

在语义上发生了重大改变时，标记才会变化。而强实体标记则不管关联的实体发生了什么性质的变化，标记都一定会改变。

下面的例子展示了客户端如何用弱实体标记向服务器请求再验证。服务器仅当文档的内容从版本 4.0 算起发生了显著变化时，才返回主体：

```
GET /announce.html HTTP/1.1
If-None-Match: W/"v4.0"
```

概括一下，当客户端多次访问同一个资源时，首先需要判断它当前的副本是不是仍然新鲜。如果不再新鲜，它们就必须从服务器获取最新的版本。为了避免在资源没有改变的情况下收到一份相同的副本，客户端可以向服务器发送有条件的请求，说明能唯一标识客户端当前副本的验证码。只在资源和客户端的副本不同的情况下服务器才会发送其副本。更多关于缓存再验证的细节，请回顾 7.7 节。

## 15.9 范围请求

关于客户端如何要求服务器只在资源的客户端副本不再有效的情况下才发送其副本，我们已经清楚地理解了。HTTP 还进一步锦上添花：它允许客户端实际上只请求文档的一部分，或者说某个范围。

363

假设你正通过慢速的调制解调器连接下载最新的热门软件，已经下了四分之三，忽然因为一个网络故障，连接中断了。你已经为等待下载完成耽误了很久，而现在被迫要全部重头再来，祈祷着别再发生这样的倒霉事了。

有了范围请求，HTTP 客户端可以通过请求曾获取失败的实体的一个范围（或者说一部分），来恢复下载该实体。当然这有一个前提，那就是从客户端上一次请求该实体到这次发出范围请求的时段内，该对象没有改变过。例如：

```
GET /bigfile.html HTTP/1.1
Host: www.joes-hardware.com
Range: bytes=4000-
User-Agent: Mozilla/4.61 [en] (WinNT; I)
...
```

在本例中，客户端请求的是文档开头 4000 字节之后的部分（不必给出结尾字节数，因为请求方可能不知道文档的大小）。在客户端收到了开头的 4000 字节之后就失败的情况下，可以使用这种形式的范围请求。还可以用 Range 首部来请求多个范围（这些范围可以按任意顺序给出，也可以相互重叠）。例如，假设客户端同时连接到多个服务器，为了加速下载文档而从不同的服务器下载同一个文档的不同部分。对于客户端在一个请求内请求多个不同范围的情况，返回的响应也是单个实体，它有