

Changes in results post upgrade - 8.3.16

On this page:

- [Impacts of changes made in CAST AIP 8.3.16 on Quality Model results post upgrade](#)
 - [.NET](#)
 - [Various rules](#)
 - [Mainframe](#)
 - [MAINFRAME-283 - Prefer using indexes instead of subscripts - 8142](#)
 - [MAINFRAME-251 - Avoid OPEN/CLOSE inside loops - 7218](#)
 - [MAINFRAME-314 - Avoid unreferenced Sections and Paragraphs - 7290](#)
 - [MAINFRAME-300 - Never truncate data in MOVE statements - 7688](#)
 - [MAINFRAME-252 - Avoid unchecked return code \(SQLCODE\) after EXEC SQL query - 7690](#)
 - [SAP / ABAP](#)
 - [SAP-172 - "CX_ROOT" should not be used in TRY .. CATCH.. ENDTRY block \(8412\)](#)
 - [Multiple changes](#)
 - [Disabled rules](#)
 - [Bug correction](#)
 - [User Input Security related](#)
 - [AIPCORE-571 - Avoid HTTP response splitting - 7740](#)
 - [.NET](#)
 - [Java](#)
 - [AIPCORE-873 - Avoid NoSql injection - 8418](#)
 - [AIPCORE-875 - Avoid log forging vulnerabilities - 8044](#)
 - [AIPCORE-920 - Avoid weak cryptographic algorithm - 8414](#)
 - [AIPCORE-93 - Avoid hard-coded credentials - 8222](#)
 - [AIPCORE-777 - Total number of violations is higher than the total number of checks for certain rules](#)
 - [AIPCORE-571 - False violations on JEE source code](#)
- [Other impacts of changes made in CAST AIP 8.3.16](#)
 - [Oracle PL/SQL \(embedded analyzer\)](#)
 - [Mainframe](#)
 - [MAINFRAME-254 - MOVE PROGRAM-ID ... TO ... syntax](#)
 - [MAINFRAME-248 - Cobol Transaction objects](#)
 - [MAINFRAME-298 - CICS Maps objects](#)

Target audience:

CAST AI Administrator



Summary: this page lists:

- [Impacts of changes made to CAST AIP 8.3.16 on Quality Model results post upgrade](#)
- [Other impacts of changes made in CAST AIP 8.3.16](#)



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

Impacts of changes made in CAST AIP 8.3.16 on Quality Model results post upgrade

.NET

Various rules

The following multi-techno rules have been disabled in 8.3.16 specifically and only for .NET technology and will no longer be triggered during an analysis. These rules often generated a large amount of false positive violations. As a result of this change, results may be impacted - no violations will be triggered for any of these rules, therefore potentially impacting grades and existing results:

- [Avoid unreferenced Classes - 7832](#)
- [Avoid unreferenced Data Members - 7912](#)
- [Avoid unreferenced Methods - 7908](#)

Mainframe

MAINFRAME-283 - Prefer using indexes instead of subscripts - 8142

A bug has been discovered which is causing false positive violations of this rule (when indexes are used, violations are still reported). This bug has now been fixed, therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change: less false positives, more accuracy.

MAINFRAME-251 - Avoid OPEN/CLOSE inside loops - 7218

A bug has been discovered which is causing false positive violations of this rule (a false link between two objects). This bug has now been fixed, therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change: less false positives, more accuracy.

MAINFRAME-314 - Avoid unreferenced Sections and Paragraphs - 7290

A bug has been discovered which is causing false positive violations of this rule (incorrect handling of the syntax FETCH / END-FETCH). This bug has now been fixed, therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change: less false positives, more accuracy.

MAINFRAME-300 - Never truncate data in MOVE statements - 7688

A bug has been discovered which is causing false positive violations of this rule when the variables have subordinate items and the comparison is based on a block. This bug has now been fixed, therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change: less false positives, more accuracy.

MAINFRAME-252 - Avoid unchecked return code (SQLCODE) after EXEC SQL query - 7690

A bug has been discovered which is causing false positive violations of this rule when SQLCODE is checked outside perform statement of a paragraph. This bug has now been fixed, therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change: less false positives, more accuracy.

