Concept internals

# Markdown Formatting

OpenClaw formats outbound Markdown by converting it into a shared intermediate representation (IR) before rendering channel-specific output. The IR keeps the source text intact while carrying style/link spans so chunking and rendering can stay consistent across channels.

## Goals

**Consistency:** one parse step, multiple renderers.

**Safe chunking:** split text before rendering so inline formatting never breaks across chunks.

**Channel fit:** map the same IR to Slack mrkdwn, Telegram HTML, and Signal style ranges without re-parsing Markdown.

## Pipeline

1. **Parse Markdown → IR**

   IR is plain text plus style spans (bold/italic/strike/code/spoiler) and link spans.

   Offsets are UTF-16 code units so Signal style ranges align with its API.

   Tables are parsed only when a channel opts into table conversion.

2. **Chunk IR (format-first)**

   Chunking happens on the IR text before rendering.

   Inline formatting does not split across chunks; spans are sliced per chunk.

3. **Render per channel**

   **Slack:** mrkdwn tokens (bold/italic/strike/code), links as `<url|label>`.

   **Telegram:** HTML tags (`<b>`, `<i>`, `<s>`, `<code>`, `<pre><code>`, `<a href>`).

   **Signal:** plain text + `text-style` ranges; links become `label (url)` when label differs.

## IR example

Input Markdown:

```
Hello **world** — see [docs](https://docs.openclaw.ai).
```

IR (schematic):

```
{
  "text": "Hello world — see docs.",
  "styles": [{ "start": 6, "end": 11, "style": "bold" }],
  "links": [{ "start": 19, "end": 23, "href": "https://docs.openclaw.ai" }]
}
```

## Where it is used

Slack, Telegram, and Signal outbound adapters render from the IR.

Other channels (WhatsApp, iMessage, MS Teams, Discord) still use plain text or their own formatting rules, with Markdown table conversion applied before chunking when enabled.

›

## Table handling

Markdown tables are not consistently supported across chat clients. Use `markdown.tables` to control conversion per channel (and per account).

- `code` : render tables as code blocks (default for most channels).
- `bullets` : convert each row into bullet points (default for Signal + WhatsApp).
- `off` : disable table parsing and conversion; raw table text passes through.

Config keys:

```
channels:
  discord:
    markdown:
      tables: code
    accounts:
      work:
        markdown:
          tables: off
```

## Chunking rules

Chunk limits come from channel adapters/config and are applied to the IR text.

Code fences are preserved as a single block with a trailing newline so channels render them correctly.

List prefixes and blockquote prefixes are part of the IR text, so chunking does not split mid-prefix.

Inline styles (bold/italic/strike/inline-code/spoiler) are never split across chunks; the renderer reopens styles inside each chunk.

If you need more on chunking behavior across channels, see **Streaming + chunking**.

## Link policy

**Slack:** `[label](url)` → `<url|label>`; bare URLs remain bare. Autolink is disabled during parse to avoid double-linking.

**Telegram:** `[label](url)` → `<a href="url">label</a>` (HTML parse mode).

**Signal:** `[label](url)` → `label (url)` unless label matches the URL.

## Spoilers

Spoiler markers ( `||spoiler||` ) are parsed only for Signal, where they map to SPOILER style ranges. Other channels treat them as plain text.

## How to add or update a channel formatter

1. **Parse once:** use the shared `markdownToIR(...)` helper with channel-appropriate options (autolink, heading style, blockquote prefix).
2. **Render:** implement a renderer with `renderMarkdownWithMarkers(...)` and a style marker map (or Signal style ranges).
3. **Chunk:** call `chunkMarkdownIR(...)` before rendering; render each chunk.
4. **Wire adapter:** update the channel outbound adapter to use the new chunker and renderer.
5. **Test:** add or update format tests and an outbound delivery test if the channel uses chunking.

## Common gotchas

- Slack angle-bracket tokens ( `<@U123>` , `<#C123>` , `<https://...>` ) must be preserved; escape raw HTML safely.
- Telegram HTML requires escaping text outside tags to avoid broken markup.
- Signal style ranges depend on UTF-16 offsets; do not use code point offsets.
- Preserve trailing newlines for fenced code blocks so closing markers land on their own line.

‹ TypeBox Typing Indicators ›