



atollic



Atollic TrueINSPECTOR

Improve software quality with static source code inspection!

atollic ab

Science Park
Gjuterigatan 9
SE-553 18 Jönköping
Sweden
+46 (0)36 10 02 20

www.atollic.com

MISRA-C:2004

Reduce errors by limiting developer flexibility



What is it?	A coding standard developed by the automotive industry for improving reliability, safety, maintainability and portability of safety-critical systems.
Rule classifications	The rules can be classified as either being Required or only Advisory to use.
141 rules in 21 rule groups	Rule grouping for Environment, Language extensions, Documentation, Character sets, Identifiers, Types, Constants, Declarations and definitions, Initialization, Arithmetic type conversions, Pointer type conversions, Expressions, Control statement expressions, Control flow, Switch statements, Functions, Pointers and arrays, Structures and unions, Preprocessing directives, Standard libraries and Runtime failures.

Code metrics

Reduce errors by limiting code complexity

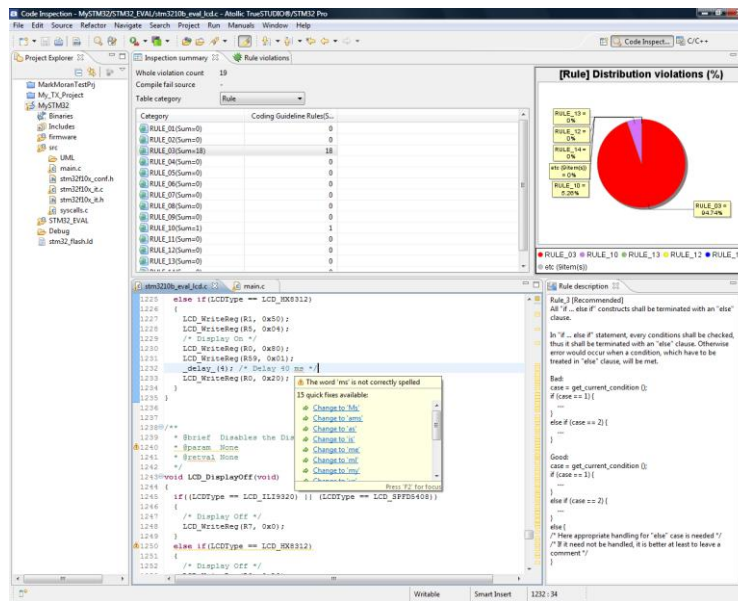


How is it gathered?	Code metrics is gathered by performing a static analysis of the source code.
What is the purpose?	Information on code complexity in different parts of the program gives hints on where the code needs to be simplified, thus reducing risk of errors and improving maintainability.
What is the benefit?	By simplifying complex sections of code, the likelihood of errors are reduced. This results in reduced costs, improved maintainability and higher software quality.



