Independent Submission
Request for Comments: 0006
Category: Informational
ISSN: 2070-1721

M. Developer
RFC.EE Project

March 2026

Block Types in the RFC.EE Editor

Abstract

RFC.EE represents every document as an ordered list of typed blocks. This document enumerates the block types that the renderer supports, groups them by role, and summarizes the content schema of each.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of an independent submission. It has been approved for publication by the RFC Editor, but is not a candidate for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc0006.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

Abstract 1
1. Introduction 4
2. Structural Blocks 4
3. Text Blocks 5
4. List Blocks 5
5. Visual Blocks 6
6. Reference and Author Blocks 6
7. Security Considerations 7
8. References 7
9. Author's Address 7

1. Introduction

A block is a fragment of document with a fixed semantic role: a heading, a paragraph, a table, a reference. Each block has a string type and a JSON content envelope whose shape depends on the type. By editing blocks rather than raw text, authors retain the ability to reformat the document without losing structure.

2. Structural Blocks

Structural blocks define the skeleton of the document.

header
: A two-column header block. Content: left_lines[], right_lines[].

title
: A centered title with one or more lines. Content: lines[].

heading
: A numbered section heading. Content: level, number, text, anchor_id.

page_break
: Inserts a page break. Content: page_number.

toc
: Automatic table of contents, generated from the heading blocks.

boilerplate
: Inserts the "Status of This Memo" or "Copyright Notice" text appropriate to the article's category and stream. Content: type (status_of_memo or copyright_notice).

3. Text Blocks

paragraph
: Word-wrapped body text indented three spaces. Content: text.

note
: A labeled paragraph with a custom prefix such as NOTE or WARNING. Content: text, label.

code_block
: A preformatted code block. Content: text (or code) and optional indent.

raw
: An escaped-verbatim text block, for material that should not be wrapped or reflowed. Content: text.

4. List Blocks

bullet_list
: Indented bulleted list. Content: items[], style (default "o").

numbered_list
: Indented numbered list. Content: items[], start (default 1).

definition_list
: Term/definition pairs, with the term at the body indent and the definition indented further. Content: items[{term, definition}].

5. Visual Blocks

ascii_table
: A plain-text table with box-drawing characters. Content: headers[], rows[][], col_widths[] (optional), caption (optional).

packet_diagram
: A 32-bit-wide packet-format diagram with bit ruler. Content: fields or rows, caption.

timeline
: A two-party message-sequence diagram. Content: entities or exchanges with direction labels.

figure
: An arbitrary ASCII-art figure. Content: text, caption, indent.

state_machine
: A preformatted state-machine drawing. Content: text.

abnf
: ABNF rules with aligned equals signs. Content: rules[{name, definition}].

6. Reference and Author Blocks

reference
: A single bibliographic reference. Content: tag, authors, title, series, doi, date, url.

author_address
: A single author address block. Content: name, org, street, city, country_code, email, uri.

7. Security Considerations

Block content is stored as JSON. The renderer applies HTML escaping to all user-supplied fields before emitting HTML, preventing cross-site scripting through document content. Authors SHOULD nevertheless avoid pasting untrusted content into block fields.

8. References

[RFC0005] M. Developer, "About the RFC.EE Editor", RFC 0005, February 2026

9. Author's Address

M. Developer
RFC.EE Project

Email: developer@example.org