

FTPEXT2 Working Group  
Internet-Draft  
Updates: 959 (if approved)  
Intended status: Standards Track  
Expires: December 31, 2011

P. Hethmon  
Hethmon Brothers  
R. McMurray  
Microsoft Corporation  
June 29, 2011

File Transfer Protocol HOST Command for Virtual Hosts  
draft-ietf-ftpext2-hosts-04

Abstract

The File Transfer Protocol, as defined in RFC 959 [RFC0959], does not provide a way for FTP clients and servers to differentiate between multiple DNS names that are registered for a single IP address. This document defines a new FTP command that provides a mechanism for FTP clients and servers to identify individual virtual hosts on an FTP server.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on December 31, 2011.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1.	Introduction . . . . .	3
2.	Document Conventions . . . . .	3
2.1.	Basic Tokens . . . . .	4
2.2.	Server Replies . . . . .	4
3.	The HOST command . . . . .	5
3.1.	Syntax of the HOST command . . . . .	5
3.2.	HOST command semantics . . . . .	8
3.2.1.	REIN command semantics . . . . .	8
3.2.2.	User-PI usage of HOST . . . . .	9
3.2.3.	State Diagrams . . . . .	10
3.3.	HOST command errors . . . . .	17
3.4.	FEAT response for HOST command . . . . .	18
4.	Security Considerations . . . . .	18
5.	IANA Considerations . . . . .	19
6.	References . . . . .	19
6.1.	Normative References . . . . .	19
6.2.	Informative References . . . . .	20
Appendix A.	Unworkable Alternatives . . . . .	20
A.1.	Overloading the CWD command . . . . .	21
A.2.	Overloading the ACCT command . . . . .	21
A.3.	Overloading the USER command . . . . .	22
A.4.	Conclusion . . . . .	23
Appendix B.	Acknowledgements . . . . .	23
Authors' Addresses	. . . . .	23

## 1. Introduction

It is common on the Internet for many DNS names to resolve to a single IP address. This practice has introduced the concept of a "virtual host", where a host appears to exist as an independent entity, but in reality shares its physical resources with one or more similar hosts.

Such an arrangement presents some problems for FTP servers, as an FTP server distinguishes incoming FTP connections by their IP addresses, not their DNS names, because hosts are uniquely identified by their address rather than name. That is, all DNS names that share an IP address are handled by the same FTP server and share the same Network Virtual File System (NVFS).

This means that different virtual hosts cannot offer different virtual file systems to clients, nor can they offer different authentication systems. Any scheme to overcome this issue needs to indicate not only the destination IP address, but also the virtual host name that is associated with the desired virtual FTP server. Typical user-FTP processes currently use hostnames to perform hostname to IP address resolution and then ignore hostnames for the rest of the FTP session, therefore any mechanism to overcome this issue would require modifications to the user-PI and server-PI.

It should be noted that this same problem existed for HTTP/1.0 as defined in [RFC1945], and was resolved in HTTP/1.1 as defined in [RFC2616] through the addition of the Host request header. The goal of this document is to bring a similar level of feature parity to FTP by introducing a new HOST command that allows user-FTP processes to specify which virtual host to connect to for a server-FTP process that is handling requests for multiple virtual hosts on a single IP address.

## 2. Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

In examples, "C>" and "S>" indicate lines sent by the client and server, respectively.

This document also uses notation defined in [RFC0959] and [RFC1123]. In particular, the terms "reply", "user", "NVFS", "NVT", "file", "pathname", "FTP commands", "DTP", "user-FTP process", "user-PI", "user-DTP", "server-FTP process", "server-PI", "server-DTP", "mode",

"type", "control connection", "data connection", and "ASCII", are all used here as defined there.

Syntax required is defined using the Augmented BNF defined in [RFC5234]. Some general ABNF definitions are required throughout the document; those will be defined later in this section. At first reading, it may be wise to simply recall that these definitions exist here, and skip to the next section.

## 2.1. Basic Tokens

This document imports the core definitions given in Appendix B of [RFC5234]. There definitions will be found for basic ABNF elements like ALPHA, DIGIT, SP, etc. To that, the following term is added for use in this document.

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

The VCHAR (from [RFC5234]) and TCHAR rules give basic character types from varying sub-sets of the ASCII character set for use in various commands and responses.

Note that in ABNF, string literals are case insensitive. That convention is preserved in this document, and implies that FTP commands and parameters that are added by this specification have values that can be represented in any case. That is, "HOST" is the same as "host", "Host", "HoSt", etc., and "ftp.example.com" is the same as "Ftp.Example.Com", "fTp.eXample.cOm", etc.

## 2.2. Server Replies

Section 4.2 of [RFC0959] defines the format and meaning of replies by the server-PI to FTP commands from the user-PI. Those reply conventions are used here without change.

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

Implementers should note that the ABNF syntax (which was not used in [RFC0959]) used in this document, and other FTP related documents, sometimes shows replies using the one line format. Unless otherwise explicitly stated, that is not intended to imply that multi-line responses are not permitted. Implementers should assume that, unless stated to the contrary, any reply to any FTP command (including QUIT) can be of the multi-line format described in [RFC0959].

Throughout this document, replies will be identified by the three digit code that is their first element. Thus the term "500 reply"

means a reply from the server-PI using the three digit code "500".

### 3. The HOST command

A new command "HOST" is added to the FTP command set to allow the server-FTP process to determine to which of possibly many virtual hosts the client wishes to connect. This command SHOULD be issued before the user is authenticated, allowing the authentication scheme, and set of legal users, to be dependent upon the virtual host chosen.

