
Subject: Re: Doubt with Buffer<> of a trivially destructible type

Posted by [koldo](#) on Thu, 12 Dec 2024 15:05:45 GMT

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

Thank you Oblivion

Yes, there are simple solutions. This case is to alloc data to be filled by a DLL made in Delphi, with particular alignment requirements.

I have reviewed deeper in how U++ handles this, and it takes care of that in run time:

```
void Free() {
    if(ptr) {
        if(std::is_trivially_destructible<T>::value) // <<=====
            MemoryFree(ptr);
        else {
            void *p = (byte *)ptr - 16;
            size_t size = *(size_t *)p;
            Destroy(ptr, ptr + size);           // <<===== Destroy() is called only if there is a destructor
            MemoryFree(p);
        }
    }
}
```

However the compiler detects this:

```
template <class T>
inline void Destruct(T *t)
{
    t->~T();           // error: object expression of non-scalar type 'double[3]' cannot be used in
a pseudo-destructor expression
}
```

```
template <class T>
inline void Destroy(T *t, const T *end)
{
    while(t != end)
        Destruct(t++);
}
```
