Diagnose an InteractiveChat formatter conflict one event at a time

A formatter can replace or cancel a message before its interactive parts reach players. Capture the input, resulting text and plugin ownership before changing event priorities or advanced packet-handling options.

Prepare the test

Use disposable example items and ordinary player accounts for chat tests. Keep private inventory contents, staff commands and personal data away from demonstrations that other players can click or hover.

Make the change

1. Use one ordinary sender and receiver to compare plain text with a single harmless interactive keyword. Record whether formatting, hover content, click actions or the entire message disappears, and whether the console reports cancellation.
2. Identify the chat formatter and moderation plugins involved and note their versions. Preserve the current InteractiveChat configuration and compare the installed `EventPriorities` comments with the upstream reference for the specific event path you are investigating.
3. On a staging copy, change one documented priority only and repeat the identical message. Do not enable advanced Paper chat event switches merely because the server runs Paper; the upstream template explicitly warns against that assumption.
4. Check both public chat and any configured private-message parsing route. Once a supported combination works, apply that narrow change during an agreed window and confirm moderation still cancels messages that it is supposed to reject.

Check the result

The intended interactive content should survive the formatter while ordinary styling and moderation remain effective. Record the accepted priority and plugin versions so a future update can be compared with the known working combination.

If the test fails

Restore the original priority if messages duplicate, bypass a filter or lose their author context. Provide the minimal reproduction to the relevant maintainers rather than disabling every chat listener or increasing packet limits at random.

Keep staging messages free of private inventory data. The purpose is to identify event ownership, so a simple disposable item and a short fixed sentence provide better evidence than a crowded public chat capture.

> CloudExa game panel console and logs view for collecting startup and diagnostic evidence
> Collect the relevant timestamped log text in the panel, then remove private data before sharing it with support.

References and related help

Check the official InteractiveChat documentation against your installed release before using a command or setting from another version.

- Test InteractiveChat item previews without exposing staff inventory
- Find Minecraft logs and send useful troubleshooting details

---

Revision #1
Created 2026-11-09 16:36:25 UTC by CloudExa Hosting Admin
Updated 2026-11-09 16:36:25 UTC by CloudExa Hosting Admin