

## Re: Structures in Assembly Language

**Source:** <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2004-08/0963.html>

---

**From:** hutch-- (hutch\_at\_movsd.com)

**Date:** 08/19/04

Date: 18 Aug 2004 18:07:58 -0700

> *To beginners:*

This is claptrap, assemblers have had the capacity to write and manipulate structures since before 1990 and structures (C), UDTs (basic), records (pascal) are a normal part of programming. The difference is some assembler CAN handle structures and some CANNOT. When you have to write code for an operating system that commonly uses structures, you either use an assembler powerful enough to do it easily or get stuck with some dirty fudge to try and emulate something that is very simple.

It may suit peddlers or toys that are not powerful enough to write Windows code properly to boast of what their toys cannot do but if you need to write successful Windows code, you will do it with an assembler that is powerful enough to handle the layout of the operating system API functions as native assembler code.

A structure is in fact an ARRAY but with the difference that all members do not have to be the same size. The great advantage of using structures is that you can address each member by name and you ALWAYS get the correct address by doing so. There is nothing clever about manually coding array offsets through multiple levels of indirection, just highly unreliable code that is nearly impossible to debug and fix.

When Betov quotes a piece of the example code from MASM32, he fails to address WHY a CreateWindowEx() API uses a WNDCLASSEX structure. Windows code can be complicated if you do it wrong and to create a Window, you first must register a class for the Window. You use the WNDCLASSEX structure like a parameter list to set up the Window class and then you create the Window using the CreateWindowEx().

Its one of th worst kept secrets that Microsoft had technical problems with the design of early 32 bit Windows so they hired a pile of VAX mainframe programmers to design the PE file format used in 32 bit Windows. This is where the nested structures came from and without that capacity, coding Windows assembler is an unreliable unfixable pig

alt.lang.asm: Re: Structures in Assembly Language

that does not work properly.

This much I will say to programmers who are learning both Windows and assembler coding, don't be misled by bullsh\*t coming out of Betov when he does not have the experience coding Windows software. Without normal capacity like structures, you are left with simplistic capacity that cannot do the complicated stuff that Windows is made with.

Regards,

hutch at movsd dot com