Waiting Best Practices

* In general, we return true for success (the condition has been met) and false to indicate that the condition has not been met and we ran out of time.
* if msTimeout is negative, and condition is initially met, return true (success).
* if msTimeout is negative, and the condition has not yet been met, return false to to indicate a timeout.
* even if we've run over the time limit, if the condition has been met, return true (success).
* if msTimeout is zero, treat as a "never timeout".
* if you have a method that never times out, call the one that does passing in zero.

public boolean waitUntilZero(long msTimeout) throws InterruptedException {

synchronized (lockObject) {

if (value == 0) return true; // condition has been met

if (msTimeout == 0L) {

while (value != 0) lockObject.**wait()**;

return true; // condition has been met

} else {

long msEndTime = System.currentTimeMillis() + msTimeout;

long msRemaining = msTimeout;

while (msRemaining >= 1L) {

lockObject.**wait(msRemaining)**;

if (value == 0) return true; // condition has been met

msRemaining = msEndTime - System.currentTimeMillis();

}

return false; // timed out

}

}

}

public void waitUntilZero() throws InterruptedException {

**waitUntilZero(0L);**

}

xxx