Useless Assignments

The Useless Assignments report shows assignments to global, static and local variables where the assigned value is never used, either because the variable is set again before it is read, or because the variable is never read after it is set. The existence of useless assignments might indicate that some logic is faulty or missing. Consider the following example:

```
int globalA = 1;
int globalB = 1;
int func(int paraA, int paraY) {
    paraY = globalB;
    globalA = paraY;
    return paraY;
}

int func(void) {
    int localX, localY, localZ;
    localX = 0;
    localY = 1;
    localZ = 2;
    localZ = func(localX, localY);
    return localZ;
}
```

By default, the report indicates the specific variable assignments where the variable itself is subsequently set again before it is read (localZ), or where the variable is never subsequently read (globalA). These are considered to be directly useless assignments.
As an option, you can set the report to check for transitively useless assignments. Notice that the assignments of paramY and globalB are now included in the report. The assignment of globalA in func2 is directly useless. Since the assignment of paramY only contributes to the useless assignment of globalA, this upstream assignment is considered to be transitively useless. One step further upstream, the same is true for the use of globalB in setting the value of paramY.