

`spmerge`: A Canonicalization Utility for Spherepop OS

Flyxion

December 13, 2025

1 Purpose

`spmerge` is a user-facing utility that proposes semantic equivalence between objects in Spherepop OS by generating `MERGE` or `COLLAPSE` events.

Its sole responsibility is to *express equivalence explicitly*. It does not infer, guess, or commit semantics on behalf of the kernel.

2 Design Position

`spmerge` occupies a narrow position in the system:

- It is not a kernel component.
- It does not mutate authoritative state.
- It does not perform semantic inference.
- It operates entirely through event proposals.

The kernel remains the sole interpreter of equivalence.

3 Input Model

`spmerge` accepts a set of object identifiers $\{o_1, o_2, \dots, o_n\}$ supplied explicitly by the user.

These identifiers must already exist in the replayed kernel state. No implicit creation of objects is permitted.

4 Operational Modes

4.1 Pairwise Merge

In the simplest mode, `spmerge` proposes a single equivalence:

`MERGE(o_a, o_b)`

This induces equivalence between two objects. Representative selection is deferred entirely to the kernel.

4.2 Batch Merge

For a set of objects $\{o_1, \dots, o_n\}$, `spmerge` generates a sequence of MERGE events forming a spanning tree over the set.

The exact ordering of merges is deterministic and stable, but semantically irrelevant due to merge confluence.

4.3 Region Collapse

In region mode, `spmerge` proposes a single:

`COLLAPSE(S, or)`

where S is a finite set of objects and $o_r \in S$ is the chosen representative.

Region collapse is explicit and irreversible once committed.

5 Determinism Guarantees

Given:

- a fixed event log prefix
- a fixed invocation of `spmerge`

the utility must generate an identical proposal stream bit-for-bit.

No randomness, timestamps, or environment-dependent behavior may influence output.

6 Replay Discipline

`spmerge` operates only on replayed kernel state. It does not consult external databases, caches, or metadata channels when deciding which events to propose.

All decisions are traceable to:

- user-supplied identifiers
- the replayed equivalence relation

7 Preview Workflow

By default, `spmerge` operates in preview mode:

1. A speculative overlay is constructed.
2. Proposed merge events are applied to the overlay.
3. The resulting derived state is displayed.

No authoritative events are committed unless the user explicitly submits the proposal.

8 View Generation

`spmerge` may emit derived views including:

- before/after equivalence classes
- rewritten relations
- representative changes

These views are observational only and have no semantic force.

9 Error Conditions

`spmerge` must fail fast under the following conditions:

- an object identifier does not exist
- a proposed merge is redundant
- the target set is empty

Silent no-ops are forbidden.

10 Explicit Non-Goals

`spmerge` does *not*:

- infer equivalence from structure
- auto-merge based on heuristics
- reorder or optimize the event log
- collapse objects without explicit instruction

These behaviors are incompatible with semantic clarity.

11 Relationship to Kernel Invariants

Every invariant enforced by `spmerge` derives directly from kernel invariants:

- equivalence is event-induced
- merge confluence is guaranteed by the kernel
- representative normalization is kernel-owned

`spmerge` merely expresses intent.

12 Future Extensions

Possible future extensions include:

- interactive merge previews
- structured justification annotations (as metadata)
- batch refactoring workflows

All extensions must preserve replay determinism and non-interference.

13 Conclusion

`spmerge` is intentionally conservative. Its value lies not in automation, but in making semantic equivalence explicit, inspectable, and replayable.

By refusing to infer meaning, `spmerge` preserves the integrity of Spherepop OS as a system of semantic time.