Network Working Group
Request for Comments: 1572
Category: Standards Track

S. Alexander, Editor
Lachman Technology, Inc.

January 1994

Telnet Environment Option

Status of this Memo

Abstract

This document specifies a mechanism for passing environment information between a telnet client and server. Use of this mechanism enables a telnet user to propagate configuration information to a remote host when connecting.

This document corrects some errors in [1].

1. Command Names and Codes

NEW-ENVIRON	39
IS	0
SEND	1
INFO	2
VAR	0
VALUE	1
ESC	2
USERVAR	3

2. Command Meanings

IAC WILL NEW-ENVIRON

The sender of this command is willing to send environment variables.

IAC WONT NEW-ENVIRON

The sender of this command refuses to send environment variables.

IAC DO NEW-ENVIRON

The sender of this command is willing to receive environment variables.

IAC DONT NEW-ENVIRON

The sender of this command refuses to accept environment variables.

IAC SB NEW-ENVIRON SEND [type ... [type ... [...]]] IAC SE

The sender of this command requests that the remote side send its environment variables. The "type" may be either VAR or USERVAR, to indicate either well known or user variable names. Only the side that is DO NEW-ENVIRON may initiate a SEND command. If a list of variables is specified, then only those variables should be sent. If no list is specified, then the default environment, of both well known and user defined variables, should be sent. If one of the variables has no name, then all the variables of that type (well known or user defined) in the default environment should be sent.

IAC SB NEW-ENVIRON IS type ... [VALUE ...] [type ... [VALUE ...] [...]] IAC SE

The sender of this command is sending environment variables. This command is sent in response to a SEND request. Only the side that is WILL NEW-ENVIRON may send an IS command. The "type"/VALUE pairs must be returned in the same order as the SEND request specified them, and there must be a response for each "type ..." explicitly requested. The "type" will be VAR or USERVAR. Multiple environment variables may be sent. The characters following a "type" up to the next "type" or VALUE specify the variable name. The characters following a VALUE up to the next "type" specify the value of the variable. If a "type" is not followed by a VALUE (e.g., by another VAR, USERVAR, or IAC SE) then that variable is undefined. If a VALUE is immediately followed by a "type" or IAC, then the variable is defined, but has no value. If an IAC is contained between the IS and the IAC SE, it must be sent as IAC IAC. If a variable or a value contains a VAR, it must be sent as ESC VAR. If a variable or a value contains a USERVAR, it must be sent as ESC USERVAR. If a variable or a value contains a VALUE, it must be sent as ESC VALUE. If a variable or a value contains an ESC, it must be sent as ESC ESC.

IAC SB NEW-ENVIRON INFO type ... [VALUE ...] [type ... [VALUE ...] [...]] IAC SE

The sender of this command is sending information about environment variables that have changed. It is identical to the IS command, except that the command is INFO instead of IS. Only the side that is WILL NEW-ENVIRON may send an INFO command. The INFO command is not to be used to send initial information; the SEND/IS sequence is to be used for that. The INFO command is to be used to propagate changes in environment variables, and may be spontaneously generated.

3. Default Specification

The default specification for this option is

```
WONT NEW-ENVIRON
DONT NEW-ENVIRON
```

meaning there will not be any exchange of environment information.

4. Motivation

Many operating systems have startup information and environment variables that contain information that should be propagated to remote machines when Telnet connections are established. Rather than create a new Telnet option each time someone comes up with some new information that they need propagated through a Telnet session, but that the Telnet session itself doesn't really need to know about, this generic information option can be used.

5. Well Known Variables

USER This variable is used to transmit the user or account name that the client wishes to log into on the remote system. The format of the value the USER variable is system dependent, as determined by the remote system.

JOB This variable is used to transmit the job ID that the client wishes to use when logging into the remote system. The format of the value the JOB variable is system dependent, as determined by the remote system.

ACCT This variable is used to transmit the account ID that the client wishes to use when logging into the remote system. The format of the value the ACCT variable is system dependent, as determined by the remote system.

PRINTER This variable is used to identify the default location for printer output. Because there does not currently exist a standard way of naming a printer on a network, the format of this variable is currently undefined.

SYSTEMTYPE This is used to transmit the type of operating system on the system that sends this variable. It value is identical to the value of the SYSTEM (SYST) command in FTP [4]. The format of the value shall have as its first word one of the system names listed in the current version of the Assigned Numbers document [5].

DISPLAY This variable is used to transmit the X display location of the client. The format for the value of the DISPLAY variable is:

<host>:<dispnum>[.<screennum>]

This information is identical to the information passed using the Telnet X-DISPLAY-LOCATION option. If both the DISPLAY environment variable, and the X-DISPLAY-LOCATION option [6] are received, and they contain conflicting information, the most recently received information received should be used.

Because it is impossible to anticipate all variables that users may wish to exchange, the USERVAR type is provided to allow users to transmit arbitrary variable/value pairs. The use of an additional type allows implementations to distinguish between values derived by the remote host software and values supplied by the user. Paranoid implementations will most likely treat both types with an equal level of distrust. The results of a name-space collision between a well-known and a user variable are implementation specific.

