rules.sonarsource.com

C++ static code analysis: Standard groupings should be used with digit separators

2-3 minutes

C++14 introduced the ability to use a digit separator (') to split a literal number into groups of digits for better readability.

To ensure that readability is really improved by using digit separators, this rule verifies:

- **Homogeneity**
- Except for the left-most group, which can be smaller, all groups in a number should contain the same number of digits. Mixing group sizes is at best confusing for maintainers, and at worst a typographical error that is potentially a bug.
- **Standardization**
- It is also confusing to regroup digits using a size that is not standard. This rule enforce the following standards:
- Decimal numbers should be separated using groups of 3 digits.
- Hexadecimal numbers should be separated using groups of 2 or 4 digits.
- Octal and Binary should be separated using groups of 2, 3

or 4 digits.

Furthermore, using groups with more than 4 consecutive digits is not allowed because they are difficult for maintainers to read.

Noncompliant Code Example

```
long decimal_int_value     = 1'554'3124;          //
Noncompliant; mixing groups of 3 and 4 digits
double decimal_float_value = 7'91'87'14.3456;     //
Noncompliant; using groups of 2 instead of 3 digits
long hexadecimal_value     = 0x8'3A3'248'6E2;     //
Noncompliant; using groups of 3 instead of 2 or 4 digits
long octal_value           = 0442'03433'13726;    //
Noncompliant; using groups of 5 instead of 2, 3 or 4 digits.
long binary_value          = 0b01010110'11101010; //
Noncompliant; using groups of 8 instead of 2, 3 or 4 digits.
```

Compliant Solution

```
long decimal_int_value     = 15'543'124;
double decimal_float_value = 7'918'714.3456;
long hexadecimal_value     = 0x83'A324'86E2;
long octal_value           = 04'4203'4331'3726;
long binary_value          = 0b0101'0110'1110'1010;
```

Exceptions

No issue is raised on the fractional or exponent part of floating point numbers, only the integral part should comply.