Independent Submission
Request for Comments: 0007
Category: Experimental
ISSN: 2070-1721

M. Developer
RFC.EE Project

March 2026

Rendering Pipeline of the RFC.EE Editor

Abstract

RFC.EE produces three output formats from a single block list: HTML, plain text, and PDF. This document describes the pipeline: how blocks become HTML, how the HTML is extracted to a 72-column plain-text file, and how the plain text is reshaped into a printable PDF.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation.

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/rfc0007.

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. From Blocks to HTML 4
3. The Plain-Text Extraction 4
4. The PDF Output 5
5. Pagination 5
6. Security Considerations 6
7. References 6
8. Author's Address 6

1. Introduction

Rendering is the core of the application. The rendering pipeline has three stages, each corresponding to a public endpoint in view.php. Every block in the database produces text at each stage, and the three outputs are designed to be visually faithful to each other.

2. From Blocks to HTML

The function renderArticle() in render.php loads an article and its blocks, then walks the list. For each block it calls a type-specific renderer that emits a chunk of HTML. The chunks are wrapped in a centered and a <pre> element that reproduces the 72-column layout. Page breaks close the current <pre>, insert a horizontal rule and a "New Page" comment, and open a fresh <pre> with the running header.

The result is a self-contained XHTML 1.0 Transitional document that mirrors the plain-text RFC layout line for line. CSS is embedded inline to keep the document portable.

3. The Plain-Text Extraction

When a reader requests &format=txt, view.php runs the HTML renderer and extracts the contents of every <pre> block. Span elements are stripped, HTML entities are decoded, and the resulting text chunks are joined with form-feed characters. The output is served with Content-Type: text/plain; charset=utf-8 and a .txt download name.

Because the HTML is the authoritative form, the extracted text preserves exactly the wrapping, indentation, and page structure that the reader sees in the browser.

4. The PDF Output

When a reader requests &format=pdf, view.php performs the same extraction as for plain text but wraps each page in a <pre> element surrounded by a page-break style. The resulting HTML is handed to dompdf, which rasterizes it onto letter-sized pages using the Courier font at 10 points. The output is served with a .pdf filename.

dompdf is a pure-PHP HTML-to-PDF converter. It is not fast, but it is adequate for documents of a few dozen pages and adds no runtime dependencies beyond PHP itself.

5. Pagination

Page numbers are assigned sequentially from 1. An author may insert page_break blocks manually at natural section boundaries. If no manual breaks exist, the renderer will still produce a correctly paginated document, because the XHTML form uses a single <pre> per logical page.

6. Security Considerations

The rendering pipeline escapes all user-supplied text before it is emitted as HTML. The PDF pipeline further strips <a> elements and all remaining tags before handing the text to dompdf, reducing the attack surface for HTML-to-PDF parser bugs. Operators SHOULD keep dompdf up to date, as the library has a history of CVEs.

7. References

[DOMPDF] dompdf Contributors, "dompdf: HTML to PDF Converter for PHP", 2024 <https://github.com/dompdf/dompdf>.

[RFC7992] Hildebrand, J. and P. Hoffman, "HTML Format for RFCs", RFC 7992, December 2016

8. Author's Address

M. Developer
RFC.EE Project

Email: developer@example.org