

Changes in results post upgrade - 8.3.37

- [.NET](#)
 - [Discoverer/NuGet Resources Extractor](#)
- [User Input Security](#)
 - [Total Checks discrepancies \(.NET / JEE\)](#)
 - [.NET - Avoid weak cryptographic algorithm \(8414\)](#)
- [SAP, .NET, C/C++, JEE, PowerBuilder](#)
 - [Avoid Classes with a High Lack of Cohesion \(7798\)](#) and [Avoid Classes with a High Lack of Cohesion variant \(7796\)](#)



Summary:

- Impacts of changes made to AIP Core 8.3.37 on Quality Model results post upgrade
- Other impacts of changes made in AIP Core 8.3.37



All changes in results related to extensions are listed in the extension documentation and will not appear in this page.

.NET

Discoverer/NuGet Resources Extractor

Improvements have been made to the .NET discoverer, embedded in AIP Core. These improvements are designed to allow additional packages to be considered by any release of the [NuGet Resources Extractor](#) (com.castsoftware.dmdotnetnugetresourcesextractor): packages referenced in the "packages.config" file (if delivered with the source code) will now be extracted and resolved (previously packages referenced in this file were ignored). This change may impact previous analysis results if upgrading to AIP Core 8.3.37 (or future releases): additional packages may be extracted that were previously ignored.

User Input Security

Total Checks discrepancies (.NET / JEE)

A bug has been discovered which was causing the **Total Checks** value (displayed in the Engineering Dashboard) to be less than the number of reported violations for the following rules triggered during a User Input Security analysis:

- [Avoid hard-coded credentials \(8222\)](#)
- [Avoid using unsecured cookie \(8240\)](#)
- [Avoid weak cryptographic algorithm \(8414\)](#)
- [Avoid use of a reversible one-way hash \(8416\)](#)
- [Avoid using hard-coded HMAC keys \(8424\)](#)
- [Avoid using insufficient random generator \(8554\)](#)

This bug has been fixed and the number of total checks will now not be less than the total number of violations reported for the rules listed above. This may mean that the number of Total Checks changes after upgrading to 8.3.37 and running a new analysis on unchanged source code.

.NET - Avoid weak cryptographic algorithm (8414)

The rule "Avoid weak cryptographic algorithm" (8414) for .NET has been improved to detect violations for "constructor call" targets for the System.Security.Cryptography API. This improvement may impact existing results - additional violations may be detected.

SAP, .NET, C/C++, JEE, PowerBuilder

[Avoid Classes with a High Lack of Cohesion \(7798\)](#) and [Avoid Classes with a High Lack of Cohesion variant \(7796\)](#)

In an effort to improve the results and reduce the number of false positives returned by the rules **Avoid Classes with a High Lack of Cohesion (7798)** and **Avoid Classes with a High Lack of Cohesion variant (7796)** (both non-critical rules), the scope of both rules has been modified. Starting AIP Core 8.3.37, only classes with at least one field and more than one method will form the scope of objects considered for these two rules. As a result of this change to the scope, existing analysis results may be impacted when upgrading to AIP Core 8.3.37 and running a new analysis with unchanged source code:

The **Lack of Cohesion** of a class is calculated according to a mathematical formula where a value is divided by <number of fields> in the class and by (number of methods in the class -1). This leads to a division by zero error for classes with no fields and for classes with only a single method. The previous quality rule implementation considered for such classes that they have the value "1" for lack of cohesion (i.e. no cohesion at all) and therefore considered them systematically as a violation. However, this choice is arguable and therefore, starting from AIP Core 8.3.37, classes with 0 fields and classes with a single method are excluded from the scope of the quality rules. In other words, the quality rules are not checked against such classes anymore. As a consequence, all violations previously reported for such classes are not reported anymore after analysis with 8.3.37 or a later service pack. CAST has observed on a representative sample of applications that the number of violations for each of the two quality rules can drop by up to 90%. However, the total number of violations (for all quality rules together) drops typically only by around 0.5%. The amount of change remained always below 2%. As a direct result of the reduction in violations returned, grades for **Changeability** and **TQI** drop also by less than 1%.