Server-FTP processes MUST treat a situation where the HOST command is issued more than once before the user has been authenticated as though a REIN command was sent before each HOST command, and then reset the user-PI to the state that existed after the TCP connection was first established and return the appropriate reply for the HOST command.

Server-FTP processes MUST treat a situation where the HOST command is issued after the user has been authenticated by using one of the following two behaviors:

- a. Treat the late HOST command as an erroneous sequence of commands and return a 503 reply.
- b. Treat the late HOST command as though a REIN command was sent before the HOST command and reset the user-PI to the state that existed after the TCP connection was first established and before the initial user authentication and then return the appropriate reply for the HOST command.

Servers should note that the response to the HOST command is a sensible time to send their "welcome" message. This allows the message to be personalized for any virtual hosts that are supported, and also allows the client to determine the supported languages, or representations, for the message, and other messages, via the FEAT response, and select an appropriate one via the LANG command. See [RFC2640] for more information.

#### 3.1. Syntax of the HOST command

The HOST command is defined as follows.

```

host-command = "HOST" SP hostname CRLF
hostname     = domain / IP-literal

domain       = sub-domain *("." sub-domain)
sub-domain   = let-dig [ldh-str]
let-dig      = ALPHA / DIGIT
ldh-str      = *( ALPHA / DIGIT / "-" ) let-dig

IP-literal   = ( "[" IPv6address "]" ) / IPv4address

IPv6address  =
    /
    / [ h16 ] "::" 5( h16 ":" ) ls32
    / [ *1( h16 ":" ) h16 ] "::" 4( h16 ":" ) ls32
    / [ *2( h16 ":" ) h16 ] "::" 3( h16 ":" ) ls32
    / [ *3( h16 ":" ) h16 ] "::" 2( h16 ":" ) ls32
    / [ *4( h16 ":" ) h16 ] "::" h16 ":" ls32
    / [ *5( h16 ":" ) h16 ] "::" ls32
    / [ *6( h16 ":" ) h16 ] "::" h16

ls32         = ( h16 ":" h16 ) / IPv4address
              ; least-significant 32 bits of address

h16          = 1*4HEXDIG
              ; 16 bits of address represented in hexadecimal

IPv4address  = dec-octet "." dec-octet "." dec-octet "." dec-octet

dec-octet    = DIGIT           ; 0-9
              / %x31-39 DIGIT ; 10-99
              / "1" 2DIGIT     ; 100-199
              / "2" %x30-34 DIGIT ; 200-249
              / "25" %x30-35    ; 250-255

host-response = host-ok / error-response
host-ok       = "220" [ SP *TCHAR ] CRLF

```

As with all FTP commands, the "HOST" command word is case independent, and MAY be specified in any character case desired.

The "hostname" (given as a parameter) specifies the virtual host to which access is desired. This SHOULD be the same host name that was used to obtain the IP address to which the FTP control connection was made, after any client conversions have been completed that convert an abbreviated or local alias to a complete (fully qualified) domain name, but before resolving a DNS alias (owner of a CNAME resource record) to its canonical name.

Internationalization of domain names is only supported through the use of Punycode as described in [RFC3492].

If the user was given an IPv4 or IPv6 literal address, and consequently was not required to derive the literal address from a hostname, the client MAY send the HOST command with the IPv4 or IPv6 literal address as specified to it. While it may seem counter-intuitive to specify a literal address by using the HOST command after the client has already connected to the server using a literal address, this should be expected behavior because a user-FTP process should not be required to differentiate between a fully qualified domain name and an IPv4 or IPv6 network literal address. That being said, if the IPv4 or IPv6 literal address specified by the client does not match the literal address for the server, the server MUST respond with a 504 reply to indicate that the IPv4 or IPv6 literal address is not valid.

When the hostname parameter contains a literal address, square brackets are expected to disambiguate IPv6 address syntax from port numbers syntax. Therefore, if the literal address is an IPv6 address, the IPv6 address is required to be enclosed in square brackets (after eliminating any syntax that might also - but is not required to - be enclosed in brackets, and from which the server deduced that a literal address had been specified.) For example, the following examples MAY be sent if the client had been instructed to respectively connect to "192.0.2.1", "FE80::c000:0201", or "192.0.2.1" and IPv6 syntax is preferred:

```
HOST 192.0.2.1
HOST [FE80::c000:0201]
HOST [::192.0.2.1]
```

The client MUST NOT send the port number as part of the HOST command, even when the client has been instructed to connect to a non-standard port. The reason for this requirement is that the user-PI will have established a connection to the server-PI before the HOST command is sent, therefore specifying a different port with the HOST command has no meaning. For example, the server-PI MUST respond with a 501 reply if the client sends a HOST command with syntax like either of the following examples:

```
HOST 192.0.2.1:2112
HOST [FE80::c000:0201]:2112
```

The hostname parameter is otherwise to be treated as a fully qualified domain name or relative name as those terms are defined in section 3.1 of [RFC1034]. This implies that the name is to be treated as a case-independent string, meaning that uppercase ASCII

characters are to be treated as equivalent to their corresponding lowercase ASCII characters, but otherwise preserved as given. It also implies some limits on the length of the parameter and of the components that create its internal structure. Those limits are not altered in any way here.

Neither [RFC1034] nor [RFC1035] impose any other restrictions upon what kinds of names can be stored in the DNS. This specification, however, only allows the use of names that can be inferred from the ABNF grammar given for the "hostname".

