**About DLL tutorial:**

The DLL files should be present in the same folder as the GoldSim.gsm file, in the same folder as the GoldSim executable file, or elsewhere in the user’s path.

Every external function is called by GoldSim with specific requests. The requests include initialization, returning the function version number, performing a normal calculation, and "cleaning up" after a simulation. The function name and argument list (the set of input and output data for the function) are specified by the GoldSim user when setting up the External element.

External functions should provide their own error handling, message handling, file management and memory management if required. It is essential that when it receives a “clean up” request, an external function should release any dynamically acquired memory and close any open files.

* Data are passed from GoldSim to the external function and back again to GoldSim via arrays of double precision floating point input and output arguments.
* GoldSim calculates the total number of input and output arguments by summing over the inputs and outputs specified on the Interface tab of the External element properties dialog. while array inputs or outputs count as the size of the array (rows \* columns)
* The input arguments and output arguments are transferred between GoldSim and the external function in exactly the order in which they are listed on the Interface tab.
* If the external function will be returning definitions for one or more Tables or Time Series (see below), GoldSim will not know in advance how long the Table definitions will be. In this case, the external function should specify a value for outargs[1] greater than or equal to the actual total number of arguments that may be returned. GoldSim will allocate this amount of memory for outargs. Note that this can be reset during the simulation by returning XFState=-2.
* **The Input and Output Argument Arrays page 1011**
* The external function is called at the following times:
  + At the beginning of the simulation (Etime = 0);
  + Every GoldSim timestep (if one of the inputs to the element has changed); and
  + At any internal step in which an Event is triggered (if one of the inputs to the element has changed).
* GoldSim calls Fortran DLL
* Fortran DLL calls VB6 executable
* VB6 executable runs the DLL to create MyTitles file if needed
* VB6 executable Reads MyTitles file and extracts outputs