

“WITH” Blocks

The **exception** to the undeclared variable name which you know has already been declared is in the “with” block and is easy to spot. You get a compiler error saying “so-and-so has not been declared”. You suspect screwy character case. With **KodeKeys**, you edit the offender and select it and hit “ctrl[0]” and all occurrences of it are changed to the upper/lower you just set. Attempt a compile and still get the same error. Look just above that for a “with” marker. Here’s an example:

```
//• WITH:  oldRect )
// CTools™ marker.

SizeWindow(wPtr, right - left, bottom - top, false);

ResizeText(good)

/* redraw text in wind**/

;
```

The error was “**right**” has not been declared. Well, you KNOW it has been, in QuickDraw.h! So the “with” is the culprit. With what? Something goes WITH “**right**”. “**oldRect**” goes with it! So, you put a dot after “oldRect”, hold down SHIFT, click in front of “oldRect.” and copy to the clip. Put the cursor in front of “right” and paste. Continue this for any other “undeclared” in the “with” block (in this case identified by braces) and move on to the next error. The finished product will look like this:

```
//• WITH:  oldRect. )// CTools™ marker.

    SizeWindow(wPtr, oldRect.right - oldRect.left,
                oldRect.bottom - oldRect.top, false);
    ResizeText(good);
/* redraw text in wind**/
```

The braces delineating the “with” block should be left in, for the time being. Sometimes there are nested or double WITH blocks, and these braces help reduce confusion. Leave the with marker in, also. You may have to come back and get information or do something else. Save removing this stuff until the final posh up,

after it runs.

Nested WITHs are handled similarly. They will say:

```
//• with something, somethingElse )
```

or

```
//• with oneThing )
```

```
SomeStuff (one, two, three);
```

```
//• with anotherThing )
```

```
SomeMoreStuff (four, five, six);
```

```
[on and on]
```

```
[blah, blah, blah]
```

It may be tricky figuring out what goes WITH what. I recommend getting a copy of Icosahedron.p and Icosahedron.c and comparing them, if you run into a severe nested with block. It has a LOT of them and both programs run.

A hint as to what goes with what is in the type of variable the names belong to, and what struct they are in. The ones with rectangle references for one “with” are fairly easy to figure out.