Hi Factor,

This is actually not related to Value at all. If you try the same with Vector<int>, you will notice the same behavior.

I think it is caused by the way Vector allocates memory when it grows. If you do v.SetCount(ITEM_COUNT,0), instead of series of Add() calls and then call Clear() and Shrink, you will get back at the memory usage you started with.

I am not sure if this is a bug. I have a strange feeling that this might be actually result of how OS manages the memory. If you ask it for a big chunk at once and then return it later, it might work different than when you are asking for chunks of size increasing as powers of two (which is what vector does).

Maybe someone knowing the memory handling better than me might shed some more light into this. But from what I've seen, I would recommend to use SetCount() every time you know how many items will be necessary.

Best regards,
Honza