### 3.2. HOST command semantics

Upon receiving the HOST command, before authenticating the user-PI, a server-FTP process SHOULD validate that the hostname given represents a valid virtual host for that server, and, if it is valid, establish the appropriate environment for that virtual host. The resultant actions needed to create that environment are not specified here, and may range from doing nothing at all, to performing a simple change of working directory, changing authentication schemes and/or username and password lists, or making much more elaborate state changes - such as creating isolated environments for each FTP session.

The "220" reply code for the HOST command is the same as the code that is used in the initial "welcome" message that is sent after the connection is established.

If the hostname specified would normally be acceptable, but for any reason is temporarily unavailable, the server-FTP process SHOULD reply to the HOST command with a 421 reply and close the connection. In this particular situation, the server-FTP process MAY choose to keep the connection open in order to allow the user-PI an opportunity to choose another virtual host with a subsequent HOST command.

If the hostname specified is unknown at the server, or if the server is otherwise unwilling to treat the particular connection as a connection to the hostname specified, the server SHOULD respond with a 504 reply.

#### 3.2.1. REIN command semantics

As specified in [RFC0959], the REIN command returns the state of the connection to what it was immediately after the transport connection was opened. This specification makes no changes to that behavior. The effect of a HOST command MUST be reset if a REIN command is performed, and a new HOST command MUST be issued in order to connect to a virtual host.



### 3.2.2. User-PI usage of HOST

A user-PI that conforms to this specification MUST send the HOST command after opening the transport connection, or after any REIN command, before attempting to authenticate the user with the USER command. The following example illustrates what a typical login sequence might look like when the HOST command is used:

```
C> HOST ftp.example.com
S> 220 Host accepted
C> USER foo
S> 331 Password required
C> PASS bar
S> 230 User logged in
```

If a user-PI sends an additional HOST command before attempting to authenticate the user, a server-FTP process that conforms to this specification MUST treat the additional HOST command as though a REIN command was sent, and reset the user-PI to the state that existed after the TCP connection was first established. For example, if a user specifies the wrong virtual host by mistake, sending a subsequent HOST command will rectify the error. The following example illustrates what the login sequence might look like when the HOST command is sent twice before a user has been authenticated:

```
C> HOST foo.example.com
S> 220 Host accepted
C> HOST bar.example.com
S> 220 Host accepted
C> USER foo
S> 331 Password required
C> PASS bar
S> 230 User logged in
```

The HOST command can be used in combination with the ACCT command to differentiate between a user's various accounts on a specific virtual host. In this scenario, the user-PI sends a HOST command which the server-PI uses to route activity to the correct virtual host; the user-PI sends credentials using the USER and PASS commands which the server-PI validates; then, the user-PI sends an ACCT command to specify any additional account information for the server-PI implementation. The following example illustrates a sequential series of client commands that specify both a HOST and ACCT, with the server responses omitted for brevity:

```
C> HOST ftp.example.com
C> USER foo
C> PASS bar
C> ACCT project1
```

This is also true when the HOST command is used with the AUTH and ADAT commands that are discussed in [RFC2228] and [RFC4217]. In this scenario, the user-PI sends a HOST command which the server-PI uses to route activity to the correct virtual host, then the user-PI uses the AUTH and ADAT commands to negotiate the security mechanism and relevant authentication token(s) with the server-PI, then the user-PI sends user credentials using the USER and PASS commands which the server-PI validates. After which the user-PI MAY send an ACCT command to specify any additional account information for the server-PI implementation. The following example illustrates a sequential series of client commands that specify both a HOST and ACCT when used in conjunction with the security commands that are discussed in [RFC2228] and [RFC4217], with the server responses omitted for brevity:

```
C> HOST ftp.example.com
C> AUTH <mechanism-name>
C> ADAT <base64data>
C> USER foo
C> PASS bar
C> ACCT project1
```

### 3.2.3. State Diagrams

The state diagrams in this section illustrate typical sequences for command and reply interchange between the user-PI and server-PI. These diagrams are modeled on the similar diagrams in section 6 of [RFC0959].

In each diagram, the (B) "begin" state is assumed to occur after the transport connection has opened, or after a REIN command has succeeded. Other commands (such as FEAT [RFC2389]) that require no authentication may have intervened.

Additionally, a three-digit reply indicates a precise server reply code. A single digit on a reply path indicates any server reply that begins with that digit, except where a precise server reply code is defined on another path. For example, a single digit "5" will apply to "500", "501", "502", etc., when those reply codes are not expressly defined in the diagram. For each command there are three possible outcomes: success (S), failure (F), and error (E). In the state diagrams below we use the symbol B for "begin", and the symbol W for "wait for reply".

In each of these diagrams, a REIN command will return the diagram to the (B) "begin" state.

The state diagram in Figure 1 shows a typical sequence of flow of control when HOST is used with USER and PASS to log in to a particular FTP virtual host.

