Subject: Re: Make THISFN simpler and more powerful
Posted by Lance on Tue, 08 Oct 2024 16:38:32 GMT
View Forum Message <> Reply to Message

OK, this is not even c++ language related. But c++ is to be blamed.

Strange padding.

```
#include <iostream>

struct A{
    int a;
    unsigned b:16; // for demonstration purpose only
    bool c:1;
};

int main()
{
   std::cout<<"sizeof(A) is "<<sizeof(A)<<std::endl;
}
```

You would expect the result be 8, but on windows MSBT22x64 and CLANG64 will give 12.
(CLANG64 and GCC64 on linux give 8, which is naturally and as expected)

If you think that's not weired enough, it goes further.

```
#include <iostream>

struct A{
 int a;
 union{
  int dummy;
  struct{
      unsigned v1:16;
      bool     v2:1;
  };
 };
};


int main()
{
 A a;
 a.v2 = true;
 a.dummy = 0;
 std::cout<<"a.v2 is "<<a.v2<<std::endl;
}
```

The output is true. The compiler is smart enough to disregard the union request and put dummy and the unnamed struct object side by side, instead of start from the same address.

Maybe there is something I did wrong?