Progress notifications for HTTP
draft-decroy-http-progress-03

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 30, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).
Abstract

This document specifies extensions to HTTP to allow progress messages for user-agents during lengthy transactions.

Table of Contents

1. Terminology ...................................................... 3
2. Introduction and motivation .................................. 4
   2.1 The need for progress information. ...................... 4
   2.2 Aim of this document ..................................... 4
3. Header Field Definitions ....................................... 5
   3.1 Progress ..................................................... 5
   3.1.1 Client request .......................................... 5
   3.1.2 1xx Responses ........................................... 6
4. Timing ............................................................. 7
4.1 Progress Intervals ............................................. 7
5. Example ............................................................ 8
6. Compatibility issues ............................................. 9
7. Implementation Notes ........................................... 10
8. Security Considerations ....................................... 11
9. Notes & TODO ................................................... 12
10. IANA Considerations .......................................... 13
11. References ..................................................... 14
   11.1. Normative References .................................. 14
Author’s Address .................................................. 15
Intellectual Property and Copyright Statements ............... 16
1. Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT" and "MAY" that appear in this document are to be interpreted as described in [RFC2119]
2. Introduction and motivation

2.1 The need for progress information

Increasingly, resource transfers using HTTP are subject to ever-more complex processing, particularly by proxy servers. Processing such as scanning resources for viruses at an HTTP proxy creates special problems that cannot be resolved cleanly with the current specification of HTTP.

Many types of processing require access to an entire message body. This can take considerable time to accumulate depending on upstream link bandwidth and/or other factors. It may not be safe to send any part of the resource to the client until processing is complete.

Frequently problems occur where human users of client agents give up waiting for visible progress, typically resulting in retries wasting time and network resources.

Furthermore, some automated clients will give up waiting when no resource data is received within a certain timeframe.

There is a clear need for upstream agents to be able to provide timely progress notifications to downstream agents to enable them to make proper decisions about whether it is appropriate to keep waiting.

2.3 Aim of this document

This document aims to solve this problem by providing a defined mechanism whereby intermediaries or server agents can provide progress notifications back to the client agent (and the human user), thereby avoiding inappropriate timeouts, and retries.
3 Header field definitions

This document defines a new header field to indicate progress. This field allows optional textual state indication as well as requiring numeric indication of completeness.

3.1 Progress

The Progress field MAY appear in any request message sent by a client or in any server or proxy-generated 1xx series of response messages. The Progress field MUST not appear in any other series of response messages.

A server that complies with this document upon receiving a request containing a Progress field MUST provide an interim response message (1XX) within a reasonable time period AND provide further periodic updates until the final response to the request is sent.

A proxy that complies with this document upon receiving a request containing a Progress field MUST forward the tag unchanged to any upstream agent. The proxy however is responsible for providing progress messages to the client, and in the event that no upstream notifications are available, it MUST satisfy the client progress notifications whilst it is still prepared to process the request.

The proxy SHOULD pass any 1XX messages back through to the client unchanged.

Progress indication is worthless if it is not timely. A discussion of timing is in section 4

The form of the field is defined below

3.1.1 Client request

Progress = "Progress" "::" interval

interval = 1*4DIGIT

; the time (s) within which the client expects a response

Example:

Progress: 10
3.1.2 1xx Responses

Progress = "Progress" ":" prog-num [SP prog-text]
prog-text = quoted-string
prog-num = percent | (bytes-processed "/" bytes-total)
percent = "%" %d0-100
; integer percentage complete, 0 - 100
bytes-processed = number
bytes-total = number | "UNKNOWN"

The textual information is intended to be displayed verbatim to a
user in an area usually associated with progress indication. The
numeric-progress field can be used to display a progress bar and/or
show how much data has been received by an upstream proxy.

The prog-text SHOULD be in a natural language and character set that
is most likely to be intelligible to the human user receiving the
response. This decision MAY be based on any available knowledge. The
default language is English and the default character set is
ISO-8859-1. If a character set other than ISO-8859-1 is used, it MUST
be encoded in the prog-text using the method described in [RFC2047].

Examples:

Progress: 20480/UNKNOWN "Generating content"

This could be sent by a server to indicate progress of generation of
content.

Progress: 20480/UNKNOWN "Download in progress"

Could be sent by an upstream proxy that is retrieving a message body,
has received 20480 bytes, and doesn’t know the content length.

Progress: 1200000/1200000 "Download complete, scanning"

Could be sent by an upstream proxy that has just completed retrieving
a message body, has received 1200000 of 1200000 bytes, and is about
to scan the content before sending it on to the client.

Progress: %25 "Scanning content for viruses"

Could be sent by an upstream proxy that is currently 25% through
virus scanning of a message body.
4. Timing

4.1 Progress intervals

The client Progress header indicates a time in seconds which is the time within which it expects a response (either final or interim containing progress information). The server or proxy SHOULD use this specified time as the initial time and periodic time for updates, or choose another time. If the server chooses to use another periodic interval, this should be one chosen with regard to the usefulness of the interval to a waiting human, and it is suggested that an interval of about 5 - 10 seconds would be appropriate.

Agents generating progress notifications MAY choose to send a notification whenever any significant change in state occurs.

However in the interests of bandwidth, agents SHOULD NOT send progress notifications more frequently than once per second. This includes an intermediary which may be generating notifications and relaying notifications from upstream.

It is left up to the implementor of intermediaries to choose which notifications they may choose to relay or generate themselves, remembering that this is intended for a human user, but will be useful to automated agents as well.
5. Example

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>Intermediary</td>
<td>Server</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------</td>
<td>-------------------</td>
</tr>
</tbody>
</table>

Connects to intermediary

->
GET http://www.wingate.com/bigfile.zip HTTP/1.1
Host: www.wingate.com
Progress: 10

->
GET /bigfile.zip HTTP/1.1
Host: www.wingate.com
Progress: 10

<- HTTP/1.1 200 OK
Content-Length: 20000000
...

time elapses
<- HTTP/1.1 102 Processing
Progress: 1000000/20000000 "Downloading"

time elapses
<- HTTP/1.1 102 Processing
Progress: 20000000/20000000 "Downloaded, Scanning"

time elapses
<- HTTP/1.1 102 Processing
Progress: 25 "Scanning"

time elapses
<- HTTP/1.1 102 Processing
Progress: 75 "Scanning"

<- HTTP/1.1 200 OK
Content-Length: 20000000
...
Transfers resource to client.

None identified. RFC2616 already mandates that any agent must be able to cope with multiple interim responses.

A server not understanding the Progress header in a request will not generate progress notifications.

A client unaware of this document will not generate requests with the Progress header.

A proxy not understanding the Progress header already should be passing unknown headers through to upstream, and also passing back 1XX responses.
7. Implementation Notes

Most user agents provide screen real-estate to display progress, often in the "status bar" of the window. It is envisaged that the progress notifications outlined in this document would be shown in there.
8. Security Considerations

No security issues identified with use of this proposal.
9. Notes & TODO

* modified from version 00 to remove sections on flow control issues.

* modified from version 01 to allow for internationalisation of strings, slight modification to syntax. Minor editorial changes.
10. IANA Considerations

None.
11. References

11.1. Normative References


Author’s Address

Adrien de Croy
Qbik New Zealand Limited
PO Box 3548
Shortland St
Auckland
New Zealand

Email: adrien@qbik.com
URI: http://www.wingate.com/