Network Working Group Mark Crispin
Request for Comments 727 MIT-AI
NIC 40025 27 April 1977

TELNET Logout Option

1. Command name and code.

LOGOUT 18

2. Command meanings.

IAC WILL LOGOUT

The sender of this command REQUESTS permission to, or confirms that it will, forcibly log off the user process at its end.

IAC WON'T LOGOUT

The sender of this command REFUSES to forcibly log off the user process at its end.

IAC DO LOGOUT

The sender of this command REQUESTS that the receiver forcibly log off the user process at the receiver's end, or confirms that the receiver has its permission to do so.

IAC DON'T LOGOUT

The sender of this command DEMANDS that the receiver not forcibly log off the user process at the receiver's end.

3. Default.

WON'T LOGOUT

DON'T LOGOUT

i.e., no forcible logging off of the server's user process.

4. Motivation for the option.

Often, a runaway user process could be hung in such a state that it cannot be interrupted by normal means. Conversely, the system itself could be bottlenecked so that response delays are intolerable. A user (human or otherwise) eventually will time out out of frustration

and take the drastic means of closing the connection to free itself from the hung process. In some situations, even the simple operation of logging out can take a long time.

Some systems treat a close to mean that it should log out its user process under it. However, many hosts merely "detach" the process so that an accidental close due to a user or temporary hardware error will not cause all work done on that job to be lost; when the connection is re-established, the user may "attach" back to its process. While this protection is often valuable, if the user is giving up completely on the host, it can cause this hung job to continue to load the system.

This option allows a process to instruct the server that the user process at the server's end should be forcibly logged out instead of detached. A secondary usage of this option might be for a server to warn of impending auto-logout of its user process due to inactivity.

5. Description of the option.

When a user decides that it no longer wants its process on the server host and decides that it does not want to wait until the host's normal log out protocol has been gone through, it sends IAC DO LOGOUT. The receiver of the command may respond with IAC WILL LOGOUT, in which case it will then forcibly log off the user process at its end. If it responds with IAC WON'T LOGOUT, then it indicates that it has not logged off the user process at its end, and if the connection is broken, the process very possibly will be detached.

A truly impatient user that feels that it must break away from the server immediately could even send IAC DO LOGOUT and then close. At the worst, the server would only ignore the request and detach the user process. A server that implements the LOGOUT option should know to log out the user process despite the sudden close and even an inability to confirm the LOGOUT request!

6. A sample implementation of the option.

The server implements the LOGOUT option both for accepting LOGOUT requests and for auto-logout warning.

Case 1:

The user connects to the server, and starts interacting with the server. For some reason, the user wishes to terminate interaction with the server, and is reluctant to go through the normal log out procedure, or perhaps the user is unable to go through the normal

log out procedure. It does not want the process at the server any more, so it sends IAC DO LOGOUT. The server verifies the request with IAC WILL LOGOUT, and then forcibly logs off the user process (perhaps by using a system call that causes another process to be logged out). It does not have to close the connection unless the user closes or it wants to close. Neither does it wait until the user has received its confirmation--it starts the log out immediately so if the user has in the mean time closed the connection without waiting for confirmation, its logout request still is performed.

Case 2:

The user connects to the server, and after logging in, is idle for a while, long enough to approach the server's autologout time. The server shortly before the autologout sends IAC WILL LOGOUT; the user sees this and sends IAC DON'T LOGOUT, and continues work on the host. Nothing prevents the server from logging out the user process if inactivity continues; this can be used to prevent a malicious user from locking up a process on the server host by the simple expedient of sending IAC DON'T LOGOUT every time it sees IAC WILL LOGOUT but doing nothing else.