Rule: Tool Selection and Validation

A compiler (which should be understood to include the linker) that complies with the Java Language Specification [JLS 2005] should be used whenever possible.

A source code analysis tool should be able to enforce as many of the guidelines in this document as possible. Not all recommendations are enforceable; some are strictly meant to be informative.

Completeness and Soundness

To the greatest extent possible, the static analysis tool should be both complete and sound with respect to enforceable guidelines.

A static analysis tool is considered complete (with respect to a specific guideline) if it does not give a false-negative result, meaning it can find all violations of a guideline.

A static analysis tool is considered sound if it does not issue false-positive results, or false alarms. Too many such false alarms rapidly lead users to ignore the results of a tool, potentially missing serious issues.

The possibilities for a given guideline are outlined in the following table.

<table>
<thead>
<tr>
<th>False Negatives</th>
<th>False Positives</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>N</td>
<td>Complete with false positives</td>
</tr>
<tr>
<td>Y</td>
<td>Incomplete with false positives</td>
</tr>
</tbody>
</table>

The final possibility is that a tool has not implemented a checker for a specific guideline.

Compilers and source code analysis tools are trusted processes, meaning that a degree of reliance is placed on the output of the tools. Consequently, developers must ensure that this trust is not misplaced, ideally by confirming that the tool supplier has run appropriate validation tests. Although it is possible to use a validation suite to test a compiler or source code analysis tools, no formal validation scheme exists at this time.

False Positives

Many guidelines list common exceptions, but it is difficult, if not impossible, to develop a complete list of exceptions for each guideline. Consequently, it is important that source code comply with the intent of each guideline and, to the greatest extent possible, tools minimize false positives that do not violate the intent of the guideline. The degree to which tools minimize false-positive diagnostics is a quality-of-implementation issue.