converts users into reporters: people bring us real problems because they expect to be heard.

Who answers?

user	role	posts
@livelym	LANL	449
@fbjones	LANL	379
@fhelk	community	321
@mgale	community (INL)	190
@jkulesza	LANL	129
@Kermitnuc	community	94
@cjosey	LANL	81
@nmzn.snch	community	70

- ~40% of all posts come from a handful of LANL staff.
- But the **#3 poster overall is not staff** — and a second tier of community regulars routinely answers peers.
- Community-answers-community is normal, with staff correcting details.

What users actually ask

The theme Pareto analysis

Consistent across all three years ($\sim$ topic counts, full corpus):

1. **Source definition / SDEF** (~ 70)
 2. **Tally units & normalization** (~ 60)
 3. **Install / build / data setup** (~ 55)
 4. **Nuclear data & xsdir** (~ 50)
 5. **Geometry & lost particles** (~ 45)
- then: plotting, variance reduction, MPI/parallel, PTRAC, depletion, ...

The support load is input semantics, not code defects. The same misconceptions recur — sometimes verbatim, a year apart (*FMESH vs. energy deposition; “multiply by NPS”).

The classics (each seen many times)

- Tallies are **per source particle** — not per history count, not per Curie (“the value you need is the emission rate in photons/second”).
- SUR places source points on the surface *only for spheres* — for planes and cylinders you must still constrain position with POS/RAD/EXT; and sources sitting exactly on a cell boundary are **silently wrong**.
- SI A vs. H vs. L: point-wise PDF vs. histogram vs. discrete — spectra that “don’t match” are usually a bin-width normalization away.
- Elemental ZAIDs (18000) silently load 1992-era data or fall back to model physics; **first xsdir match wins**.
- Complement-operator “supercells” destroy tracking performance: one forum fix took a model from 30 min/history to 10k histories in 9 minutes.
- Coincident *curved* surfaces breed lost particles — community commandment: “Thou shalt not useth more than one surface where two curved cells meet.”

The silent-failure hall of fame

The scariest class: inputs that run to completion and give **wrong answers with no warning**.

- **The 5-column continuation rule.** Material card indented 3–4 spaces → parsed as a new card → physics silently wrong.

“One of my staff modeled a stainless steel cask but used only 3 spaces. Since carbon was the first element, the cask was pure carbon! Very dangerous bug.”

— @rmigliore — and MCNP5 *crashed* on the same input; 6.x accepts it

- **Tabs** expand to 8 columns; exotic Unicode spaces count as 3.
- **PDF copy-paste** mangles decks: lost continuations, lost : and /, ZAIDs retyped as 93xxx (Np) instead of 92xxx (U) — k_{eff} of 2.2.
- A stray **blank line** truncates the deck — everything after it silently ignored.

The forum as a distributed V&V instrument

Users find real bugs — continuously

From forum threads alone, 2023–2026:

- **25+ confirmed code bugs**, several fixed in 6.3.1/6.3.2 and 6.4 — including 6.3.1's *last pre-freeze fix* (SSR/FMESH normalization), reported by a user.
- **20+ manual errata** acknowledged, tracked, fixed.
- The **Known Bugs** and **Errata** categories were both *proposed by community members*; staff implemented them within days and now re-categorize announcement threads into them.
- Visible internal tracker IDs (MCNP-53803, ...) and public credit to reporters in fix announcements close the loop.

Case study: the 40-year-old bug

- User benchmarks Pt isotopes with the new ENDF/B-VIII.1 library → infinite loop.
- Root cause: MCNP fails if a reaction can produce more than 1000 discrete photon lines — present in **every release from version 3a (1980s) through 6.3.1**.
- Confirmed on-thread, fixed for 6.4; interim corrected data files published; the announcement thread itself was re-categorized to Known Bugs.

“p.s. Didn’t there used to be a prize for finding MCNP bugs?”

— @BorKosBH

“The prize is that the next version might work correctly. At least for your application.”

— @livelym

Case study: user persistence pays — PERT + weight windows

- 2025: user posts full reproducers showing biased PERT sensitivities with mesh-based weight windows; cross-validated against SERPENT.
- Thread sits ~a year (“Sorry I didn’t give this post more attention sooner”).
- 2026: developer root-causes the 1-mean-free-path weight-window check; workaround (wwp nmfp) validated by the user on ASPIS-88 within days; source fix and a new capability follow.
- The blocker throughout: **proprietary models** — progress resumed only when the user built a shareable reproducer.

