Safe Handling of Passwords

Some operations and functions support the use of a username and password when making a network connection. If you use sensitive passwords you must take certain precautions to prevent them from being accidentally revealed.

1. Always use the /V=0 flag when using a username or password with the /U (username) and /W (password) flags or the /AUTH flag. Otherwise, the debugging information that is printed to the history area will contain those values and anyone who sees the experiment could see them.
2. Do not hard code username or password values into procedures, since anyone with access to the procedure file could read them.
3. Do not store username or password values in global variables. Since global variables are saved with an experiment, if someone else had access to your experiment they could see this information.

Here is an example of how a username and sensitive password can be used in a secure manner:

```
Function SafeLogin()
   String username = ""
   String password = ""
   Prompt username, "Username"
   Prompt password, "Password"
   DoPrompt "Enter username and password", username, password
   if (V_flag == 1)
      // User hit cancel button, so do nothing.
      return 0
   endif

   // Percent-encode in case username and password contain reserved characters.
   String encodedUser = URLEncode(username)
   String encodedPass = URLEncode(password)

   String theURL
   sprintf theURL, "http://%s:%s@www.example.com", encodedUser, encodedPass
   String response = FetchURL(theURL)
   // NOTE: For FTP operations and URLRequest, make sure to use /V=0 so that
   // the username and password are not printed to the history.

   return 0
End
```

Note that the user is prompted to provide the username and password when the function is called and that only local string variables are used to store the username and password. The values in those string variables are not stored once the function is done executing.

Note also that the password is not hidden during entry in the dialog. Igor currently does not provide a way to do this.

Network Timeouts and Aborts

Some network calls may return an error code to Igor if they timeout. Depending on the specific operation or function, there can be a number of causes for a timeout.

If a network connection cannot be made after a period of time it will timeout. The amount of time allowed for a connection to be established is dependent on several factors.

You can always abort a network operation or function by pressing the **User Abort Key Combinations**.