

---

Subject: Bug (?) in ide\Debuggers\Exp.cpp  
Posted by [mr\\_ped](#) on Tue, 23 May 2006 10:25:56 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

(I'm just trying to figure out why the "data[2]" does lead to "Only pointer can be dereferenced" ... still reading the Exp.cpp + parser.cpp classes, I will need probably some more time (like couple of days, as I have just couple of minutes per day for Ultimate++) to get familiar with those classes)

I found out suspicious code in Exp.cpp (line 136, 605dev1 src):

```
Pdb::Val Pdb::Compute(Pdb::Val v1, Pdb::Val v2, int oper)
{
  if(v1.ref) {
    int q = (int)GetInt(v2) * (v1.ref > 1 ? 4 : SizeOfType(v1.type));
    v1 = GetRVal(v1);
    switch(oper) {
      case '+': v1.address += q; break;
      case '-': v1.address -= q; break;
      default: ThrowError("Invalid pointer arithmetics");
    }
    return v1;
  }
  if(v1.ref) {
    int q = (int)GetInt(v1) * (v2.ref ? 4 : SizeOfType(v2.type));
    v2 = GetRVal(v2);
    if(oper == '+')
      v2.address += q;
    else
      ThrowError("Invalid pointer arithmetics");
    return v2;
  }
}
```

Looks to me like the second if should be:

```
if (v2.ref) {
```

Besides that it looks to me like (almost) duplicate code... which I personally quite dislike, when it can be avoided easily.

Maybe some

```
Pdb::val & _v1 = v1;
Pdb::val & _v2 = v2;
if (!v1.ref && v2.ref) _v1 = v2, _v2 = v1;
//further _v1 and _v2 are used instead of v1/v2
```

would be more elegant and more powerful (supporting "-" operator even if v2 is "ref"), but I don't have enough insight into the code to judge whether the "more powerful" part is actually needed, or counterproductive. And it would maybe still look somewhat cumbersome anyway. (And I didn't try

that proposal, so I'm not sure if it doesn't have some further catch.)

Edit: now I'm not sure if `_v1 = v2` would not overwrite also original `v1` content ... probably yes?

Once the reference is set, how to change it in C++?

I already forgot how references work exactly (as I do use plain C in work right now, for last year or so)... Maybe I should check the C++ language reference again.

---