6. Implementation Rules

WILL and DO are used only at the beginning of the connection to obtain and grant permission for future negotiations.

Once the two hosts have exchanged a WILL and a DO, the sender of the DO NEW-ENVIRON is free to request that environment variables be sent. Only the sender of the DO may send requests (IAC SB NEW-ENVIRON SEND IAC SE) and only the sender of the WILL may transmit actual environment information (via the IAC SB NEW-ENVIRON IS ... IAC SE command). Though this option may be used at any time throughout the life of the telnet connection, the exchange of environment information will usually happen at the startup of the connection. This is because many operating systems only have mechanisms for

propagating environment information at process creation, so the information is needed before the user logs in.

The receiving host is not required to put all variables that it receives into the environment. For example, if the client should send across USERVAR "TERM" VALUE "xterm" as an environment variable, and the TERMINAL-TYPE [3] option has already been used to determine the terminal type, the server may safely ignore the TERM variable. Also, some startup information may be used in other ways; for example, the values for "USER", "ACCT" and "PROJ" values might be used to decide which account to log into, and might never be put into the users environment. In general, if the server has already determined the value of an environment variable by some more accurate means, or if it does not understand a variable name, it may ignore the value sent in the NEW-ENVIRON option. The server may also prefer to just put all unknown information into the users environment. This is the suggested method of implementation, because it allows the user the most flexibility.

The following is an example of use of the option:

```
Host1                                Host2
IAC DO NEW-ENVIRON
                                     IAC WILL NEW-ENVIRON
[ Host1 is now free to request environment information ]
IAC SB NEW-ENVIRON SEND VAR
"USER" VAR "ACCT" VAR USERVAR
IAC SE
[ The server has now explicitly asked for the USER and ACCT
  variables, the default set of well known environment variables,
  and the default set of user defined variables.  Note that the
  client includes the USER information twice; once because it was
  explicitly asked for, and once because it is part of the
  default environment.  ]
                                     IAC SB NEW-ENVIRON IS VAR "USER"
                                     VALUE "joe" VAR "ACCT" VALUE
                                     "kernel" VAR "USER" VALUE "joe"
                                     VAR "DISPLAY" VALUE "foo:0.0"
                                     USERVAR "SHELL" VALUE "/bin/csh"
                                     IAC SE
```

It is legal for a client to respond with an empty environment (no data between the IAC SB and IAC SE) when no well-defined or user variables are currently defined. For example:

```
IAC SB NEW-ENVIRON IS IAC SE
```

is a valid response to any of the following:

```
IAC SB NEW-ENVIRON SEND IAC SE
IAC SB NEW-ENVIRON SEND VAR IAC SE
IAC SB NEW-ENVIRON SEND USERVAR IAC SE
IAC SB NEW-ENVIRON SEND VAR USERVAR IAC SE
```

(The last example is equivalent to the first...)

The earlier version of this specification [1] incorrectly reversed the values for VAR and VALUE, which put the specification at odds with existing implementations. In order to resolve that problem, as well as other minor problems, a new option number has been assigned to the NEW-ENVIRON option. This allows implementations of this memo to interoperate with no ambiguity.

For a discussion on how to implement to interoperate with the various implementations that pre-date this memo, see [2].

It is expected that any implementation that supports the Telnet NEW-ENVIRON option will support all of this specification.

7. Security Concerns

It is important for an implementor of the NEW-ENVIRON option to understand the interaction of setting options and the login/authentication process. Specifically careful analysis should be done to determine which variables are "safe" to set prior to having the client login. An example of a bad choice would be permitting a variable to be changed that allows an intruder to circumvent or compromise the login/authentication program itself.

8. References

[1] Borman, D., Editor, "Telnet Environment Option", RFC 1408, Cray Research, Inc., January 1993.

[2] Borman, D., "Telnet Environment Option Interoperability Issues", RFC 1571, Cray Research, Inc., January 1994.

[3] VanBokkelen, J., "Telnet Terminal-Type Option", RFC 1091, FTP Software, Inc., February 1989.

[4] Postel, J., and J. Reynolds, "File Transfer Protocol (FTP)", STD 9, RFC 959, USC/Information Sciences Institute, October 1985.

[5] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1340, USC/Information Sciences Institute, July 1992.

[6] Marcy, G., "Telnet X Display Location Option", RFC 1096, Carnegie Mellon University, March 1989.

Acknowledgements

The original version of this document was written by Dave Borman of Cray Research, Inc. In addition, the comments of the Telnet Working Group of the IETF are gratefully acknowledged.

Security Considerations

Security issues are discussed in Section 7.

Editor's Address

Steve Alexander
Lachman Technology, Inc.
1901 North Naper Boulevard
Naperville, IL 60563-8895

Phone: (708) 505-9555 x256
EMail: stevea@lachman.com