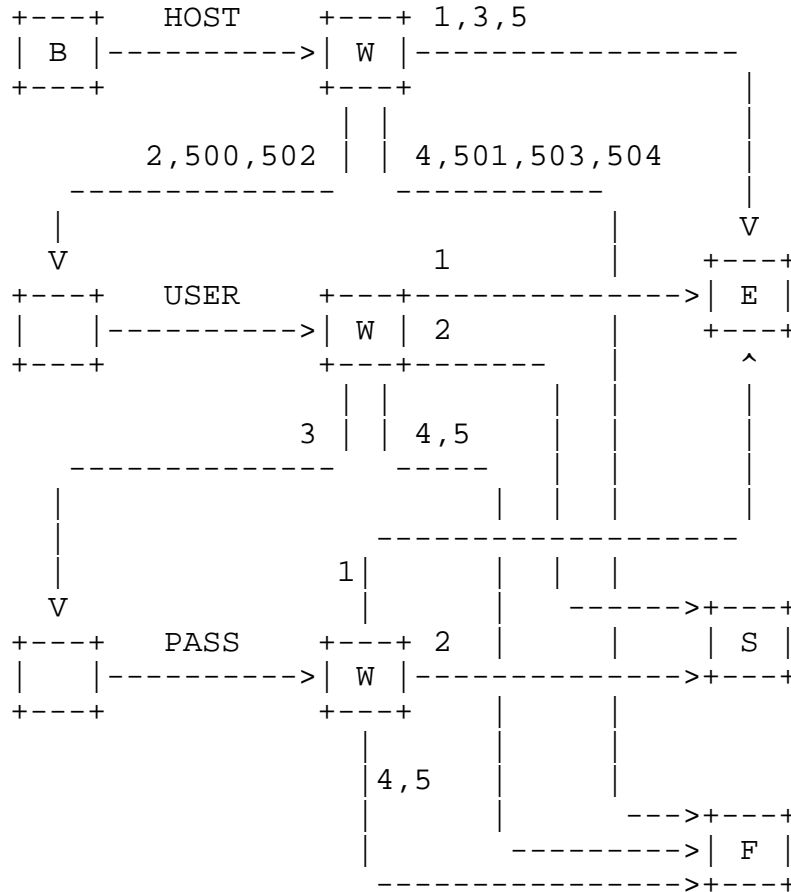


Figure 1: Typical login sequence with HOST command

The state diagram in Figure 2 shows the flow of control when a HOST command is sent after a user has already successfully logged in to a virtual host with USER and PASS.

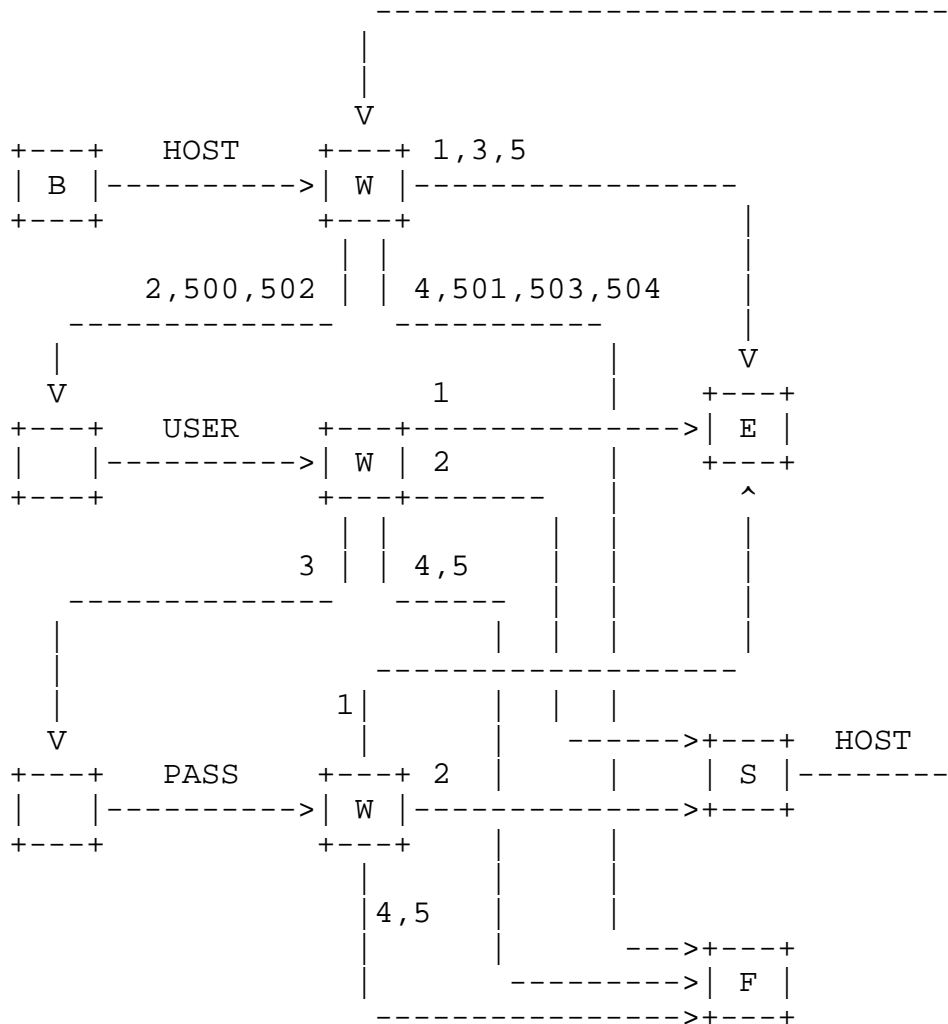


Figure 2: Login sequence with repeated HOST command

After a user has logged in, an additional account may be required by the server and specified by the client by using ACCT command. With this in mind, the state diagram in Figure 3 shows a typical sequence of flow of control when HOST is used with USER and PASS to log in to an FTP virtual host and ACCT is used to specify an account.

