
Subject: Re: ASSERT when using ValueMap
Posted by [NilaT](#) on Tue, 07 Mar 2017 10:54:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

cbpporterLong story short, Value/ValueMap is not debug-able. Period.
mirek it is simply the fact that Value code is immensely complicated. Really, absolutely totally fucked hard.

This may be the most accurate description about U++ ever made!
U++ - it's free, it's good, but it's not documented and it's NOT DEBUG-ABLE REALLY FUCKED
HARD CODE WITH SOME REALLY DEEP SHIT OPTIMIZATIONS - download now :d :d :d

Haha laughing so hard right now.
Ok, anyway... Thanks for fixing it, I don't know what RawToValue exactly does, so I use it everywhere. As I understand it, Value is some kind of container which can hold almost anything. But to do so, you have to "convert" it, and that's what RawToValue does... At least that's my belief.

One thing I'm still curious about... What the heck is this assert:
`ptr()->GetType() >= 255 || !svo[ptr()->GetType()]`
and what does it have in common with
`inline _UInt4_t _Fetch_add_seq_cst_4(volatile _UInt4_t *_Tgt, _UInt4_t _Value)`
`{ /* add _Value to *_Tgt atomically with`
`sequentially consistent memory order */`

`return (_INTRIN_SEQ_CST(_InterlockedExchangeAdd)((volatile long *)_Tgt, _Value));`
`}`

Thanks
