
Subject: Re: [PROPOSAL]: VarArgs class for U++ (va_ macros replacement, in U++ style)

Posted by [Oblivion](#) on Sun, 21 Feb 2016 20:34:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:1) From the documentation: "Any is a special type of container capable of containing none or single element of any type.". Why do you want to pass no arguments with Any? I believe you wanted to use Value instead.

2) va_ macros do not allocate anything on heap. Vector does, and Value can do that. So, it can be more expensive to prepare arguments for a call, then just call a function, and there is also memory fragmentation ...

3) Vector has a set of methods which is standard for vector-like classes. Design of associative containers in Upp is quite unique though.

Hello Novo, thank you for taking time to criticising my proposal.

1) I don't want to pass no arguments. AFAIK, it depends on the method used. Here, it should be guaranteed to be created with content. Besides, I considered writing this class using Value, and as a matter of fact I wrote it. But soon I ran into problems. Value only returns a constant reference, where I also want to allow passing of non-const references, pointers including callbacks, that can be manipulated (although I can use `const_cast`. In fact, this give me an idea...).

2) A fair point. But I don't think the memory overhead will be unacceptable (I admit I have to run tests, though). There are use cases, especially in async programming where calling variadic functions using traditional ways can get too tricky. (after all, VarArgs was born of this reason).

3) I don't quite understand this one. While VarArgs is technically a container (I defined it as such in the description, I know), it isn't per se. It is supposed to be a convenience class with vector-like interface.

Regards,

Oblivion