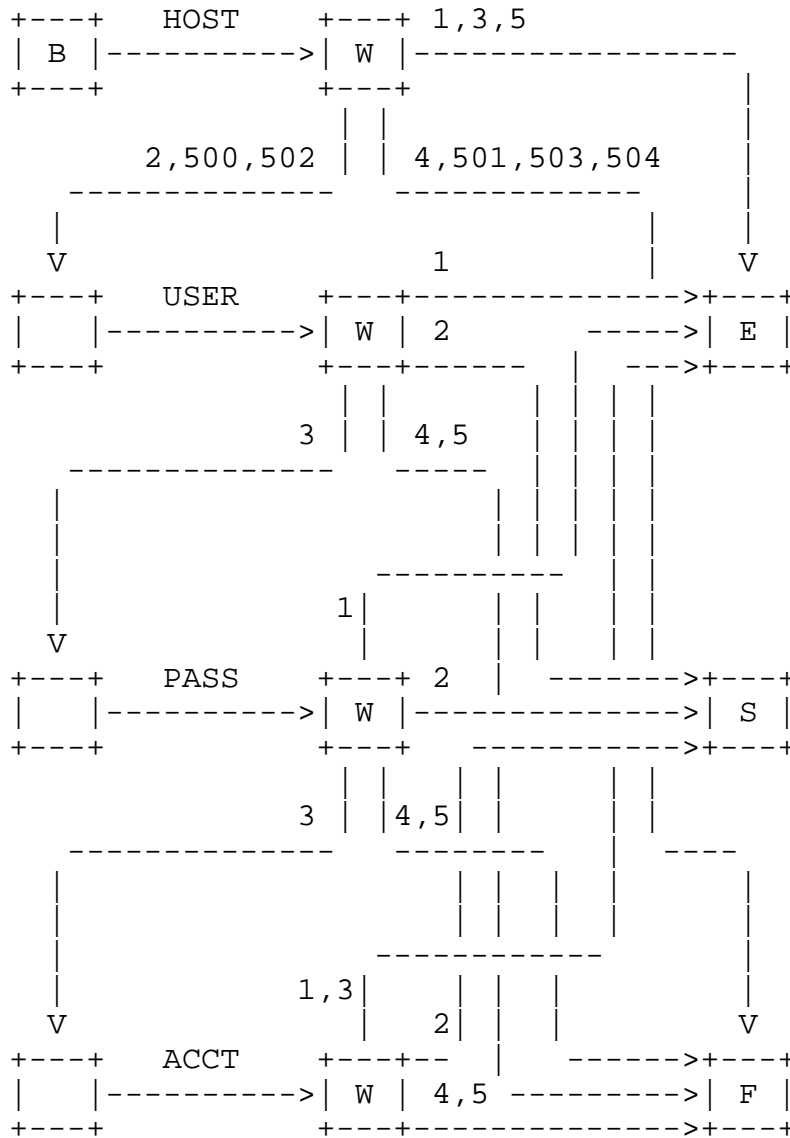


Figure 3: Login sequence with HOST and ACCT commands

When the HOST command is used in combination with the FTP security extensions that were introduced in [RFC2228], it SHOULD precede the security handshake. This allows both user-PI and server-FTP processes to map an FTP HOST to security data appropriately. The state diagram in Figure 4 shows a typical sequence of flow of control when HOST is used with the AUTH and ADAT commands that are discussed in [RFC2228].

