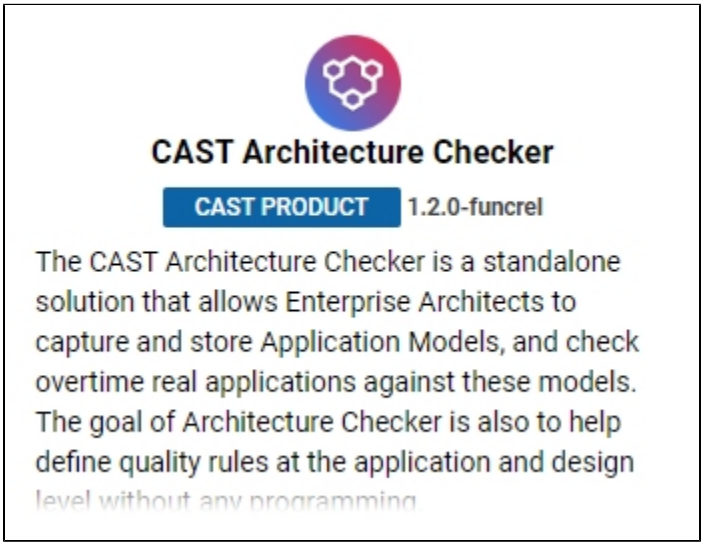
Changes or new features - 8.3.23

Summary: CAST AIP 8.3.23 introduces a number of features and changes as listed below.

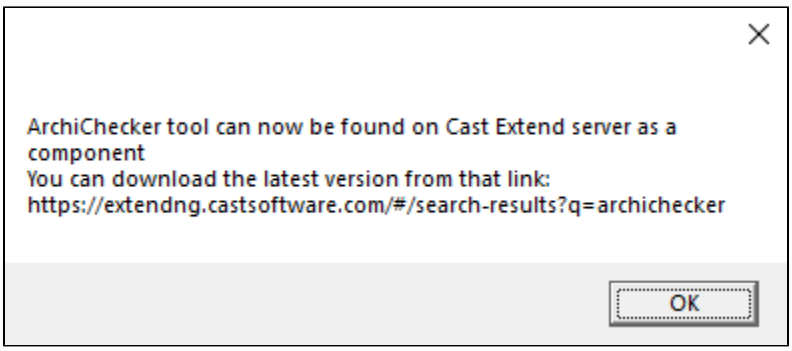
CAST AIP setup and CAST Architecture Checker

Starting from this release of CAST AIP, **CAST Architecture Checker** will no longer be installed as part of the CAST AIP setup, whether installing CAST AIP from scratch or on a server where a previous release of CAST AIP exists. CAST Architecture Checker has evolved into a **standalone component** where all feature requests and bug fixes are now managed. This standalone component can be downloaded from **CAST Extend** (**https://extendng.castsoftware.com/#/search-results?q=archichecker**):



For more information about the standalone release, how to install it and how to use it, please see **CAST Architecture Checker**.

Note that the CAST AIP setup will create a Windows Start menu shortcut for CAST Architecture Checker - when clicked, a popup message is displayed explaining that CAST Architecture Checker can be downloaded from CAST Extend:



CAST Delivery Manager Tool

Auto selection of Maven JAR files

In CAST AIP 8.3.23, if a JAR named after the name of the requested artifact is provided in the source code folder, it is accepted and a missing artifact alert is no longer raised. Previously, a manual remediation task was required to associate a JAR to a missing Maven project. Now, it is automatic. As a result of this change, for Maven projects, there may be no need to provide a Maven resource package if the corresponding JAR files have been provided in a sub-folder of the source package.

> Note that this behaviour does not apply to Gradle based projects. Any required Maven repositories must be delivered manually.

Change to the selection of artifacts with non-standard qualifiers in the version name

In CAST AIP 8.3.23, the CAST Delivery Manager Tool will remove non-standard qualifiers before comparing the Maven versions, so that, even if the plain version is lower than the qualified version, it can be accepted as a suitable version.

User Input Security

Change to support of org.springframework.jdbc

In previous releases of CAST AIP, support of the API **org.springframework.jdbc** for the User Input Security feature relied on **automatic blackboxing**. In CAST AIP 8.3.23, this has now changed and static rules will be used instead. Note that this change may **impact existing analysis results**.

Bug fixing

- **Error handing improvement** - If the User Input Security has no file(s) to analyze this is highly likely to be due to an internal bug. In this situation, the User Input Security will now report an error instead of indicating that the analysis has succeeded.
- **Variations in the number of violations between two consecutive snapshots with the same source code** - a bug has been fixed which was causing varying numbers of violations to be displayed for certain User Input Security related rules between two consecutive snapshots of the same application with unchanged source code. This was due to a bug causing different paths to be calculated for a given entrypoint.

Documentation updates

A new page has been added to help users improve the performance of an anlsysis that includes a User Input Security check. See **User Input Security - Advanced configuration to improve performance** for more information.

SecurityAnalyzer.log updates

A minor change has been made to change a message level from WARN to INFO:

```
Reached new field resolution algorithm failure count limit (10), will use default field resolution algorithm,
next time.
```

Rule documentation updates

The following changes have been applied to rule documentation (no impact on analysis results):

8098	Avoid uncontrolled format string	The **Sample** section contained a code error. The following line:
		`FormatterCase formatter = FormatterCase();`
		has been replaced by:
		`FormatterCase formatter = new FormatterCase();`

8240	Avoid using unsecured cookie	A limitation has been added in the **Description** section: `Limitation: in .NET environments, this rule does not check if the key requireSSL in web.config file.`

CAST Transaction Configuration Center

Change in behaviour when loading .TCCSetup configuration files (the automatic configuration refresh process)

Previously when uploading a .TCCSetup file which already existed and where the package-version of the file differs with the existing package-version in the Management schema, the following behaviour was used: each rule will be loaded with status **active** by default, except if the rule was present in the previous version, its definition is unchanged, and it had been **manually deactivated** by the user, in that case, the rule will be set to **inactive** as well.

From **CAST AIP 8.3.23**, this behaviour changes as follows (see also **TCC - Working with standard configuration files (.TCCSetup)**):

Each rule which was present in the previous version will be loaded with the same status as before, whichever the definition of this rule is the same or has changed. In this latter case, a warning will be logged to inform the user of this change, as in the example below where both the definition of an active Entry Point rule and of a deactivated End Point rule have both changed in a new version of the 'Base_HTML5' package:

```
WRN: -the rule "Standard Entry Point - HTML5 AspDotNet" (type='Transaction entry points') will remain active
because although its definition has changed, it has the same name and type as the previously active rule
"Standard Entry Point - HTML5 AspDotNet" (type='Transaction entry points').
WRN: -the rule "Standard End Point - HTML5" (type='Transaction end points') will remain deactivated because
although its definition has changed, it has the same name and type as the previously deactivated rule "Standard
End Point - HTML5 (type='Transaction end points').
```