

文件传输协议的特征流通机制

本备忘录状态

本文档为Internet community说明了一种Internet标准追踪协议。它需要进一步进行讨论和建议以得到改进。请查询“Internet 正式协议标准” (STD 1) 的最新版本，以获得这个协议的标准化状况。发布本备忘录不受限制。

概要

文件传输协议有时会扩展新的指令或工具。FTP协议的执行并未假定所有最新定义的机制要立即执行。本文提供了一种机制，它能使FTP协议的客户端发现指定的FTP服务器支持的新特性。

1. 介绍

本文为FTP协议增加了两个新的指令：“FEAT”和“OPTS”。

这些指令允许客户端发现服务器支持的可选择的指令，和它们如何被支持，和从任何能够支持的FTP指令中进行选择。

2. 文档规范

本文使用了BCP14[2]定义的文档规范。它提供了一些像MUST, SHOULD, 等大写字母单词的解释。

[1]中定义的条款将在这里使用。包括ASCII, reply, server-FTP process, user-FTP process, server-PI, user-PI, and user。

语法使用[3]中定义的Augmented BNF。一些常用的ABNF定义贯穿全文。在阅读之前，请简单地回忆一下这些定义。

2.1基本记号

本文引用的定义在[3]的附录A中给出。这些定义可以在基本的ABNF原理中找到，比如像ALPHA, DIGIT, VCHAR, SP, 等。为了本文的使用，增加了下列条款：

TCHAR = VCHAR / SP / HTAB ; visible plus white space

TCHAR类型和[3]中给出的VCHAR类型，给出了在不同的指令和回应中使用的基本字符类型，它们来自于ASCII字符集的不同子集。

error-response = error-code SP *TCHAR CRLF
error-code = ("4" / "5") 2DIGIT

注意，在ABNF中，字符串是不区分大小写的。本文仍保持这个约定。不过注意，一般地，ALPHA是区分大小写的，VCHAR和TCHAR也是区分的。

2.2 服务器回应

[1]的4.2节定义了server-PI对来自于user-PI的FTP指令的回应的格式和含义。这些约定在此没有改变。执行者会注意到，本文中的ABNF语法（这在[1]中没有使用）和其他FTP回应文本，有时用一行的格式给出回应。除了特别说明，这并不意味着多行的回应是不允许的。执行者也应假定，本文中任何对于任何FTP指令（包括QUIT）的回应可以使用多行的格式来描述。回应它们的第一个参数会被定义为3个数字的代码。所以“500 Reply”的意思是一个使用了3个数字代码“500”的来自server-PI的回应。

3. 额外性能的知识——FEAT指令

没必要期望所有的服务器都必须支持未来FTP协议改善后定义的所有新指令。为了让客户端能检测特定的服务器支持那些新指令，而不必尝试所有可能的指令，FTP指令系统为此增加了一条新指令。该指令请求服务器列出它支持的所有的扩展指令，或者扩充机制。就是说，所有定义和说明过的指令和特征（但没有在[1]中和本文中定义）必须被包含在FEAT指定的的输出表格里。

在FEAT指令的回应中，User-FTP 列出的未知特征必须可以看到。这不是一个错误，而是简单地指出，对user-FTP未知的新特征规范，server-FTP执行者已经看到和执行了。

3.1 特征指令（FEAT）语法

feat = "Feat" CRLF

FEAT指令由单词“FEAT”单独组成。没有参数。

3.2 FEAT指令回应

当server-FTP处理不支持FEAT指令时，它将对FEAT指令回应一个500或502的应答。这个简单的回应一般是“未被接受的指令”，任何未知的指令都可以引用。如果命令语法有错误，比如使用了参数，将回应一个501应答。

Server-FTP处理认可FEAT指令，但不执行扩展特征，所以没有任何应答，应该回应“no-features”的211应答。不过，在这种情况下，server-FTP不认可FEAT指令，实际上是无法识别的，因为500或502应答也可以这样使用。“no-features”应答必须不使用多行的应答格式，只允许和需要一行正确的回应。

对FEAT指令的应答必须遵照以下语法。应答文本的第一行是自由形式的，没有解释，也没有特殊的作用，因为该文本不会显示给最终用户。其他应答行的语法必须遵照规范的定义。

```
feat-response = error-response / no-features / feature-listing
no-features   = "211" SP *TCHAR CRLF
feature-listing = "211-" *TCHAR CRLF
               1*( SP feature CRLF )
               "211 End" CRLF
feature       = feature-label [ SP feature-parms ]
feature-label = 1*VCHAR
feature-parms = 1*TCHAR
```

