Ruby master - Bug #11438

native_thread_init_stack() get machine.stack_start unequal to thread's stack start address, x86 win32

08/13/2015 07:31 AM - rickerliang (l ly)

Status:	Open		
Priority:	Normal		
Assignee:	cruby-windows		
Target version:			
ruby -v:	2.2.2	**Backport:**	2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN

Description

In function native_thread_init_stack() use VirtualQuery to get thread's stack start address.But some situation(ruby embbed in other application and initial it on the fly),native_thread_init_stack() will be called at low stack address and VirtualQuery return memory info BaseAddress + RegionSize < thread stack base(teb.StackBase).
In this situation,subsequently call stack_check() at high stack address will cause stack_overflow exception,because esp > machine.stack_start:
(teb.StackLimit < machine.stack_start < esp < teb.StackBase)
but actually it is not stack overflow at this time.
Use teb.StackBase instead of VirtualQuery get thread stack base is a more reliable solution.