

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C#
- CSS
- Flex
- Go
- HTML
- Java
- JavaScript
- Kotlin
- Kubernetes
- Objective C
- PHP
- PL/I
- PL/SQL
- Python
- RPG
- Ruby
- Scala
- Swift
- Terraform
- Text
- TypeScript
- T-SQL
- VB.NET
- VB6
- XML**

# XML static code analysis

Unique rules to find Bugs and Code Smells in your XML code

- All rules 36
- Vulnerability 6
- Bug 5
- Security Hotspot 9
- Code Smell 16

Tags ▾

Search by name... 🔍

Track uses of "TODO" tags	
EJB interceptor exclusions should be declared as annotations	
Track uses of disallowed dependencies	
Newlines should follow each element	
XML parser failure	
Track breaches of an XPath rule	
Lines should not be too long	
pom elements should be in the recommended order	
Artifact ids should follow a naming convention	
Group ids should follow a naming convention	
"action" mappings should not have too many "forward" entries	
Source code should be indented consistently	
Tabulation characters should not be used	

## Artifact ids should follow a naming convention

Analyze your code

Code Smell Minor convention maven

Shared naming conventions allow teams to collaborate effectively. This rule raises an issue when a pom’s artifactId does not match the provided regular expression.

### Noncompliant Code Example

With the default regular expression: `[ a-z ] [ a-z-0-9 ] +`

```
<project ...>
  <artifactId>My_Project</artifactId>  <!-- Noncompliant -->

  <!-- ... -->
</project>
```

### Compliant Solution

```
<project ...>
  <artifactId>my-project</artifactId>

  <!-- ... -->
</project>
```

Available In:

sonarlint | sonarcloud | sonarqube