SAP / ABAP

SAP-172 - "CX_ROOT" should not be used in TRY .. CATCH.. ENDTRY block (8412)

The parent technical criterion for this rule was incorrectly set to **61020: Programming Practices - Modularity and OO Encapsulation Conformity**, and this has now been changed to **61014: Programming Practices - Error and Exception Handling**. Therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change. Grades for the new and previous parent technical criteria and Health Factors will change.

Multiple changes

Disabled rules

The following rules have been disabled in CAST AIP 8.3.16, therefore after upgrade to this release and the generation of a post upgrade consistency snapshot on unchanged source code, results may differ:

- **Avoid using literals in assignments (hardcoded values) (7522)**
- **Avoid "SELECT *" queries (7344)**

Bug correction

Various bugs have been fixed in this release, therefore after upgrade to this release and the generation of a post upgrade consistency snapshot on unchanged source code, results may differ:

| | |
|---|--|
| Avoid using AT Events in combination of LOOP AT WHERE constructs (7536) | This rule has been found to not function correctly in previous releases of CAST AIP. This bug has now been fixed, therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change. You may have violations for this rule where previously there were none. |
| Avoid empty catch blocks (7788) | <p>A bug has been discovered which was causing violations to be not reported for the following syntax:</p> <pre>CATCH SYSTEM-EXCEPTIONS . ENDCATCH .</pre> <p>This bug has now been fixed, therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change. You may have violations for this syntax where previously there were none.</p> |

| | |
|---|--|
| Avoid using hardcoded paths (7526) | <p>A bug has been discovered which was causing violations to be not reported if they occurred in the VALUE part of DATA statements, for example:</p> <pre style="border: 1px solid black; padding: 5px;">DATA d TYPE char22 VALUE 'c:/mypath' .</pre> <p>This bug has now been fixed, therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change. You may have violations for this syntax where previously there were none.</p> |
|---|--|

User Input Security related

AIPCORE-571 - Avoid HTTP response splitting - 7740

This rule has been updated to add specific target methods for both .NET and JEE. The methods listed below are now take into account, therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change. You may have violations where previously there were none.

.NET

- System.Net.HttpListenerResponse.AddHeader([mscorlib]System.String,[mscorlib]System.String) // Arg 2
- System.Web.HttpResponseBase.AddHeader([mscorlib]System.String,[mscorlib]System.String) // Arg 2
- System.Web.HttpResponse.AddHeader([mscorlib]System.String,[mscorlib]System.String) // Arg 2
- System.Web.HttpCookieCollection.Add(System.Web.HttpCookie) // Arg 1
- System.Web.HttpCookieCollection.Set(System.Web.HttpCookie) // Arg 1

Java

- javax.servlet.http.HttpServletResponse.addCookie(javax.servlet.http.Cookie) // Arg 1
- javax.servlet.http.HttpServletResponse.addHeader([ext]java.lang.String,[ext]java.lang.String) // Arg 2
- javax.servlet.http.HttpServletResponse.setHeader([ext]java.lang.String,[ext]java.lang.String) // Arg 2
- org.apache.http.impl.client.BasicCookieStore.addCookie(org.apache.http.cookie.Cookie) // Arg 1
- org.apache.http.client.CookieStore.addCookie(org.apache.http.cookie.Cookie) // Arg 1
- javax.servlet.http.HttpServletResponseWrapper.setHeader([ext]java.lang.String,[ext]java.lang.String) // Arg 2
- javax.servlet.http.HttpServletResponseWrapper.addHeader([ext]java.lang.String,[ext]java.lang.String) // Arg 2

AIPCORE-873 - Avoid NoSql injection - 8418

This rule existed for .NET technologies, however there was no support for JEE. This has now been fixed and therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change if you have JEE / NoSQL source code. You may have violations where previously there were none.

AIPCORE-875 - Avoid log forging vulnerabilities - 8044

This rule has been updated to take into account the following targets in the namespace System.Diagnostics.Debug, methods:

- Write
- Writelf
- WriteLinelf

Therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change. You may have violations where previously there were none.