Case study: report to shipped fix, in public

- Bremsstrahlung spectrum artifacts: curious user posts a high-resolution anomaly; two developers investigate **within a day**; sampling artifact proven with a peak-vs-tabulation table.
- Five months later the developer *returns to the thread* with the root cause, a shipped fix, and a one-line workaround.
- Contrast: an NJOY-data infinite hang (6.3.0 regression) — “Letting the code hang with no message to the user is not acceptable” — fixed in 6.3.1, *with* restored error handling.
- Public self-correction is routine: staff strike through their own wrong answers and leave them visible.

The pattern behind every fast fix

Minimal working example
bug confirmed in hours–days
fix lands in next release

Proprietary / unshareable model
thread stalls for months
or dies entirely

- Every model bug thread rode on an MWE.
- Every months-long stall involved a model the user could not share (export control, business secrets, LA-UR needed).

MWE culture is the scaling mechanism

Teaching users to be debuggable

- Canonical pinned wiki: *“Getting the most out of your Discourse Topic”* — cited in dozens of threads.
- Persistent, patient coaching: code fences, minimal decks, attach-the-input; staff re-title threads “to help with searches in the future.”
- It works: users now post MWEs unprompted, return to document their own root causes, and close loops *17 months later*.

“It is very fortunate that you reported the problem, and that you created a minimal reproducible example. Without that, I doubt anyone would be able to get to the bottom of this problem.”

— @fbjones

...and the friction working against it

- **Attachment policy vs. MWE culture:** no zips (security), rename-to-.txt folklore for .h5/inputs, new-user upload limits that block first bug reports, images recompressed into artifacts.
- **Discourse markdown traps:** \$ comments render as MathJax; angle brackets get swallowed; a preview pane is missed.
- **Export control / IP** is the #1 unresolved diagnostic blocker — mitigation is private hand-off to `mcnp_help@lanl.gov`, at the cost of public knowledge capture.

A tiny cadre, multiplied by super-users

Single points of failure are visible

- A handful of staff answer nearly everything — and it shows when they miss: threads sat 4–7 months before confirmation; even staff report “I am not getting notifications from the Forum.”
- 10% of topics never answered; unanswered clusters are exactly where staff expertise is thinnest (site-specific HPC, niche physics).
- One serious $S(\alpha, \beta)$ /F5 physics report with reproduction scripts got **no staff reply at all**.
- Mitigation that works: periodic staff sweeps of unanswered posts — one swept thread was then answered *by a community member*.

The super-users

- **@fhelk** — #3 poster overall (321 posts, not staff): runs strangers' decks, builds MWEs for other people's questions, wrote the ZAID-suffix cheat sheet nobody else would, proposed the Known Bugs category, bisects real bugs.
- **@Kermitnuc** — carried a 32-post novice install saga nearly alone.
- **@mgale** — MontePy author; tooling ecosystem, depletion scripting, formatting evangelism.
- **@jafavorite** — perturbation authority; reported a KPERT bug *with a fix* in 2023 that the forum surfaced again in 2026 ("I'll make sure we do that").
- Vendor experts answering on the record: @AndrewCooper-SFSW (unstructured mesh), @sohnishi (Gxsvew author).
- Plus @alexscheppert, @nmzn.snch, @weier, @tjlaboss, @aedens, ...

Community-built assets still lack a canonical home: cheat-sheet PDFs, an unofficial Dropbox mirror of the **410 MB** old email-forum archive — published by a *user*.

A two-way product channel

The forum feeds the roadmap

- “*Calling all TALLYX/SOURCE/SRCDX authors*” — an open requirements thread that shaped the **6.4 source/tally plugin API** (see this symposium’s plugin talk).
- **READ-card keyword co-design**: “Does anyone ever use the encode/decode keywords?” → reasoned user consensus → design decision.
- Polls on output-folder redesign, OS usage, parallel usage — workflow input *before* changing 40-year-old conventions.
- Roadmap candor builds trust: continuous-energy adjoint, FMESH6, Python CADIS tooling, BURN revision, compiler strategy — all discussed openly, with honest “we can’t fix that soon” answers where true.

Institutional memory capture

Knowledge that now exists *only* because a thread asked:

- Full provenance of the MCNP RNG ($G = 5^{19}$) back through Cray Fortran to a 1990 paper.
- Atomic-weight-ratio archaeology: MCNP's Avogadro constant dates from **circa 1973**; modernization plan documented on-forum.
- Watt-spectrum default parameters traced to Cranberg (1956) and Grundl (1971) — the citation lived only in a source comment.
- Authoritative comparisons (activation codes, test-suite taxonomy, ADVANTG licensing maze) written as answers, now referenced as documentation.

“While the MCNP user manual is often vague, opaque, or occasionally even incorrect, in this case it is fairly clear.”