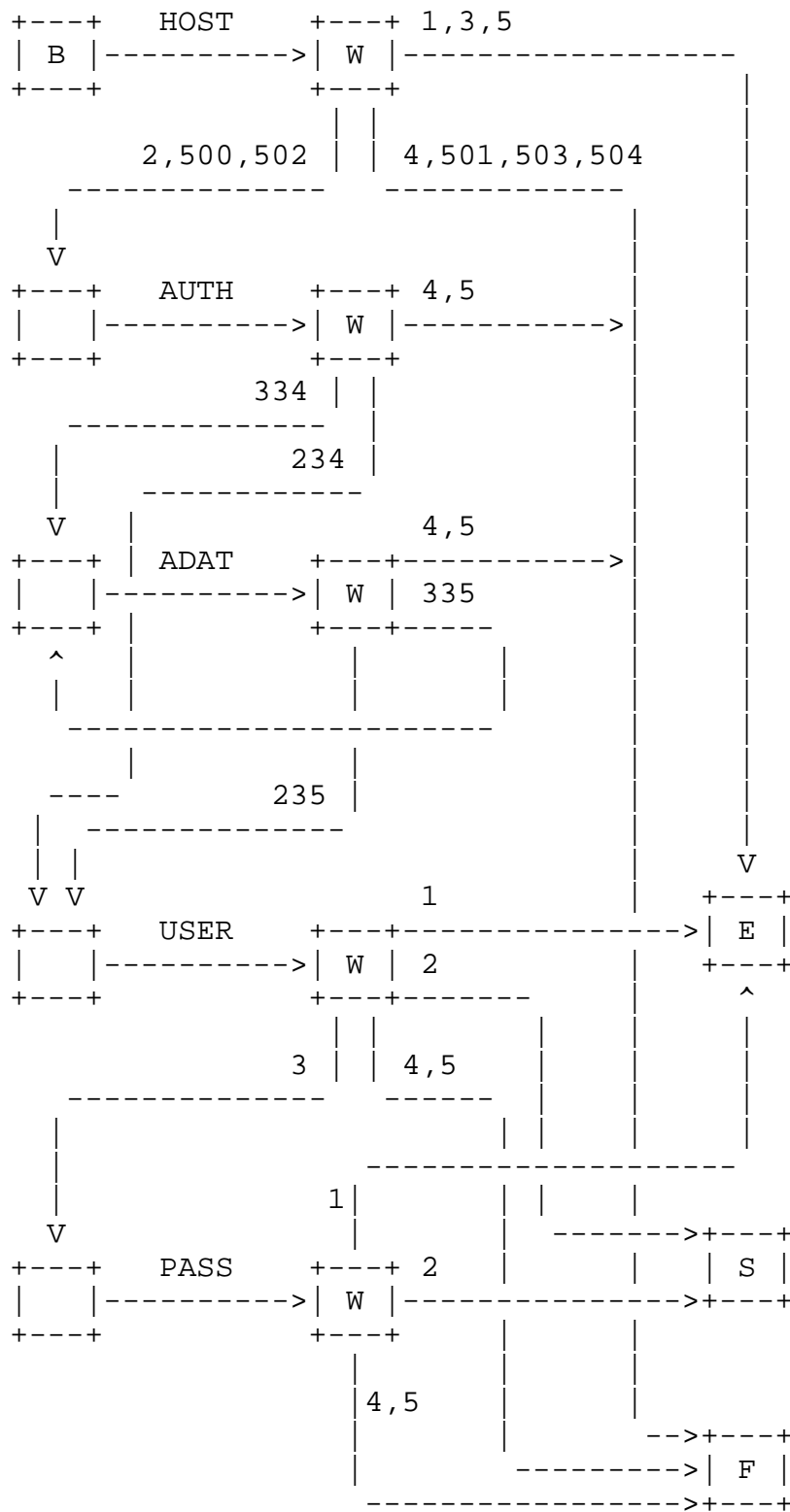
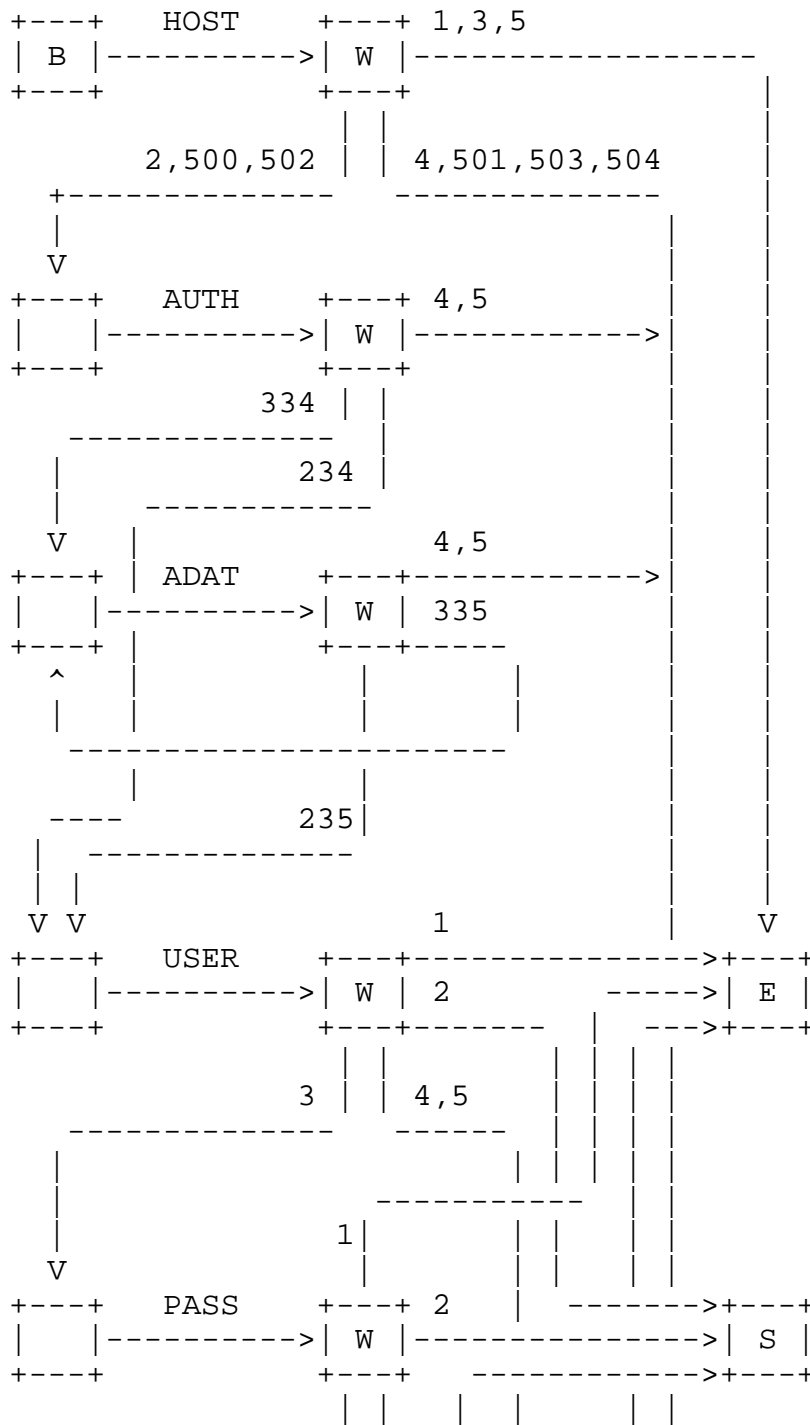


Figure 4: Login sequence with HOST and AUTH/ADAT commands

After a user has logged in with the security commands that are discussed in [RFC2228], an additional account may be required by the server and specified by the client by using ACCT command. The state diagram in Figure 5 shows a typical sequence of flow of control when HOST is used with the AUTH and ADAT commands to log in to an FTP virtual host and ACCT is used to specify an account.