注意，每个feature-listing中的特征行都以一个单独的空格开始。这个空格不是可有可无的，也不是一般的留空。该空格保证在feature-listing的最后不被曲解，甚至没有模糊的可能，它也是必须的。每个扩展支持必须在一个单独的行中列出，以便于每个扩展指令支持的可能包含的参数。feature-label在FEAT指令的回应中使用，它指明FTP指令集中增加的每个新特征。通常，它随着一个新增加的指令命名，尽管这不是必需的。实际上一个新特征加入一个新指令时，它也不是必需的。任何包含的参数都在命令有关的说明中定义。这个规范也指明多少参数会被解释。

feature-label和feature-parms名义上是区分大小写的，不过，关于指定的标签和指定的参数的精确解释的定义，会指定通常情况下标签和参数应该遵循的大小写风格。当传输特征回应时，推荐使用大写的字母。

FEAT指令本身不包含在被支持的特征列表里，支持FEAT指令包含在除了500或502应答的回应里。

一个典型的回应FEAT指令的例子应该是像下面这个形式这样的多行回应：

```
C> feat
S> 211-Extensions supported:
S> MLST size*;create;modify*;perm;media-type
S> SIZE
S> COMPRESSION
S> MDTM
S> 211 END
```

详细扩展显示了如何在其他地方定义的简单例子。对于回复也是，返回的特征名没有指令名，同样地，只简单地指出服务器拥有的属性或其他信息。

返回的特征顺序并不重要，server-FTP执行不需要特殊的顺序，甚至当指令重复时，会一直返回相同的顺序。支持FEAT的FTP执行必须包含在FEAT指令的回应里，对所有被证明适当的FTP扩展，超出的指令和机制在RFC959 [1]描述，包含任何FEAT存在之前存在的。就是说，当客户端收到FTP服务器的FEAT回应，它会假设服务器支持的扩展是FEAT回应中列出的那些。

User-FTP会清楚地处理已经开发的一些FTP扩展，它们在本文和FEAT指令发表之前已被广泛使用。这样做的结果是对于FEAT指令的错误回应并不需要暗示这些扩展不被server-FTP处理支持。如果一个User-PI收到关于FEAT指令的错误回应，它将能对这样个别的扩展进行测试。

3.3FEAT的基本原理

当不绝对需要的时候，一个server-PI的标准机制通知user-PI任何支持的特征和扩展，如果未来还要引入更多的扩展的话，这将有助于减少user-PI和server-PI之间多余的通信。如果没有这个机制存在，user-FTP 处理将不得不从user-PI 和 server-PI之间的一系列交换中轮流查找每个扩展。

4. OPTs指令

当其他FTP指令（目标指令）稍后发出时，OPTs（options）指令允许user-PI指定希望得到的server-FTP处理行为。准确的行为和语法，将随着目标指令而变化，并将被目标指令的定义所说明。在没有为特殊的指令定义OPTs行为的地方，该指令没有有效的选项。

请求语法：

```
opts          = opts-cmd SP command-name
                [ SP command-options ] CRLF
opts-cmd       = "opts"
command-name   = <any FTP command which allows option setting>
command-options = <format specified by individual FTP command>
```

回应语法：

```
opts-response = opts-good / opts-bad
opts-good     = "200" SP response-message CRLF
opts-bad      = "451" SP response-message CRLF /
                "501" SP response-message CRLF
response-message = *TCHAR
```

当OPTs指令中指定的command-name经过验证后，必须发送一个“opts-good”回应（200应答）。command-options也要是适当的和经过验证的。一个

“opts-bad”的回应在其他情况下被发送。501回应对任何的永久性错误都适用。就是说，在若干稍候的时间里仅仅重复指令的任何情况，状态不会有其他的改变，也是一个错误。在服务器一些暂停的情形下会发送451回应，和用户与服务器之间的通信状态无关，防止指令发出后被接受。如果在稍候的时间里重复发生，server-FTP 的转换环境会作出处理，允许指令成功。不管是FEAT指令，还是OPTS指令，都没有任何选择功能，所以没有“OPTS FEAT”或“OPTS OPTS”指令。

5. 安全考虑

除了已经存在于FTP协议中，没有被认为是用该扩展造成的重大的安全问题。不过，这个扩展确实提供了一种用户能够决定FTP服务器的能力的机制，并且可以从附加信息作出判断。通过对服务器使用不同的指令进行试探可以获取相同的基本信息。如果不提供FEAT指令，该方法将显示一个正在尝试用不同的扩展指令登录的攻击者。没有充分考虑补救活动的可能性。

附加特征的安全性在FEAT指令中报告，通过OPTS指令操作，将对这些定义的特征做出报告。

6. 参考书目

- [1] Postel, J. and J. Reynolds, “文件传输协议(FTP)”, STD 9, RFC 959, 1985年10月。
- [2] Bradner, S., “使用RFCs的关键字来指出需求级别”, BCP 14, RFC 2119, 1997年3月。
- [3] Crocker, D. and P. Overell, “扩展的BNF语法规范: ABNF”, RFC 2234, 1997年十一月。

感谢

(略——译者注)

作者地址

(略——译者注)

完整的版权声明

(略——译者注)

原文：RFC 2389 《Feature negotiation mechanism for the File Transfer Protocol》

译者：张偶 2002年5月