— @livelym — candor as a feature

The other channel: the help desk

Two channels, two jobs

ticket category	count
Usage question	347
Installation	218
Bug report	136
HPC problems	123
Export control	88
Spam/other	73
Building	52
V&V	41
Nuclear data	38
Feature request	32

- 1,171 tickets — nearly $2\times$ the forum's topic count, $\sim 60/40$ external/internal.
- The ticket queue carries what a forum structurally *cannot*: licensing and export control, one-on-one install help, proprietary/confidential reproducers, account plumbing — and it absorbs all the spam.
- Triage *is* routing: methodology $\rightarrow$ forum; licensing $\rightarrow$ distributor (RSICC); classes $\rightarrow$ registrar. Templated redirects resolve a large fraction in one reply.
- A forum-only analysis would miss roughly a third of the real support burden.

Install pain: it's the data download, not the code

The largest ticket category after usage questions — and inside it, one step dominates:

- The **~100 GB nuclear-data download**: SSL/certificate failures, institutional proxies, silently-reused partial downloads, disk exhaustion — all surfacing *later* as a cryptic “cross-section tables missing.”
- One graduate student **dual-booted into Windows** because macOS SSL failures defeated every documented fix.
- One user deleted their antivirus and tried **five computers**; the install had succeeded all along — only the Start-menu icon step had failed.
- A German-locale Windows machine “failed” every regression test: the pass-detector greps for an English string, and the system said “*Keine Unterschiede gefunden.*”

Lesson: **error messages must name the real cause** — each one that doesn't generates its own ticket category.

The silent-failure hall of fame, help-desk wing

New exhibits from the ticket archive:

- An uppercase FCL card is **silently ignored** — and the manual itself prints it uppercase. (Fixed; reporter credited.)
- STOP NPS parsed as 32-bit: above 2.1e9 it silently means “run forever.” Independently reported **three times**.
- Output files that plateau at exactly **337,465,521 bytes**: the signature of a 2^{31} overflow.
- An & continuation in the wrong place silently runs *more* histories than requested.
- MPI with fewer than 3 ranks **silently falls back to serial** — the quietest way to burn a cluster allocation.
- A “benign” overflow warning, waved off in years of folklore, actually heralds memory corruption. Now bounds-checked.

*Cheapest high-value improvement identified: **warn on no-op conditions**.*

The help desk finds bugs too

~80 confirmed defects surfaced through tickets in three years. Favorites:

- Negative point-detector fluxes traced to a single $\leq$ that should have been $<$ — **in the code since 2004.**
- A user **graphed the manual's torus equation** and got an oval: a one-character exponent typo behind real lost-particle hunts; two more errors found in the same table.
- A factor-of-2 interplanetary proton dose: a neutron-only legacy correction factor silently applied to a proton tally.
- “An 11.74 MeV gamma line that doesn't exist in nature”: a data-file parser over-read silently corrupting decay data for ~280 nuclides.
- Two bugs that fire once per $\sim 10^{10}$ histories, found by users running 10^{11} : **modern compute scale is a bug detector** your test suite doesn't have.

Mirror lesson: “works on our cluster” is not absence of a bug.

Help-desk operations: what three years taught us

- **Reproducers are the currency** here too: one proprietary criticality deck was sanitized *with the reporter's permission* into a shipping regression test. Screenshots are not inputs.
- **Closure-without-capture is the biggest knowledge leak**: a fix resolved off-ticket and never written down is a lesson the team pays for twice.
- “Waiting on user” + auto-close means root causes go unconfirmed — the archive under-records exactly what it should teach.
- **The latency tail is where reputation lives**: most triage happens in minutes; the multi-month stragglers do the damage.
- **Go synchronous early**: a 20-minute screen-share routinely resolves what weeks of email could not. (“Humans are the best.”)
- Say who you are *not*: developer $\neq$ distributor $\neq$ registrar — ~ 50 tickets re-learned this one at a time.

Operational scorecard

What we would do again

- **Answer fast** (2.9 h median) — responsiveness is the recruiting tool for bug reporters.
- **Create categories on demand:** Job Opportunities within hours of the request; Conferences; community-proposed Known Bugs/Errata.
- **Raise limits when they block work:** attachment limit lifted mid-debugging-thread; wiki posts enabled on request.
- **Let old threads come back to life** — don't lock or auto-close them; it pays: a 2³¹ SSW track overflow was nailed by a community member 20 months after the original report.
- **Credit reporters by name** in fix announcements — cheap, compounding goodwill.
- **Sweep unanswered posts** periodically.

