例如，文件名 Sven Ölssen.html（包含了一个元音变音）可能被网站服务器编码为 Sven%20%D6lssen.html。把空格编码为 %20 是对的，但从技术上说，把 Ö 编码为 %D6 是非法的，因为代码 D6（十进制值 214）落在了 ASCII 代码范围之外。ASCII 只定义了最大值为 0x7F（十进制值 127）的代码。

16.5.5 URI中的模态切换

有些 URI 也用 ASCII 字符的序列来表示其他字符集中的字符。例如，可能使用 iso-2022-jp 编码插入“ESC (J”，切换到 JIS-Roman 字符集，用“ESC (B”切换回 ASCII 字符集。这在一些本地化的环境中可以工作，但这种方式没有进行良好的定义，而且没有标准化的方案来识别 URL 所使用的特定编码。正如 RFC 2396 的作者所说的那样：

> 不过，对于含有非ASCII字符的原始字符序列来说，境况更加复杂。如果可能用到多个字符集的话，传输表示字符序列的8位字节序列的因特网协议期待能有办法来识别所用的字符集[RFC 2277]。
>
> 然而，在通用的URI语法中没有提供进行这种识别的手段。个别的URI方案可以请求单一的字符集，定义默认的字符集，或提供指示所用字符集的方法。期待将来对这个规范的修改能为URI中的字符编码提供一种系统化的处理方案。

目前，URI 对国际化应用还不是非常友好。URI 的可移植性目标比语言灵活性方面的目标更重要。人们正在尽最大努力使 URI 更加国际化，但在短期内，HTTP 应用程序还是应当坚持使用 ASCII。它从 1968 年就出现了，所以只用它的话，一切还不至于太糟。

391

16.6 其他需要考虑的地方

本节讨论在编写国际化的 HTTP 应用程序时，必须牢记的其他一些东西。

16.6.1 首部和不合规范的数据

HTTP 首部必须由 US-ASCII 字符集中的字符构成。不过，并不是所有的客户端和服务器都正确地实现了这一点，你可能会时不时收到一些代码值大于 127 的非法字符。

很多 HTTP 应用程序使用操作系统和库例程来处理字符（比如 Unix 中的字符分类库 ctype），但不是所有这些库都支持 ASCII 范围（0 ~ 127）之外的字符代码。

在某些情况下（一般来说，是较老的实现），当输入非 ASCII 字符时，这些库可能会返回不正确的结果，或者使应用程序崩溃。假设报文中含有非法数据，在使用这