ChatClear Action Reference

Reference for the `ChatClear` Display action — clears the conversation transcript held for the calling client connection without invoking the LLM.

Purpose

`ChatClear` is a Display Action that wipes the conversation transcript held for the calling client connection on the server side. No LLM call is made. The action is synchronous, returns immediately, and is idempotent — clearing an already-empty transcript is a no-op that still returns `status="ok"`.

Use it to give operators a “Start new conversation” button: one press discards all prior turns so the next `ChatRequest` starts with a blank transcript rather than inheriting context from the previous session.

When to use

- **Shift handover** — when a different operator takes over, clear the prior operator’s chat context before handing off the panel.
- **Topic switch** — when the conversation has drifted off-topic and the operator wants to start fresh without carrying prior assumptions into the next query.
- **Testing and development** — reset the transcript between test runs without restarting the runtime or reloading the Display.
- **Privacy** — prevent the next operator from seeing the previous operator’s questions and replies in the same chat panel.

> The login-change reset built into `ChatRequest` (lazy reset on the next turn after a user change) is distinct from `ChatClear`. `ChatClear` is operator-initiated and immediate; the login-change reset is platform-initiated and lazy. Both result in an empty transcript for the next turn.

How it works

When fired, `ChatClear` resolves the target chat session and removes its transcript from the server-side in-memory transcript cache:

1. The action targets the first `TChatSession` on its own Display. It ignores the **Object** field.
2. The resolved target provides the `(clientGuid, chatName)` key. The server drops the transcript entry for that key.
3. The reply envelope is returned to the **Return** tag (if configured) with `status="ok"`, `text=""`, and empty `toolTrace[]` and `warnings[]` arrays.

Because no LLM call is made, the action typically returns in under a millisecond. It does not appear in tool traces and does not count against the tool-dispatch cap.

Configuration

On a Button (or any clickable control), add an **Action** dynamic with:

Field	Setting
Action type	`ChatClear`
Object	Not used. `ChatClear` ignores this field — leave it empty.
Return	Optional tag that receives the reply envelope JSON. Useful for diagnostics or confirmation UI. May be omitted for a simple “clear and forget” button.
Result 1, Result 2, … (optional)	Tags computed from the reply via Expressions — for example, `@Tag.ClearStatus.JsonString("status")` to surface a confirmation badge.
Query	Not used by `ChatClear`. The action takes no user input.

The Action editor hides fields that do not apply to `ChatClear`. The Query field is hidden; Object, Return, and Result/Expression rows surface.

Reply envelope

The reply envelope follows the same JSON schema as `ChatRequest` and `AI.Execute` — see Local AI Reply Envelope Schema for the full field reference. For `ChatClear` specifically:

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
{
  "text":      "",
  "status":    "ok",
  "toolTrace": [],
  "latencyMs": 0,
  "warnings":  []
}
```

`text` is always an empty string — there is no LLM output. `latencyMs` reflects the wall-clock time for the transcript-drop operation, which is normally sub-millisecond. `toolTrace` and `warnings` are always empty arrays on a successful clear.

Gates

`ChatClear` checks a single gate before executing:

- **`SolutionCapabilities[LocalAI].Enabled` must be `true`** — the master Local AI kill-switch. When disabled, `ChatClear` returns `status="disabled"` without touching the transcript cache.

Unlike `ChatRequest`, `ChatClear` does *not* inspect `ModelOptions` tool-surface bits. Those bits control LLM dispatch; transcript management is independent of the LLM configuration.

Target resolution

`ChatClear` wipes the conversation transcript held for the calling client connection. There is one such transcript per connection, so no target selection is required and none is available.

> The `Object` field is not used by `ChatClear` and is ignored. In releases before 10.1.6, a non-empty value was sent to the server as a conversation key and addressed a conversation no panel displayed — the transcript stayed on screen and the action appeared to do nothing. From 10.1.6 the transcript clears whether or not the field is filled.

Because the transcript is per connection rather than per panel, a `ChatClear` wired to one panel's close event also clears any other chat panel open for that client.

See also

- ChatCompress Action Reference — compresses the transcript to a single summary message via one LLM call, preserving semantic continuity while reducing token cost.
- ChatRequest Action Reference — the primary chat action that sends operator queries to the LLM and manages the transcript.
- ChatSession Control Reference — the Display control that renders the conversation thread.
- Local AI — the parent section, includes a step-by-step quick-start.
- Local AI Reply Envelope Schema — full reply envelope schema.

In this section...