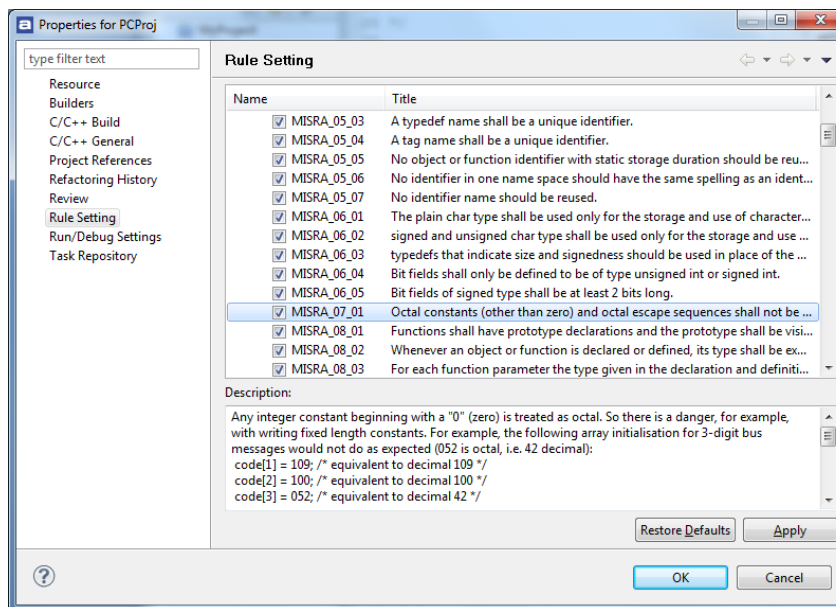
Atollic TrueINSPECTOR

A tool for professional source code analysis



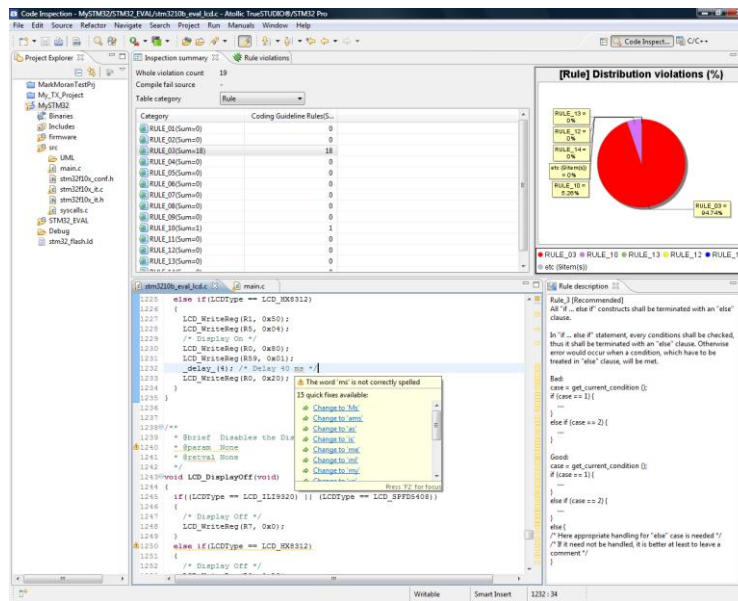
- Analyzes source code (complexity, cohesion, control flow graph, etc)
- Validates the source code against the MISRA-C:2004 coding standard
- Present code metrics (such as cyclomatic value of code complexity)
- Detects potential bugs and helps improve software quality

Selecting rules



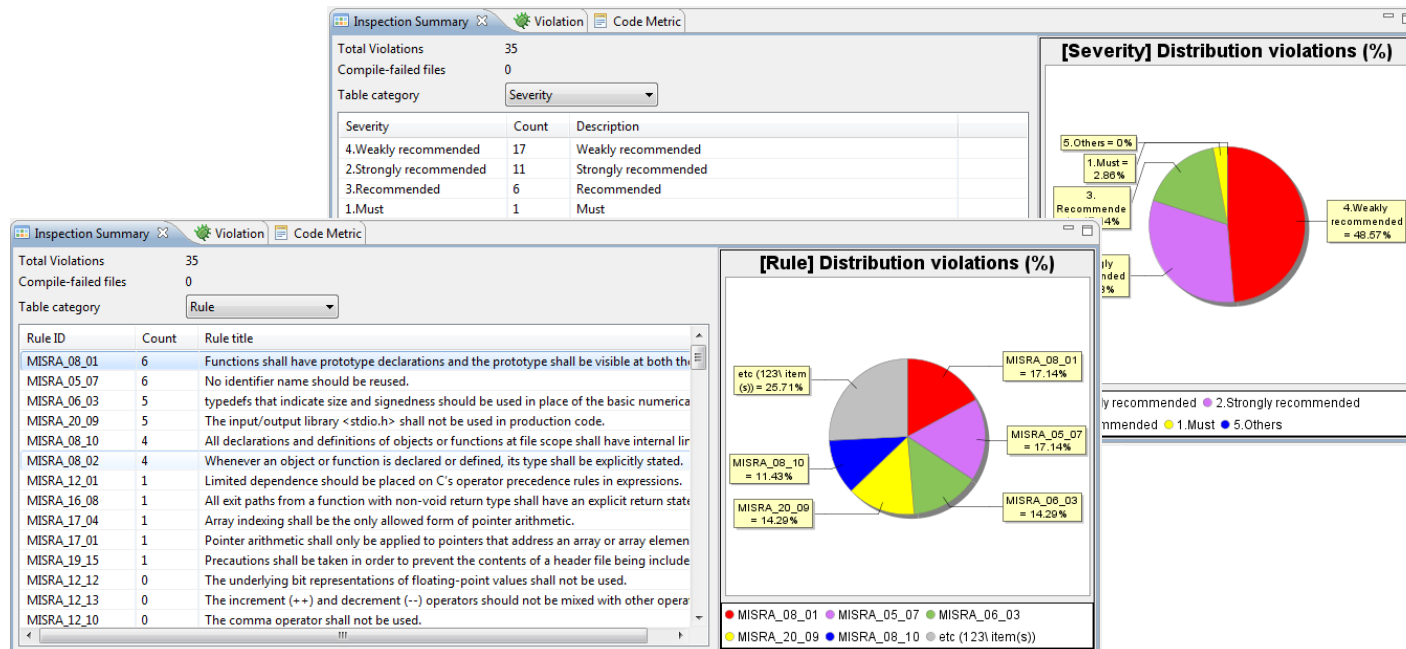
- Atollic TrueINSPECTOR allows individual selection of rules in a formal rule database
- Each rule is annotated with a clear explanation of what it does, in addition to its formal name and title

