

7/6/2016

Cppcheck: static code analysis for C++

Clemens Hentschke
Bug squashing seminar

dkfz.

GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION



50 Years – Research for
A Life Without Cancer

What is Cppcheck?

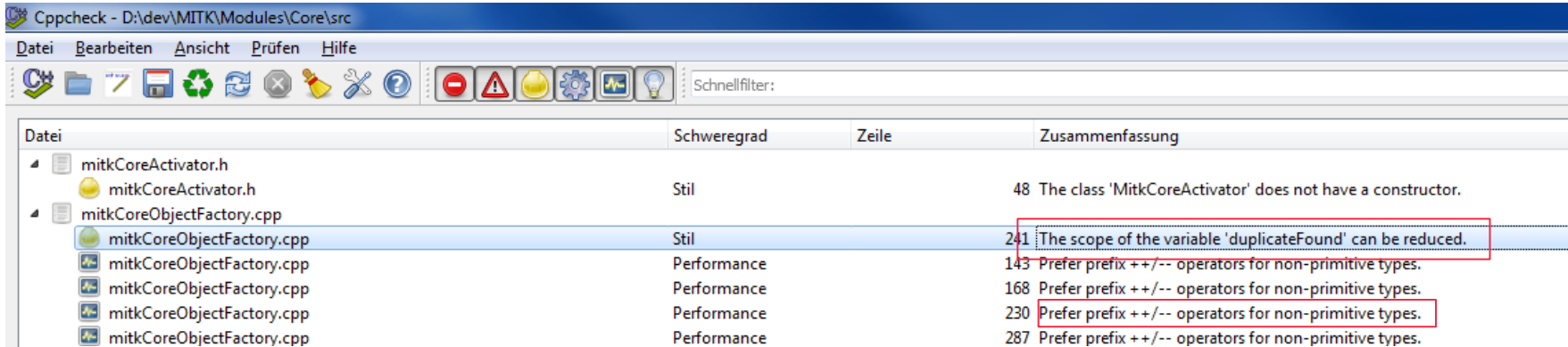
- Analyzes code
 - Static analysis: no syntax errors are detected
 - Aim to find bugs
 - Complementary to compilers
- Open source: <http://cppcheck.sourceforge.net>

Types of bugs

- Scope of the variable can be reduced
- Prefix ++/-- operators should be preferred for non-primitive types
- The exception should be caught as a (const) reference
- Condition 'var==0' is always true
- Variable 'value' is assigned a value that is never used.
- Unused variable: var
- Array out of bounds
- ...

Practical example – MITK I

- Modules/Core/src



Datei	Schweregrad	Zeile	Zusammenfassung
mitkCoreActivator.h			
mitkCoreActivator.h	Stil	48	The class 'MtkCoreActivator' does not have a constructor.
mitkCoreObjectFactory.cpp			
mitkCoreObjectFactory.cpp	Stil	241	The scope of the variable 'duplicateFound' can be reduced.
mitkCoreObjectFactory.cpp	Performance	143	Prefer prefix ++/-- operators for non-primitive types.
mitkCoreObjectFactory.cpp	Performance	168	Prefer prefix ++/-- operators for non-primitive types.
mitkCoreObjectFactory.cpp	Performance	230	Prefer prefix ++/-- operators for non-primitive types.
mitkCoreObjectFactory.cpp	Performance	287	Prefer prefix ++/-- operators for non-primitive types.

```

[...]
for (iterator it = m_ExtraFactories.begin();
     it != m_ExtraFactories.end(); it++ )
{
  [...]
  [...]
  bool duplicateFound = false;
  std::pair pairOfIter;
  for (iterator it = map.begin(); it != map.end(); ++it)
  {
    duplicateFound = false;
    [...]
  }
}

```

Practical example – MITK II

```
[...]
catch (itk::ExceptionObject e)
{
    MITK_ERROR << e << std::endl;
}
[...]
```

The scope of the variable 's' can be reduced.

Prefer prefix ++/-- operators for non-primitive types.
 Prefer prefix ++/-- operators for non-primitive types.
 Prefer prefix ++/-- operators for non-primitive types.
 Prefer prefix ++/-- operators for non-primitive types.
 Prefer prefix ++/-- operators for non-primitive types.
 Prefer prefix ++/-- operators for non-primitive types.
 Prefer prefix ++/-- operators for non-primitive types.
 Prefer prefix ++/-- operators for non-primitive types.
 Prefer prefix ++/-- operators for non-primitive types.

The scope of the variable 'geometry' can be reduced.
 Prefer prefix ++/-- operators for non-primitive types.

Exception should be caught by reference.

Prefer prefix ++/-- operators for non-primitive types.

The scope of the variable 'i' can be reduced.

The scope of the variable 'value' can be reduced.

The scope of the variable 'layer' can be reduced.

Condition 'transform==0' is always true

```
if ( transform != nullptr )
{
    [...]
}
else if ( transform == nullptr )
{
    [..]
}
```

Practical example – MITK III

DataManagement\mitkShaderProperty.cpp			
DataManagement\mitkShaderProperty.cpp	Performance	91	Prefer prefix ++/-- operators for non-primitive types.
DataManagement\mitkSlicedData.cpp			
DataManagement\mitkSlicedData.cpp	Stil	349	The scope of the variable 'slicedGeometry' can be reduced.
DataManagement\mitkSlicedGeometry3D.cpp			
DataManagement\mitkSlicedGeometry3D.cpp	Warnung	950	Redundant assignment of 'm_DirectionVector' to itself.
DataManagement\mitkSlicedGeometry3D.cpp	Stil	479	The scope of the variable 'geometry' can be reduced.
DataManagement\mitkSlicedGeometry3D.cpp	Stil	495	The scope of the variable 'geometry' can be reduced.
DataManagement\mitkStandaloneDataStorage.cpp			
DataManagement\mitkStandaloneDataStorage.cpp	Performance	37	Prefer prefix ++/-- operators for non-primitive types.
DataManagement\mitkStandaloneDataStorage.cpp	Performance	218	Prefer prefix ++/-- operators for non-primitive types.
DataManagement\mitkStandaloneDataStorage.cpp	Performance	224	Prefer prefix ++/-- operators for non-primitive types.
DataManagement\mitkTemporoSpatialStringProperty.cpp			
DataManagement\mitkTemporoSpatialStringProperty.cpp	Stil	36	Variable 'value' is assigned a value that is never used.
DataManagement\mitkTimeGeometry.cpp			

```

if ( centerOfRotationDistance > 0 )
{
    m_DirectionVector = m_DirectionVector;
}
else
{
    m_DirectionVector = -m_DirectionVector;
}

std::string value = s;
SliceMapType slices{ { 0, s } };
m_Values.insert(std::make_pair(0, slices));

```

What else?

- Available as git pre-commit hook
(<https://github.com/danmar/cppcheck/blob/master/tools/git-pre-commit-cppcheck>)
- Available as Visual Studio plugin
(<https://github.com/VioletGiraffe/cppcheck-vs-addin/releases/tag/1.3.3>)
- Full list of Checks
(<https://sourceforge.net/p/cppcheck/wiki/ListOfChecks/>)

Conclusion

- Cppcheck
 - Is a tool to find (simple) bugs/style/performance issues
 - Is easy to use
 - Gives advises
 - Leads to cleaner code
- it can't hurt to use 😊

Questions?



dkfz.

GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION



50 Years – Research for
A Life Without Cancer