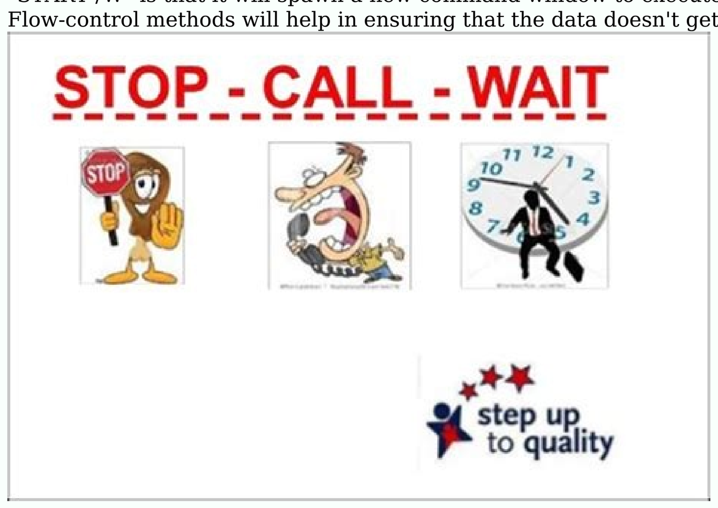
Stop call wait process

Stop call wait procedure. How do you stop call waiting. Stop call wait rule. What is stop call wait.

The Six Big Losses of OEELineView Academy (was OFX Academy) I want to repeatedly execute a program in a loop. Sometimes, the program crashes, so I want to kill it so the next iteration can correctly start. I determine this via timeout. I have the timeout working but cannot get the Exit Code of the program, which I also need to determine its result. Before, I did not wait with timeout, but just used -wait in Start-Process, but this made the script hang if the started program crashed. With this setup I could correctly get the exit code though. I am executing from ISE. for ($i=0; $i -le $max_iterations; $i++) { $proc = Start-Process -filePath $programtorun -ArgumentList $argumentlist -workingdirectory $programtorunpath -PassThru # wait up to x seconds for normal termination Wait-Process -Timeout 300 -Name $programname # if not exited, kill process if(!$proc.hasExited) { echo "kill the process" #$proc.Kill() <- not working if proc is crashed Start-Process -filePath "taskkill.exe" -Wait -ArgumentList '/F', '/IM', $fullprogramname } # this is where I want to use exit code but it comes in empty if ($proc.ExitCode -ne 0) { # update internal error counters based on result } } How can I Start a process Wait for it to orderly execute and finish Kill it if it is crashed (e. g. hits timeout) get exit code of process I liked the "START /W" answer, though for my situation I found something even more basic. My processes were console applications. And in my ignorance I thought I would need something special in BAT syntax to make sure that the 1st one completed before the 2nd one started. However BAT appears to make a distinction between console apps and windows apps, and it executes them a little differently.
The OP shows that window apps will get launched as an asynchronous call from BAT. But for console apps, that are invoked synchronously, inside the same command window as the BAT itself is running in. For me it was actually better not to use "START /W", because everything could run inside one command window. The annoying thing about "START /W" is that it will spawn a new command window to execute your console application in. While sending the data from the sender to the receiver the flow of data needs to be controlled. Suppose a situation where the sender is sending the data at a rate higher than the receiver is able to receive and process it, then the data will get lost. The Flow-control methods will help in ensuring that the data doesn't get lost. The flow control method will keep a check that the senders send the data only at a rate that the receiver is able to receive and process.



There are mainly two ways in which this can be achieved i.e. using Stop-and-wait protocol or sliding window protocol. In this blog, we are going to learn about the Stop-and-wait protocol. So, let's get started. It is the simplest flow control method. In this, the sender will send one frame at a time to the receiver. The sender will stop and wait for the acknowledgment from the receiver. This time(i.e. the time between message sending and acknowledgement receiving) is the waiting time for the sender and the sender is totally idle during this time. When the sender gets the acknowledgment(ACK), then it will send the next data packet to the receiver and wait for the acknowledgment again and this process will continue as long as the sender has the data to send. This can be understood by the diagram below: The above diagram explains the normal operation in a stop-and-wait protocol. Now, we will see some situations where the data or acknowledgment is lost and how the stop-and-wait protocol responds to it. Situation 1 Suppose if any frame sent is not received by the receiver and is lost. So the receiver will not send any acknowledgment as it has not received any frame. Also, the sender will not send the next frame as it will wait for the acknowledgment for the previous frame which it had sent. So a deadlock situation arises here. To avoid any such situation there is a time-out timer. The sender waits for this fixed amount of time for the acknowledgment and if the acknowledgment is not received then it will send the frame again.

