

Isn't it time there was a standard align statement?

# Isn't it time there was a standard align statement?

---

*Source:* [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2007-01/msg01813.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2007-01/msg01813.html)

---

- *From:* [artifact.one@xxxxxxxxxxxxxxxx](mailto:artifact.one@xxxxxxxxxxxxxxxx)
  - *Date:* 13 Jan 2007 11:15:53 -0800
- 

It'd be really pleasant (in my opinion) if the next revision of the C language actually allowed some portable control over data alignment.

Compiler-specific mechanisms for this stuff are so varied that it becomes impossible to even abstract the details away behind preprocessor macros.

What I'd like to see:

```
/* per structure alignment */
align(16) struct xyz {
char x;
char y;
int z;
};
```

```
/* per member alignment (obviously padding before the first
member is illegal, so the entire structure would become aligned
in this case */
align(16) struct xyz {
align(16) char x;
char y;
int z;
};
```

```
/* per variable alignment */
align(16) unsigned int x;
```

I don't care about the syntax.

Now, obviously, C is meant to be implemented on everything from self-aware weather-control mainframes, to motorized tie racks, so in the case of the host implementation not supporting the specified alignment, a warning should be emitted and either the closest or natural alignment should be given. Warnings can obviously be made fatal with compiler specific switches – and that's no business of the language.

Isn't it time there was a standard align statement?

It just seems that this really should be standardized as it clearly is useful for a vast number of programmers who need to get close to the hardware but don't want to stray into assembly code (think AltiVec, SSE).

Sounds like EXACTLY the point of the C language, doesn't it?

I wouldn't mind so much if compiler implementors had come up with a vaguely portable way of doing this, but they haven't even come close.

GCC and Intel have won joint first prize for 'most pleasant implementation'

though (`__attribute__` or `_declspec()`).

cheers,

MC

.