

# The text/markdown Media Type

IETF 91 APPSAWG

Honolulu, HI, USA, 2014-11-10

Sean Leonard <dev+ietf@seantek.com>

Penango, Inc.



(Apologies to [Martin Thomson](#))

# Overview (as of Today)

- Family of **related** plain text formatting syntaxes, aka **lightweight** markup language (LML)
- Lots of implementations (with extensions and variations)
  - [Original](#), Ruby Markdown (MD), GitHub Flavored MD (GFM), PEG grammar, MultiMD, PHP Markdown Extra, pandoc, Python-MD, Vim-Flavored-MD, kramdown...
  - Not all designed to be compatible or interchangeable
- [draft-ietf-appsawg-text-markdown-03](#)  
[draft-seantek-text-markdown-use-cases](#)
- Registration of text/markdown media type
- Informational RFC



# Progress

## **draft-01**

- Long intro
- Parameters: charset, rules, processor, processor-ver, processor-args
  - processor-args depended on POSIX
  - Registries of rules, processors

## ***draft-02***

- Use cases explained
- Parameters: charset, flavor, processor [+opt version & args], output-type

## **draft-03 + use-cases**

- Shortened intro; added Markdown editor; shortened spec; ~~processor~~
- Parameters: charset, syntax [+opt version & extensions], output-type
- *use-cases* separate doc: longer intro, strategies for preserving Markdown syntax info, additional syntax registrations, examples

# Points of Controversy ([draft-03](#))

- Still too long
- Still too complicated
- output-type not suited to media type purposes
- IANA Registration Template(s) too complicated
- No agreement on fragment identifiers
- Some agreement but probably not enough to declare consensus on “uses” (Section 1.3)
- Concerns about Unicode (non-ASCII) identifiers

# Proposal (syntax/variant)

- Simplify syntax parameter and change to variant:
- Optional Parameters:
  - variant: An optional identifier that serves as a “hint” to the recipient of the specific Markdown variant that the author intended. When omitted, there is no hint; the interpretation is entirely up to the receiver and context. This identifier is plain US-ASCII and case-insensitive. [...] Other parameters MAY be included with the media type. [...] As an alternative, the variant MAY be registered under another media type; this text/markdown registration does not preclude other registrations.
  - Definition:
    - “Variant” means a **lightweight markup language that differs in some respect** from other Markdown-derived languages. The purpose of a variant is to distinguish a given variation from other Markdown variations, as well as from Gruber's original syntax specification and implementation, where two parties wish to interoperate by implementing the common variation.
- Eliminate (almost all of) Section 3 (syntax)
- Simplify Section 6 (IANA Considerations)
- Broaden scope of IANA Registry by calling it “Markdown Variants”; propose non-media-type labeling techniques (not normative)
  - “Content-Type: text/markdown; variant=pandoc” (MIME) == “file.pandoc.markdown” (filesystem)



## Proposal (output-type)

- Change output-type parameter to something in Content-Disposition [\[RFC2183\]](#) [\[RFC6266\]](#)
- Maybe “preview-type” (*cf.* hypothetical preview-image or preview-icon)
- Markdown draft or separate draft?
- Markdown-specific applicability, or broad applicability?

# Other Stuff

- Fragment Identifier Considerations
  - Media type registration is the primary/only place to define fragments
  - Generally, “not complicated” is desirable
  - text/plain fragment identifiers ok...but not really Markdown-y
  - Using “native” Markdown identifiers would prove useful
  - Encoding issues
- Two documents or one

# Discussion Time + Hums

- Shall the `syntax` parameter be renamed “variant”?
- Shall the `s/v` parameter be a simple identifier?
- Shall the `output-type` parameter be treated under `Content-Disposition`?
- Shall exemplary registrations be in a) one unified draft, b) a separate draft, or c) registered without IETF review (not in any draft)?
- Shall the Markdown `s/v` identifier registry have an explicitly broader purpose than just the `text/markdown` parameter registration?