Window Sizes for Zstandard Content Encoding

draft-jaju-httpbis-zstd-window-size

Nidhi Jaju and Felix Handte

HTTPBIS - IETF 119 Brisbane - March 2024

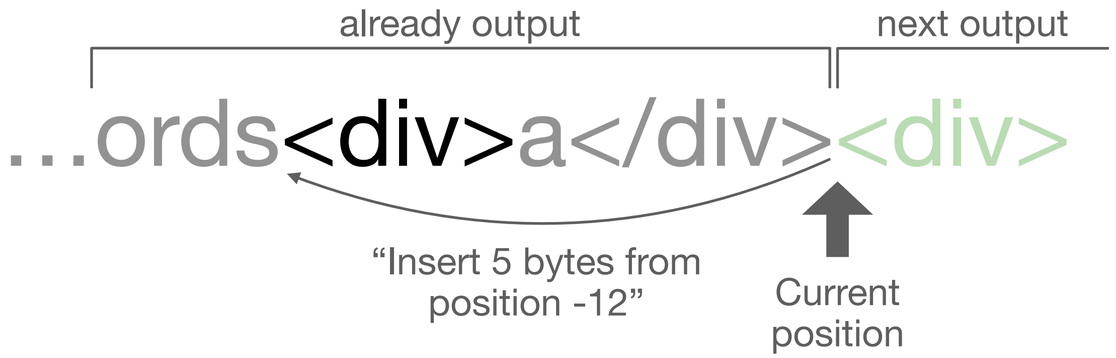
Zstandard

Data compression scheme designed by Meta (RFC 8878)

“Fast lossless compression algorithm, targeting real-time compression scenarios at zlib-level and better compression ratios.”

Higher compression ratios than gzip
Lower CPU cost than Brotli

Window Sizes



The sliding window is the last N bytes of decompressed data, where N is the longest back-reference that the encoder will emit

Larger window sizes → higher compression ratios

“For improved interoperability, it's recommended for decoders to support values of Window_Size **up to 8 MB** and for encoders not to generate frames requiring a Window_Size **larger than 8 MB**. It's **merely a recommendation** though, and decoders are free to support higher or lower limits, depending on local limitations.”

- RFC8878

Interoperability Issue

Zstd CLI

- Uses up to 8 MB window size by default
- Uses up to 128 MB window size if `--long` or `--ultra` flags are used

Chromium

- Accepts up to 8 MB window sizes

Curl

- Accepts up to 128 MB window sizes

Proposal

Change “`zstd`” Content Encoding token
to mean the compressed content used
a window size of not more than 8 MB

“To maintain interoperability of Zstandard in HTTP Content Encoding, decoders **MUST** support window sizes of **up to and including 8 MB** and encoders **MUST NOT** generate frames requiring a window size of **larger than 8 MB**, when using the "zstd" Content Encoding token.”

- draft-jaju-httpbis-zstd-window-size

Next Steps

Feedback welcome!

- Mailing List, Github Issues

Adoption?