

组织：中国互动出版网 (<http://www.china-pub.com/>)

RFC 文档中文翻译计划 (<http://www.china-pub.com/compters/emook/aboutemook.htm>)

E-mail: ouyang@china-pub.com

译者：顾国飞 (ggfei ggfei@263.net)

译文发布时间：2001-4-2

版权：本中文翻译文档版权归中国互动出版网所有。可以用于非商业用途自由转载，但必须保留本文档的翻译及版权信息。

Network Working Group
Request for Comments: 856

Obsoletes: NIC 15389

J. Postel
J. Reynolds
ISI
May
1983

RFC856 TELNET 二进制传输

(RFC856--TELNET BINARY TRANSMISSION)

本 RFC 规范了一个 ARPA Internet community 上的标准。在 ARPA Internet 上的所有主机应当采用和实现这个标准。

目录

1. 命令和代码	2
2. 命令意义	2
3. 默认情况	2
4. 选项出现的原因	2
5. 选项描述	3
6. 实现问题	3

1. 命令和代码

TRANSMIT-BINARY 0

2. 命令意义

- **IAC WILL TRANSMIT-BINARY**
这个命令请求的发送方请求开始传输, 或确定现在要传输的数据在接收方会以八位二进制方式解释。
- **IAC WON'T TRANSMIT-BINARY**
如果连接已在二进制方式下, 发送此命令要求接收方恢复原来标准的 NVT ASCII 方式解释数据。如果现在还未在二进制方式下, 发送方拒绝传输将被接受者解释为二进制数据的字符(也就是说, 数据传输者要求继续按现在方式进行传输)。只有当双方均同意的情况下才有可能进行二进制传输。
- **IAC DO TRANSMIT-BINARY**
发送者要求传输数据, 或确定数据将要被传输, 这些数据均被解释为 8 位二进制的。
- **IAC DON'T TRANSMIT-BINARY**
如果现在处于二进制状态下, 命令发送方要求数据发送方进行标准的 NTV ASCII 的传输。如果连接未在二进制状态下, 发出命令者要求数据发送方按现在的状态发送数据。只有当双方均同意的情况下才有可能进行二进制传输。

3. 默认情况

默认情况为: WON'T TRANSMIT-BINARY 和 DON'T TRANSMIT-BINARY, 连接未在二进制状态下。

4. 选项出现的原因

有时候利用 telnet 上的二进制传输会更有效率, 这就是出现的根本原因吧。而双方只要把对数据的解释方式加以改变就可以完成这一选项, 因此也比较方便。

5. 选项描述

开始二进制传输后，接收方对没有 IAC 开始的数据以二进制进行解释。IAC 后面的是标准的 TELNET 命令。如果 IAC 后面的命令不可识别，它和 IAC NOP 命令的效果一样。

6. 实现问题

实现二进制传输则不能进行其它模式的传输，这是可以预见的。然而，如果双方能够理解它们同处于二进制传输模式或者例如它们同处于 Echo 模式，如果他们对此进行了协商，则不会出现什么问题。

我们看到上面的命令意义解释可以注意到 WON' T 和 DON' T 的意义要看现在是不是处于二进制传输模式下，假设现在处于 EBCDIC 模式下，而且一方也不知道任何二进制传输的命令，如果它接收到 DO TRANSMIT-BINARY，它根本不知道这是什么，因此返回 WON' T TRANSMIT-BINARY，如果对于 WON' T TRANSMIT-BINARY 的默认值是 NVT ASCII，发送 DO TRANSMIT-BINARY 可能希望接收方转到 NVT ASCII，但接收 DO TRANSMIT-BINARY 的一方有可能不这么做。

因此，我们有这样一条规则：当连接不处于二进制状态时，默认值（也就是说，对 WON' T 和 DON' T 的解释）是维持现状，无论是在在 NVT ASCII, EBCDIC 或者是其他状态。然而，当连接处于二进制状态时，这规则就不顶用了。这就要求连接双方维持一个保存所有可用的连接状态的栈，这样才能正确解释 WON' T 和 DON' T。在二进制状态下，WON' T 和 DON' T 会使状态返回 NVT ASCII。

因为 telnet 是一个双向的通道，因此必须保证双向的数据流都是二进制的。在实现时遵守防止循环的规则，这一规则在 telnet 协议中有描述。下面我们看看从一个进程和终端开始或接收二进制传输的情况：

a. 从终端开始二进制传输

实现者应该考虑在二进制状态下如何产生 8 位有效数据，其中不带什么校验位之类的东西。

b. 二进制传输到进程

实现者应该考虑在二进制状态下如何进行如何接收所有的二进制数据。例如 TOPS-20 会在终端级解释一些特定字符（例如，ETX, 中断 control-C），而不把它们传送到进程。

c. 从进程开始的二进制传输

实现者应该考虑传输的字符如何不对对方的终端解释为其它的字符。例如 TOPS-20 会将非打印字符转为一个箭头和一个可打印字符。

d. 二进制传输到终端

实现者应该考虑接收到的数据如何传送到本地终端。包括本地应该加入的一些字符，校验运算或字符转换。