

---

Subject: Re: 2024rc1

Posted by [mirek](#) on Mon, 21 Oct 2024 07:05:24 GMT

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

---

Lance wrote on Mon, 21 October 2024 02:27with the u++ moving to c++17,

code like this (Core/Vcont.h line 13)

```
void Malloc(size_t size) {  
    if(std::is_trivially_destructible<T>::value)  
        ptr = (T *)MemoryAlloc(size * sizeof(T));  
    else {  
        void *p = MemoryAlloc(size * sizeof(T) + 16);  
        *(size_t *)p = size;  
        ptr = (T *)((byte *)p + 16);  
    }  
}
```

can benefit from constexpr-if compile time trimming to produce more compact and faster binary (theoretically).

How? I have noticed that some people tend to constexpr to everything, but I fail to see a reason. If that is supposed to perform the test only in compile time, then 30 years old compiler will do that anyway. But I might be missing something perhaps?