

## Technical Article

# HTTP Chunking

by Eric D. Larson

## Question

Why does the body of my HTTP transaction contain "extra" characters?

## Answer:

HTTP 1.1 supports chunked encoding, which allows HTTP messages to be broken up into several parts. Chunking is most often used by the server for responses, but clients can also chunk large requests.

HTTP 1.1 supports persistent connections by default. It is very important on wireless networks to do everything possible to avoid latency problems, and persistent connections are one way to reduce network latency by eliminating the overhead of creating a new connection for every transaction. If every HTTP request required the connection to tear down and set up again, performance would suffer greatly.

In persistent connections the length of every transaction must be counted exactly. If the whole message is available then a simple `Content-Length` header identifies the size of the request or response. The client or server just reads the number of bytes indicated by `Content-Length` from the stream, making it possible to use the same socket connection for the next request to (and response from) the same server.

One of the difficulties, typically experienced by interactive applications, is that they do not know how much data they are going to send. In HTTP 1.0 a server could just leave out the `Content-Length` header and keep writing to the connection. When the server was done, it would simply close the connection. The classic HTTP client would read until it got a -1 end-of-file indication. To get around this problem HTTP 1.1 added a special header, `Transfer-Encoding`, that allows the response to be chunked. Each write to the connection is pre-counted and a final zero-length chunk written at the end of the response signifies the end of the transaction.

In some cases a server or client may want the o  
a `Connection: close` header can be added to inf

[Start chat](#)[Contact or call](#)

arise when a MIDP client tries to communicate with a web server that only supports HTTP 1.0. the client sends a chunked request and the server doesn't understand it. The MIDP spec requires HTTP 1.1 servers, and without HTTP 1.1 you can't reliably handle persistent connections. This is a problem only when using older web servers, or custom server code that doesn't support persistent connections.

The problem surfaces when a server tries to read a chunked request it isn't prepared for. In the body of a chunked message, each chunk of data begins with the size of the chunk and an extra CR/LF:

```
C\r\n
Some data...
11\r\n
Some more data...
0\r\n
```

This message contains two chunks, the first is 12 bytes long (hex C), the second 17 bytes long (hex 11). If your server is expecting a single message of 29 bytes, you have a problem.

One way to reduce the number of chunks is to eliminate flush() calls and use only a single write() call if possible. Flush only if there is something significant about starting the HTTP request at that point in the application. Let the system handle its own output-stream buffering. Each platform might have a different optimal mechanism for segmenting the data.

Instead of code like this...

```
os.write(someBytes);
os.flush();
os.write(someMoreBytes);
os.flush();
os.close();
```

...if you can, simply use a single write, without calling flush():

```
os.write(allTheBytes);
os.close();
```

[Start chat](#)[Contact or call](#)

The J2ME Wireless Toolkit's Network Monitor feature provides an excellent way to debug MIDlet HTTP transactions. Optionally you could use a packet-sniffer utility such as snoop or tcpdump.

## Acknowledgments

Thanks to Gary Adams for answering this question on the KVM-INTEREST list

### Resources for

Developers  
Startups  
Students and  
Educators

### Partners

Oracle  
PartnerNetwork  
Find a Partner  
Log in to OPN

### Solutions

Artificial  
Intelligence  
Internet of Things  
Blockchain

### What's New

How we're taking  
on COVID-19  
Java 16 download  
Try Oracle Cloud  
Free Tier

### Contact Us

US Sales: +1.800.633.0738

How can we help?

Subscribe to emails

© 2021 Site Privacy / Do Cookie Ad Careers  
Country/Region Oracle Map Not Sell My Preferences Choices  
Info

Start chat

Contact or call