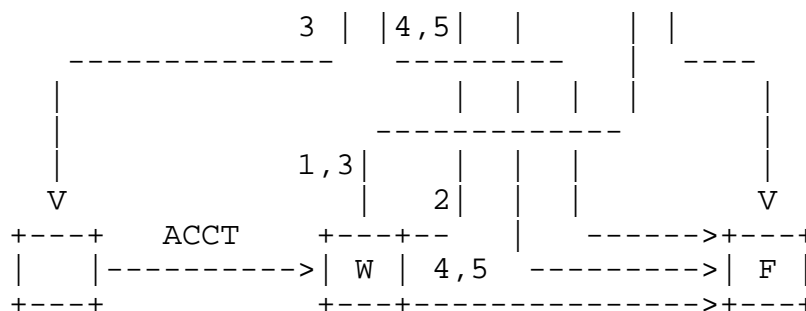


Figure 5: Login sequence with HOST and AUTH/ADAT/ACCT commands

### 3.3. HOST command errors

The server-PI SHOULD reply with a 500 or 502 reply if the HOST command is unrecognized or unimplemented.

As discussed in section 3 of this document, if a HOST command is sent after a user has been authenticated the server SHOULD do one of the following:

- a. Send a 503 reply for an invalid sequence of commands.
- b. Treat the HOST command as though a REIN command was sent and reset the user-PI to the state that existed after the previous HOST command was sent and before the user had been authenticated, and then return the appropriate reply for the HOST command.

A 501 reply SHOULD be sent if the hostname given is syntactically invalid, and a 504 reply SHOULD be sent if a syntactically valid hostname is not a valid virtual host name for the server. In all such cases, the server-FTP process MUST do one of the following:

- a. Ignore the HOST command and act as if a HOST command had not been sent. A user-FTP process MAY then send a subsequent HOST command with a different hostname.
- b. Close the connection.

A user-PI receiving a 500 or 502 reply to a HOST command SHOULD assume that the server-PI does not implement virtual servers by using the HOST command. The user-PI MAY then proceed to login as if the HOST command had not been sent.

A user-PI receiving an error reply that is different from the errors that have been described here SHOULD assume that the virtual HOST is unavailable, and terminate communications.

A server-PI that receives a USER command to begin the authentication sequence without having received a HOST command SHOULD NOT reject the USER command. Clients conforming to earlier FTP specifications do not send HOST commands. In this case the server MAY act as if some default virtual host had been explicitly selected, or MAY enter an environment that is different from that of any supported virtual hosts, perhaps one in which a union of all available accounts exists and which presents an NVFS that appears to contain subdirectories that contain the NVFS for all supported virtual hosts.

#### 3.4. FEAT response for HOST command

When replying to the FEAT command [RFC2389], a server-FTP process that supports the HOST command MUST include a line containing the single word "HOST". This word is case insensitive, and MAY be sent in any mixture of upper or lower case, however it SHOULD be sent in upper case. That is, the response SHOULD be:

```
C> FEAT
S> 211- <any descriptive text>
S> ...
S> HOST
S> ...
S> 211 End
```

The ellipses indicate place holders where other features may be included, and are not required. The one-space indentation of the feature lines is mandatory [RFC2389].

#### 4. Security Considerations

As discussed in section 3 of this document, a server implementation MUST treat a HOST command that was sent before a user has been authenticated as though a REIN command was sent, and a server implementation MAY treat a HOST command that was sent after a user has been authenticated as though a REIN command was sent. In either of these scenarios, the server implementation MUST reset the authentication environment, as that would allow for segregation between the security environments for each virtual host on an FTP server. The implementation details for security environments may vary greatly based on the requirements of each server implementation and operating system, and those details are outside the scope of the protocol itself. For example, a virtual host "foo.example.com" on an FTP server might use a specific username and password list, while the virtual host "bar.example.com" on the same FTP server might use a different username and password list. In such a scenario, resetting the security environment is necessary for the virtual servers to

appear to behave independently from a client perspective, while the actual server implementation details are irrelevant at the protocol level.

Section 15.1.1 of [RFC4217] discusses the use of X.509 certificates for server authentication. Taking the information from that document into account, when securing FTP sessions with the security mechanisms that are defined in [RFC4217], client implementations SHOULD verify that the hostname they specify in the parameter for the HOST command matches the identity that is specified in the server's X.509 certificate in order to prevent man-in-the-middle attacks.

A general discussion of issues related to the security of FTP can be found in [RFC2577].

## 5. IANA Considerations

IANA is requested to register the following FTP extension according to the procedure established by [RFC5797]:

cmd	FEAT Code	description	type	conf	RFC#s/References and Notes
HOST	HOST	Hostname	a	o	TBD

NOTE TO RFC EDITOR: Please update TBD in the above table with the number of this document.

## 6. References

### 6.1. Normative References

- [RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol (FTP)", STD 9, RFC 959, October 1985.
- [RFC1034] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, November 1987.
- [RFC1035] Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, November 1987.
- [RFC1123] Braden, R., "Requirements for Internet Hosts -- Application and Support", STD 3, RFC 1123, October 1989.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2228] Horowitz, M. and S. Lunt, "FTP Security Extensions", RFC 2228, October 1997.
- [RFC2389] Hethmon, P. and R. Elz, "Feature negotiation mechanism for the File Transfer Protocol", RFC 2389, August 1998.
- [RFC2640] Curtin, W., "Internationalization of the File Transfer Protocol", RFC 2640, July 1999.
- [RFC3492] Costello, A., "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492, March 2003.
- [RFC4217] Ford-Hutchinson, P., "Securing FTP with TLS", RFC 4217, October 2005.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 5234, January 2008.