AIPCORE-920 - Avoid weak cryptographic algorithm - 8414

This rule has been updated to raise a violation when the use of Triple DES (3DES or TDES) is detected (previously the use of Triple DES would not raise a violation). Therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change. You may have violations where previously there were none.

AIPCORE-93 - Avoid hard-coded credentials - 8222

Improvements have been made to improve the detection of signatures for the DBCP and SSH libraries' sendcredential methods. Therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change. You may have violations where previously there were none.

AIPCORE-777 - Total number of violations is higher than the total number of checks for certain rules

A bug has been discovered that is causing the values for Total Checks and Number of Violations to be erroneous (the total Number of Violations is higher than the total number of checks performed, which then generated an erroneous Compliance value) for the following User Input Security related rules:

- Avoid using insufficient random values for cookies - 8242

- Avoid weak cryptographic algorithm - 8414
- Avoid use of a reversible one-way hash - 8416
- Avoid using hard-coded HMAC keys - 8424

This bug has now been fixed therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change. Number of violations should be equal to or less than the Total checks, generating a coherent Compliance value.

AIPCORE-571 - False violations on JEE source code

Various false violations have been discovered in User Input Security related rules, therefore the following changes have been made in an effort to reduce the number of false violations:

- **java.util.Scanner** is no longer considered as an input method
- Some constructors of types with **Exception** in their name are incorrectly blackboxed as target "files" therefore improvements have been made. Examples:
 - Blackbox on the fly: [classpath]java.io.IOException.+ctor(ref [classpath]java.lang.String) with target file
 - Blackbox on the fly: [classpath]java.io.FileNotFoundException.+ctor() with target file
 - Blackbox on the fly: [mscorlib]System.IO.FileNotFoundException.+ctor([mscorlib]System.String) with target file
 - Blackbox on the fly: [mscorlib]System.IO.DirectoryNotFoundException.+ctor([mscorlib]System.String) with target file
- The definition of class **java.io.ObjectInputStream** was missing and this has now been corrected
- The target **web** (XSS) for the **javax.servlet.http.HttpServletResponse.sendError** class was missing and this has now been corrected

Therefore, after an upgrade to CAST AIP 8.3.16 and the generation of a post-upgrade consistency snapshot on unchanged source code, results may change. You may no longer have violations where previously violations existed.

Other impacts of changes made in CAST AIP 8.3.16

Oracle PL/SQL (embedded analyzer)

A change has been made to the way in which the analyzer handles the XMLROOT syntax (use of an "identifier" instead of "VERSION"). As a result of this change, after an upgrade to CAST AIP 8.3.16 and the generation of a post upgrade consistency snapshot on unchanged source code, you should expect many Oracle PL/SQL objects to be marked as modified.

Mainframe

MAINFRAME-254 - MOVE PROGRAM-ID ... TO ... syntax

A bug has been discovered which is causing the creation of an incorrect Cobol program object called "TO" for the "MOVE PROGRAM-ID ... TO ..." syntax found in cobybook files. This bug has now been fixed (the syntax is correctly handled) therefore, as a result of this change, after an upgrade to CAST AIP 8.3.16 and the generation of a post upgrade consistency snapshot on unchanged source code, results may change - less invalid objects providing more accuracy.

MAINFRAME-248 - Cobol Transaction objects

When running a Mainframe analysis, Cobol Transaction objects may be created with object names that contain only special characters such as * or /. This is due to the way the Inference Engine functions. A fix has been provided to avoid creating objects via the Inference Engine which contain only special characters (in other words, objects must contain at least one alphabetical character), therefore, as a result of this change, after an upgrade to CAST AIP 8.3.16 and the generation of a post upgrade consistency snapshot on unchanged source code, results may change - less invalid objects providing more accuracy.

MAINFRAME-298 - CICS Maps objects

A bug has been discovered which is causing the creation of CICS Maps objects as "unknown" and the same objects are created multiple times causing issues with link resolution. This bug has now been fixed therefore, as a result of this change, after an upgrade to CAST AIP 8.3.16 and the generation of a post upgrade consistency snapshot on unchanged source code, results may change - CICS Maps objects are handled correctly.