Integration with the editor



- Violating lines are flagged with markers in editor
- Tooltips explain the rule violation in detail
- The standard Problems view list rule violation problems, in addition to compiler errors and warnings
- Navigation from Problem view to editor lines

Inspection summary



- Summary display of Rules, Files and Severity results
- Pie charts gives excellent overview
- Text reports gives detailed information
- Different sorting options

Rule violations

The screenshot displays a static analysis tool interface with three main components:

- Inspection Summary:** Shows 35 items. A table lists violations with columns for Diagnosis, Rule, Severity, Source, Line, and Function.
- Export View Content:** A dialog box for exporting reports. The file name is "MyProject", the location is "C:\MISRA\Reports", and file extensions include DOC (.doc), HTML (.html), PDF (.pdf), PPT (.ppt), and XLS (.xls).
- RuleExplanation:** A window showing details for rule MISRA_05_07: "No identifier name should be reused." It includes a description and examples of bad and good code.

Diagnosis	Rule	Severity	Source	Line	Fun
MISRA_05_07 (6 items)					
An identifier 'i' is not unique identifier.(declared at[line:35, file:C:\Users\Magnus\Atollic\TrueSTU	MISRA_05_07	★★	main.c	61	add
An identifier 'i' is not unique identifier.(declared at[line:35, file:C:\Users\Magnus\Atollic\TrueSTU	MISRA_05_07	★★	main.c	79	sub
An identifier 'i' is not unique identifier.(declared at[line:35, file:C:\Users\Magnus\Atollic\TrueSTU	MISRA_05_07	★★	main.c	91	mul
An identifier 'xx' is	ueST MISRA_05_07	★★	main.c	59	add
An identifier 'xx' is	ueST MISRA_05_07	★★	main.c	77	sub
An identifier 'xx' is	ueST MISRA_05_07	★★	main.c	89	mul
MISRA_06_03 (5 items)					
MISRA_08_01 (6 items)					
MISRA_08_02 (4 items)					
A variable type is n	MISRA_08_02	★★★	main.c	55	calc
A variable type is n	MISRA_08_02	★★★	main.c	59	add
A variable type is n	MISRA_08_02	★★★	main.c	77	sub
A variable type is n	MISRA_08_02	★★★	main.c	89	mul
MISRA_08_10 (4 items)					
MISRA_12_01 (1 item)					

```

RULE : No identifier name should be reused.
DESCRIPTION : Regardless of scope, no identifier should be re-used
across any files in the system. This rule incorporates the provisions of
Rules 5.2, 5.3, 5.4, 5.5 and 5.6.
Where an identifier name is used in a header file, and that header file is
included in multiple source files, this rule is not violated. The use of a
rigorous naming convention can support the implementation of this
rule.

=====

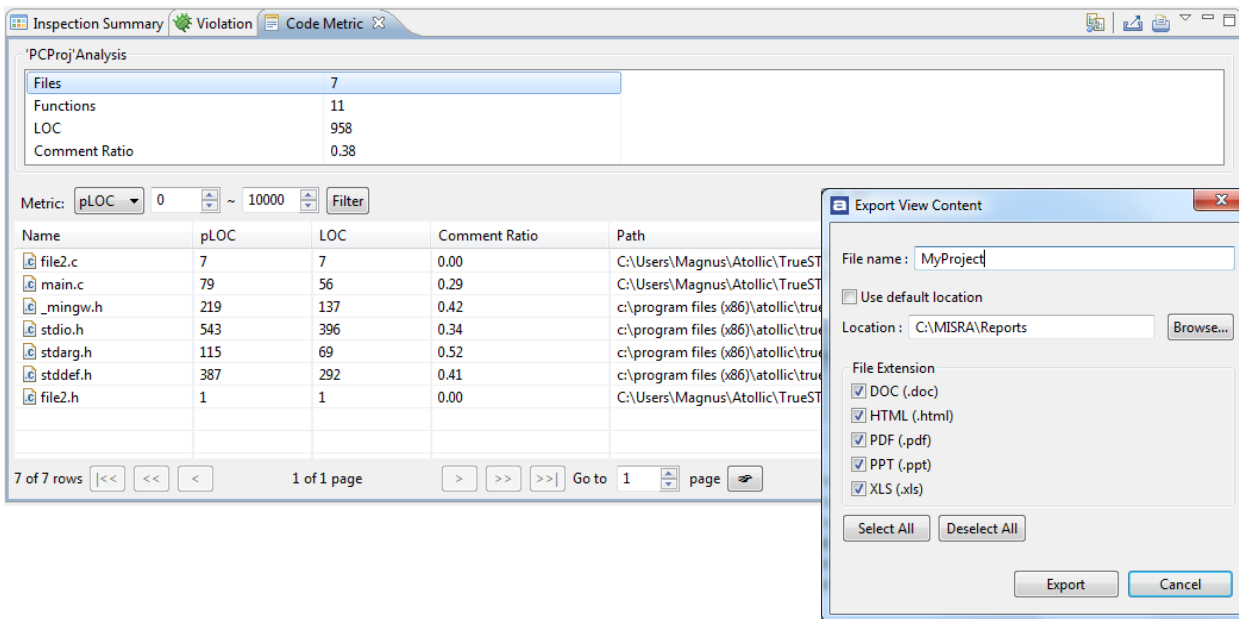
1. All identifiers shall be unique identifier.

▶ Bad Example ◀
struct air_speed {
    uint16_t speed; /* knots */
} * x;
struct gnd_speed {
    uint16_t speed; /* mph */ /* Not Compliant - speed is in different
units */
} * y;
x->speed = y->speed;

▶ Good Example ◀
struct air_speed {
    uint16_t a_speed; /* knots */
  
