Plan per-event inventory history before raising record limits

InventoryRollbackPlus limits saved records by event type and player, rather than promising a fixed number of days. A player who changes worlds frequently can cycle through world-change records faster than someone who rarely joins. Plan the useful history and its storage cost before raising every limit.

Record the starting point

Preserve the current configuration and a verified copy of the existing history. Record whether this installation stores YAML files or uses MySQL, along with available capacity at that destination. Increasing game-panel disk allocation does not necessarily enlarge a separate database server.

Work through the change

1. Open the installed `config.yml` and compare its `max-saves` section with the maintainer reference. The documented event keys are `join`, `quit`, `death`, `world-change` and `force`. Record each current count without replacing it with upstream example defaults.
2. Observe a representative player's event frequency and inspect the oldest and newest records of each relevant type. Estimate the history needed from actual activity: a count of ten joins covers a very different period for daily and hourly reconnects.
3. Measure current history storage and the population it represents. Use those observations for a rough capacity budget, allowing for new players and larger inventories rather than treating one player's record size as a universal constant.
4. If capacity permits, increase only the event count that lacks useful history. Apply the setting through the installed release's documented method, then use a consenting online test player and a harmless event to confirm new records remain available.

Check the result

The selected event should have the intended higher ceiling, while other event limits remain unchanged. Record the observation period and recheck storage after representative activity. Raising a limit retains more future records; it cannot recreate older records already rotated out.

If the result differs

If no new records appear, investigate event capture and storage errors separately. If file usage stays flat on a MySQL installation, inspect the actual database destination instead of assuming retention failed. Avoid changing storage backend and record limits during the same comparison.

Pause further increases if capacity is uncertain. Lowering counts can rotate away recovery points as new records arrive, so do not reduce them as an automatic rollback. Preserve the verified history and choose any later retention reduction deliberately; never delete data folders or database tables to make a capacity chart smaller.

References and related help

- InventoryRollbackPlus maintainer reference 1
- InventoryRollbackPlus maintainer reference 2
- Restore a recorded player inventory with InventoryRollbackPlus
- Back up game files before making changes

Revision #1
Created 2026-11-09 16:36:27 UTC by CloudExa Hosting Admin
Updated 2026-11-09 16:36:27 UTC by CloudExa Hosting Admin