## 6.2. Informative References

- [RFC1945] Berners-Lee, T., Fielding, R., and H. Frystyk, "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945, May 1996.
- [RFC2577] Allman, M. and S. Ostermann, "FTP Security Considerations", RFC 2577, May 1999.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [RFC5797] Klensin, J. and A. Hoenes, "FTP Command and Extension Registry", RFC 5797, March 2010.

## Appendix A. Unworkable Alternatives

Due to the level of scope for adding a new command to FTP, a brief discussion of suggested alternatives to a HOST command and their respective limitations is warranted. The suggested alternatives that are discussed in this appendix have been proposed in the past, but each of these ideas was deemed insufficient for the reasons that are listed within each section of the appendix.

### A.1. Overloading the CWD command

One suggested method to emulate a form of virtual hosts would be for the client to simply send a "CWD" command after connecting, using the virtual host name as the argument to the CWD command. This would allow the server-FTP process to implement the file stores of the virtual hosts as sub-directories in its NVFS. This suggestion is simple in concept, and most server-FTP implementations support this without requiring any code changes. While this method is simple to describe, and to implement, it suffers from several drawbacks:

- a. The "CWD" command is available only after the user-PI has authenticated itself to the server-FTP process. Thus, all virtual hosts would be required to share a common authentication scheme if they used this method.
- b. To make the virtual host truly transparent, either the server-FTP process needs to be modified to include information that shows the special nature of this first CWD command (negating most of the advantage of this scheme), or all users must see the same identical NVFS view upon connecting (they must connect in the same initial directory), or the NVFS must implement the full set of virtual host directories at each possible initial directory for any possible user.
- c. Unless the server is specially modified, a user connecting this way to a virtual host would be able to easily move to any other virtual host supported at the same server-FTP process, exposing the nature of the virtual host.

### A.2. Overloading the ACCT command

Another suggested method would be to simply overload the "ACCT" for FTP virtual hosts, but this proposal is unacceptable for several reasons with regard to when the ACCT command is sent during the request flow. Sections 5.4 and 6 of [RFC0959] document the request flow for a login sequence as USER -> PASS -> ACCT. This flow of commands may be acceptable when you are considering a single user having multiple accounts on an FTP server, but fails to differentiate between virtual hosts when you consider the following two issues:

- a. The first problem with overloading the ACCT command is certificate negotiation when using the FTP security extensions that are documented in [RFC2228] and [RFC4217]. In order to safeguard user credentials, security mechanism and certificate negotiation must occur before login credentials are sent by the client. The problem with using the ACCT command in this scenario is that there is no way of ensuring that the certificate matches

the correct virtual host before the user credentials are sent.

- b. The second problem with overloading the ACCT command is how user credentials are implemented for FTP virtual hosts. FTP server implementations may allow the use of custom user credentials on a per-virtual-host basis. For example, in one particular implementation the virtual host negotiation occurs, and then the user credentials are looked up using the account mechanism that is specific to that virtual host. So once again the virtual host negotiation must take place before the user credentials are sent.

### A.3. Overloading the USER command

An additional suggestion would be to overload well-known syntax through the existing USER command, as illustrated in the following example:

```
C> USER foo@example.com
S> 331 Password required
C> PASS bar
S> 230 User logged in
```

In this example, the user "foo" might be attempting to log on to the virtual host "example.com" on an FTP server. This suggestion may seem plausible at first, but introduces several implementation problems. For example:

- a. Some network environments already use the "username@hostname" syntax for network credentials, where the "hostname" portion refers to the location of the user's credentials within the network hierarchy. Using the "foo@example.com" syntax it becomes difficult to differentiate between the user "foo" logging into a virtual host named "example.com" on an FTP server versus the user "foo@example.com" logging into an FTP server with no specified virtual host.
- b. When using the FTP security extensions that are documented in [RFC2228] and [RFC4217], security mechanism and certificate negotiation must occur before login credentials are sent by the client. More specifically, the AUTH/ADAT commands must be sent before the USER command in order to safeguard user credentials. If you overload the USER command, there is no way of ensuring that the certificate matches the correct virtual host before the user credentials are sent by the client.

#### A.4. Conclusion

The conclusion from the examination of the existing possibilities seems to be that in order to obtain an adequate emulation of "real" FTP servers, client and server modifications to support virtual hosts are necessary. Therefore a new FTP command seems the most likely solution to provide the required level of support.

#### Appendix B. Acknowledgements

Robert Elz and Paul Hethmon provided a detailed discussion of the HOST command in their Internet draft titled "Extensions to FTP" as part of their work with the FTPEXT Working Group at the IETF. Their work formed the basis for much of this document, and their help has been greatly appreciated. They would also like to credit Bernhard Rosenkraenzer for having first suggested and described the HOST command.

Alexey Melnikov, Alfred Hoenes, John Klensin, and Joe Touch have made several suggestions about earlier versions of this document; many of their suggestions have been incorporated, and their contributions are gratefully acknowledged. In addition, Alec Rowell's assistance in making sections of this document more readable was invaluable.

#### Authors' Addresses

Paul Hethmon  
Hethmon Brothers  
2305 Chukar Road  
Knoxville, TN 37923  
USA

Email: [phethmon@hethmon.com](mailto:phethmon@hethmon.com)

Robert McMurray  
Microsoft Corporation  
One Microsoft Way  
Redmond, WA 98052  
USA

Email: [robmcm@microsoft.com](mailto:robmcm@microsoft.com)