

# How do I get the current value of the RSP register from a C/C++ function? (No answer, but a solution.)

[devblogs.microsoft.com/oldnewthing/20130501-00](http://devblogs.microsoft.com/oldnewthing/20130501-00)

May 1, 2013



Raymond Chen

A customer using Visual Studio wanted to know how to obtain the current value of the x64 RSP register from a C/C++ function. They noted that on the x86, you can drop to inline assembly, and on the ia64, you can use the `__getReg` intrinsic to retrieve the value of any register. There is no corresponding `__getReg` intrinsic on x64.

There's no really good way of doing this. The customer can sort of get close with `_AddressOfReturnAddress()`. Other approximations would be to call `_alloca(0)`, or to call a helper function which allocates a local variable and returns its address.

Now, if you were the one answering this question and you stopped there, then you'd be guilty of answering the question without first understanding the problem. There must be some larger problem they are trying to solve, and they managed to break it down into two parts, the easy part and the impossible part, and they wanted help with the impossible part.

The customer was kind enough to elaborate.

“We are using RSP to generate a system-wide unique value.”

Whoa, there. RSP is not all that unique. After all, if the function gets called twice, the RSP value might be the same the second time. (Maybe they clean up the unique value when the function returns, so this isn't a problem.) And of course, since each process has its own address space, another process can perform the same operation with the same value of RSP.

If you want a system-wide unique 64-bit value, use the function `AllocateLocallyUniqueId`.

Raymond Chen

**Follow**

