

# Response Compression

Starting in Tapestry 5.1, the framework automatically GZIP **compresses** content streamed to the client. This can significantly reduce the amount of network traffic for a Tapestry application, at the cost of extra processing time on the server to compress the response stream.

This directly applies to both rendered pages and streamed assets from the CLASSPATH.

Context assets will also be compressed ... but this requires referencing such assets using the "context:" binding prefix, so that generated URL is handled by Tapestry and not the servlet container.

## Related Articles

- [Request Processing](#)
- [Configuration](#)
- [Assets](#)

## Compression Configuration

Small streams generally do not benefit from being compressed; there is overhead when using compression, not just the CPU time to compress the bytes, but a lot of overhead. For small responses, Tapestry does not attempt to compress the output stream.

The configuration symbol `tapestry.min-gzip-size` allows the cutoff to be set; it defaults to 100 bytes.

In addition, some file types are already compressed and should not be re-compressed (they actually get larger, not smaller!). The service [ResponseCompressionAnalyzer](#)'s configuration is an unordered collection of content type strings that should *not* be compressed. The default list of content types that are NOT compressed are:

- image/\* (image/jpeg, image/png, image/gif, etc) *except* image/svg+xml, which is compressed
- application/x-shockwave-flash
- application/font-woff
- application/x-font-ttf
- application/vnd.ms-fontobj

## StreamResponse

When returning a [StreamResponse](#) from a [component event method](#), the stream is totally under your control; it will not be compressed. You should use the [ResponseCompressionAnalyzer](#) service to determine if the client supports compression, and add a `java.util.zip.GZIPOutputStream` to your stream stack if compression is desired.