

---

Subject: Rect::Union() question  
Posted by [Didier](#) on Thu, 03 Jul 2025 20:33:04 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello,

While going through Upp::Rect code searching for the right methods to use to calculate a bounding box, I stumbled on some intriguing code:  
two same methods : one fully template and the same template method specialized for <double>:  
nothing special ... except that I don't understand why one does 'p.x + 1' and the other one doesn't.

I see no valid reason to put the '+1' here : If I use 'float' instead of 'double' i will have a +1 applied even if my values are in the [0 ,0.1] range  
So I think they're may be an error here or at least with 'float'

```
Cor/Gtypes.h
template <class T>
void Rect_<T>::Union(Pt p) {
    if(IsNull(p)) return;
    if(IsNullInstance()) {
        right = 1 + (left = p.x);
        bottom = 1 + (top = p.y);
    }
    else
    {
        if(p.x >= right) right = p.x +1;
        else if(p.x < left) left = p.x;
        if(p.y >= bottom) bottom = p.y +1;
        else if(p.y < top) top = p.y;
    }
}

template <>
inline void Rect_<double>::Union(Point_<double> p) {
    if(IsNull(p)) return;
    if(IsNullInstance())
    {
        left = right = p.x;
        top = bottom = p.y;
    }
    else
    {
        if(p.x < left) left = p.x;
        else if(p.x > right) right = p.x;
        if(p.y < top) top = p.y;
        else if(p.y > bottom) bottom = p.y;
    }
}
```

---