Situation 2 Consider a situation where the receiver has received the data and sent the acknowledgment but the ACK is lost. So, again the sender might wait till infinite time if there is no system of time-out timer. So, in this case also, the time-out timer will be used and the sender will wait for a fixed amount of time for the acknowledgment and then send the frame again if the acknowledgement is not received. There are two types of delays while sending these frames: Transmission Delay: Time taken by the sender to send all the bits of the frame onto the wire is called transmission delay. This is calculated by dividing the data size(D) which has to be sent by the bandwidth(B) of the link. Td = D / B Propagation Delay: Time taken by the last bit of the frame to reach from one side to the other side is called propagation delay. It is calculated by dividing the distance between the sender and receiver by the wave propagation speed. Tp = d / s ; where d = distance between sender and receiver, s = wave propagation speed The propagation delay for sending the data frame and the acknowledgment frame is the same as distance and speed will remain the same for both frames. Hence, the total time required to send a frame is: Total time= Td(Transmission Delay) + Tp(Propagation Delay for data frame) + Tp(Propagation Delay for acknowledgment frame) Total time=Td+ 2Tp The sender is doing work only for Td time (useful time)and for the rest 2Tp time the sender is waiting for the acknowledgment. Efficiency = Useful Time/ Total Time η = Td / (Td+2*Tp) η = 1/(1+2a) →(1) where a=Tp / Td The number of bits that a receiver can accept in total time duration (i.e. transmission time(Td) + 2 * propagation delay(Tp)). It is also called effective bandwidth or bandwidth utilization. In Stop and Wait, in the total duration, the receiver can accept only one frame.
One frame is of data size D i.e. D bits in one frame. Therefore, Throughput= D / (Td + 2Tp) Throughput = D / Td(1+2a) →(2) where a= Tp / Td From the definition of Transmission delay, Td=D/B Cross multiplying B and Td, we get B = D/Td → (3) Now putting the value of equation 3 in equation 2, we get, Throughput= B /(1+2a) → (4) Now, putting the value of equation 1 in equation 4, we get, Throughput= η * B It is very simple to implement. The main advantage of this protocol is the accuracy. The next frame is sent only when the first frame is acknowledged. So, there is no chance of any frame being lost. We can send only one packet at a time. If the distance between the sender and the receiver is large then the propagation delay would be more than the transmission delay. Hence, efficiency would become very low. After every transmission, the sender has to wait for the acknowledgment and this time will increase the total transmission time. This makes the transmission process slow . This is how the flow of data is controlled using the stop-and-wait protocol. Hope you learned something new today. Do share this blog with your friends to spread the knowledge. Visit our YouTube channel for more content. You can read more blogs from here . Keep Learning :) Team AfterAcademy! Understanding some of the more common terms in the lean vocabulary will be helpful during your implementation. We mention many of them throughout this website, so this is a good page to review if you want a better understanding of these terms. Some of them may be defined in greater detail on other webpages, so think of this page as the glossary of terms.Although this website is dedicated to lean manufacturing, the lean vocabulary, as you may or may not know, is somewhat synonomous and/or linked with the TPS vocabulary in many ways. As my background is in TPS, many of the terms on this page (and within this site) will have the TPS term used to describe a concept.Glossary of TermsAndon:A system that alerts all associates (from floor members to management) of an abnormal situation (production line down, behind schedule). Usually a light or an electronic board.FIFO (First In, First Out):Product is consumed in the order it was used produced or received. First product in the building is the first product to be used.Fluctuation Stock:A set amount of stock that is introduced or removed as required to maintain level production volumes.Heijunka (Load Smoothing Production):A system designed to levelize production requirements through kanban control to ensure the same number of pieces are produced each day and/or shift. Fluctuation stock absorbs the variation.Jidoka:A machines ability to detect abnormalities and stop the process. Operators have the same authority.Just-in-time Manufacturing:One of the more common terms in the lean vocabulary, this term refers to the production or conveyance of parts or material only when they are needed and in the quantity required.Kanban:An instruction for production or conveyance. The most common form of a kanban is a hand sized signboard however, there are other methods used as kanbans (click on the "Kanban" button on the nav bar for more info).Kaizen (Continuous Improvement):One of the key terms in the lean vocabulary, this refers to the need to continuously improve upon current processes.

Continuous Improvement does not stop until all waste is eliminated.Non-Value-Added:Any work within the company that does not add value to the finihed product. Much of the office work within an organization, although necessary, would be considered non-value added.Poka Yoke:This Japanese term refers to mistake proofing devices. These devices are often built into or interlocked with machinery to prevent missing an operation.Pull System:Each process pulls product from a previous process. The pull of product is the signal for the preceeding process to replenish the parts that were pulled. This is also referred to as a replenishment system.Standard-in-process stock:The amount of product kept between operations/stations/processes in order to maintain proper process flow.Standardized Work:A carefully documented and balanced work process that must be adhered to by each operator to ensure consistency.Stop-Call-Wait:This concept refers to an operators responsibility to "Stop" a process when something abnormal (defective) occurs, "Call" for support and "Wait" for the support to arrive before proceeding.TPS (Toyota Production System):A production system developed by Toyota (during difficult finacial times for them) to facilitate small lot production in an economically feasible manner. It focuses on continuous improvement of processes, elimination of waste, level production and quality built into the process (among other principles). It is the basis for lean manufacturing.Value-Added:A common term in the lean vocabulary referring to any necessary work that adds value to a product. Most production work would be considered value-added work as it contributes directly to the finished product.Visual Control:This term refers to management by sight. Processes and measurables are set up so they can be understood at a glance. 5S is a type of visual control – a place for everything and everything in it's place.These are some of the more common terms used as part of the lean vocabulary (or TPS vocabulary). If there are any other terms that you would like clarification on, please use the "Contact Us" button on the nav bar to the left.