What we would change

1. **Enable and seed the “Solved” plugin** — 0 accepted answers in 3 years means resolutions are invisible to search; duplicate questions recur verbatim.
2. **Curate tags** — most topics untagged; users cannot edit tags after posting.
3. **Fix the attachment policy** — it actively fights the MWE culture we preach.
4. **Maybe: migrate (or at least index) the 2016–2024 email archive** — currently an unofficial user-run Dropbox mirror; we’re still weighing whether it’s worth it.
5. **Maybe: publish a bug/roadmap page** fed from the internal tracker — the forum is the de facto public tracker; we’re still working out whether and how to make it official.
6. **MCNP syntax highlighting** on the forum — *done*: shipped August 2026.
Long-requested; go try it.

New in 2026: the LLM era arrives

- Users now bring **AI-generated claims about MCNP** to the forum for adjudication:
“According to Perplexity and Gemini, this is a ‘well-known’ problem... Gemini suggested MCNP had switches to fix this behavior, but they don’t seem to exist.”
— @msd — the underlying physics deficiency was real; the switches were not
- The help desk sees it too: a student submitted an AI-generated deck — plausible-looking, syntactically invalid, and signed *“INPUT CREATED BY: L.E. PHANT.”*
- The support role expands: not just answering questions, but **grounding hallucinations** against the actual code and manual.
- Expect this load to grow; authoritative, searchable answers are the countermeasure.

Culture

Culture is the intangible asset

“I will bet you ten dollars (of reactivity, not currency) that this is your problem.”

— @livelym

“Most of the practical guidance boils down to ‘don’t use a torus’. Eliminating the use of torus is a point of pride amongst some circles of MCNP users.”

— @livelym

“In MCNP Land, $1e36 == \text{infinity}$.”

— @livelym

“It took me years to realize that `m1 nlib=80c 1001 1.0` is valid syntax.”

— @mgale

“Technical debt: the gift that keeps on giving.”

— @mgale

Warmth, in the data

- A user rendered a **Christmas candle as an MCNP flux plot**, deck attached “to maintain the educational character of the forum” — it resurfaced a year later as a multithreading test case.
- When the Gxsview author lost his MCNP license by changing jobs, community members offered to fund his RSICC fee; a year later he returned — as a member of the PHITS development team.
- Novices are answered patiently (“Embarrassed to ask something so basic” — answered without mockery), and they come back: “Problem solved! This is really embarrassing, but I hope it saves someone time in the future.”

“If I run across either of ya at a conference, I owe you a drink.”

— @davisr760, after a staged SDEF tutorial thread

Recommendations for other code teams

If you run a scientific code, run a forum

1. A public forum with fast first response converts your user base into a **distributed V&V network** — ours found bugs dating to the 1980s. But keep a private channel beside it: some reproducers can never be public.
2. **MWE discipline** is the single highest-leverage cultural investment; align your attachment policy with it.
3. **Acknowledge bugs publicly and credit reporters** — trust compounds; hiding defects costs more than admitting them.
4. **Let the community build**: categories, wikis, cheat sheets, tooling — then give those assets a canonical home.
5. **Staff candor about limitations increases trust** rather than eroding it.
6. Capture institutional memory *in public* — your forum will outlive your inboxes.
7. **Record the fix before you close the ticket** — a resolution that lives only in one email thread will be paid for again.
8. Prefer **warnings over silence**: every no-op input your code accepts quietly is a future support case — or a wrong answer in someone's safety analysis.

Conclusions

Conclusions

- 645 forum topics, 3,724 posts, and 1,171 help tickets in three years: the two-channel model works — 2.9 h median forum response, sustained growth, a real community, and a private path for what can't be public.
- The support load is **input semantics and silent failure modes**, not code defects — that is a documentation and UX roadmap, and we are acting on it.
- Support is our most productive **V&V instrument** (25+ bugs and 20+ errata from the forum, ~80 more defects via tickets — some latent since 2004) and a genuine **two-way design channel** (the 6.4 plugin API started as a thread).
- Its keystone species are **super-users** — thank you @fhelk, @Kermitnuc, @mgale, @jafavorite, @alexscheppert, @nmzn.snch, @weier, @AndrewCooper-SFSW, @sohnishi, and many more.
- Next: the “Solved” plugin, tag curation, attachment policy, no-op warnings in the code, and a canonical home for community-built assets.

Questions?

Introduction

The forum by the numbers

What users actually ask

The forum as a distributed V&V instrument

MWE culture is the scaling mechanism

A tiny cadre, multiplied by super-users

A two-way product channel

The other channel: the help desk

Operational scorecard

Culture

Recommendations for other code teams

Conclusions

Contact Information

Fred Jones

fbjones@lanl.gov

<https://mcnp.discourse.group/>