```

- List inspection violations per rule
- Explanation of rule details, the problem, and suggested solution
- Navigation links for every violation (jump to offending line in editor)
- Searching/filtering
- Export reports to MS-Word, MS-Excel, MS-PowerPoint, HTML and PDF

Code metrics



The screenshot displays a software interface for code metrics analysis. The main window, titled 'PCProj' Analysis, shows a summary of metrics:

Metric	Value
Files	7
Functions	11
LOC	958
Comment Ratio	0.38

Below the summary, a detailed table lists individual files with their respective metrics. The 'Metric' is set to 'pLOC' with a range from 0 to 10000. The detailed table is as follows:

Name	pLOC	LOC	Comment Ratio	Path
file2.c	7	7	0.00	C:\Users\Magnus\Atollic\TrueST
main.c	79	56	0.29	C:\Users\Magnus\Atollic\TrueST
_mingw.h	219	137	0.42	c:\program files (x86)\atollic\true
stdio.h	543	396	0.34	c:\program files (x86)\atollic\true
stdarg.h	115	69	0.52	c:\program files (x86)\atollic\true
stddef.h	387	292	0.41	c:\program files (x86)\atollic\true
file2.h	1	1	0.00	C:\Users\Magnus\Atollic\TrueST

An 'Export View Content' dialog box is overlaid on the detailed table. It allows the user to export the data to various formats. The 'File name' is 'MyProject', and the 'Location' is 'C:\MISRA\Reports'. The 'File Extension' options are checked for DOC (.doc), HTML (.html), PDF (.pdf), PPT (.ppt), and XLS (.xls). The dialog also includes 'Select All', 'Deselect All', 'Export', and 'Cancel' buttons.

- Display of source code metrics
- Number of files, functions, lines and comment line ratio
- Cyclomatic value of code complexity
- Export reports to MS-Word, MS-Excel, MS-PowerPoint, HTML and PDF



atollicTM

Improve your software quality with
Atollic TrueINSPECTOR!

Any questions?

"Embedded passion"