CAST HIGHLIGHT

CODE SCAN BEST PRACTICES

CAST

Software Intelligence for Digital Leaders

CAST HIGHLIGHT

CODE SCAN BEST PRACTICES

CAST
Software Intelligence for Digital Leaders

In some specific contexts (Tech Due Diligence, M&A...), it is useful to refine the code scan scope to provide more focused insights from CAST Highlight. This document provides a set of best practices to refine the code scan scope based on the desired focus area: Software Health, Cloud Readiness, Software Composition Analysis or all of them.

	Health & Cloud	SCA	All Use Cases
Focus Area	Assess Software Health and Cloud Readiness of custom source code that composes your app/product, by making sure scores are not biased with third-party software (e.g. Open Source, COTS), as your development team generally can't modify this external code base.	Assess Vulnerability, License and Obsolescence risks of your app/product, considering the developed and deployed software, by making sure third-party components (e.g. Open Source, COTS) are part of the assessment.	Assess Software Health, Cloud Readiness and Third-Party component risks across the portfolio to support all of the use cases with comprehensive results.
Source Code Scan	✔	✔	✔
Deployed/Build Scan		✔	✔

CAST HIGHLIGHT

CODE SCAN BEST PRACTICES

CAST
Software Intelligence for Digital Leaders

	Software Health & Cloud Readiness
Source Code Scan	Include: - Source code Exclude: - Third-party component sources (typically in “lib”, “third-party”, “3rd-party”, “COTS”, “external”, “node_modules” folders, etc.) - Tests - Generated code (e.g. t.ds, .flow.js) - Deployment, SCM folders and files (e.g. .git, .svn, gradlew, .vscode, etc.) CAST Highlight Files to Upload: {ScanName}.{Technology}_{timestamp}.csv {ScanName}.{Technology}_{timestamp}.CloudReady.csv
Deployed/Build Scan	Not required
Scan Steps	1. Scan the source code 2. Upload the source code scan results 3. Submit the application results

CAST HIGHLIGHT

CODE SCAN **BEST PRACTICES**

CAST
Software Intelligence for Digital Leaders

	Software Composition Analysis
Source Code Scan	Include: - Source code - Dependency files (e.g. pom.xml, package.json, .vcsproj, etc.) Exclude: - Test - Generated code - Samples from third-party libraries - Deployment, SCM folders and files (e.g. .git, .snv, gradlew, .vscode, etc.) CAST Highlight files to upload: {ScanName}.{Technology}_{timestamp}.csv framework.validated.csv
Deployed/Build Scan	✔ Recommended (e.g. content of a WAR, installed folder in Windows, JARs, DLLs, etc.) CAST Highlight files to upload: BinaryLibraries.csv {ScanName}.{Technology}_{timestamp}.ThirdParties.csv
Scan Steps	1. Scan the source code 2. Scan the deployed/build output 3. Upload both scan results 4. Submit the application results

	All Use Cases
Source Code Scan	Include: - Source code - Dependency files (e.g. pom.xml, package.json, etc.) Exclude: - Third-party component sources (typically in “lib”, “third-party”, “3rd-party”, “COTS”, “external”, “node_modules” folders, etc.) - Tests - Generated code (e.g. t.ds, .flow.js) - Deployment, SCM folders and files (e.g. .git, .svn, gradlew, .vscode, etc.) CAST Highlight files to upload: {ScanName}.{Technology}_{timestamp}.csv {ScanName}.{Technology}_{timestamp}.CloudReady.csv framework.validated.csv
Build Output Scan	✔ Recommended (e.g. content of a WAR, installed folder in Windows, etc.) CAST Highlight files to upload: BinaryLibraries.csv {ScanName}.{Technology}_{timestamp}.ThirdParties.csv
Scan Steps	1. Scan the source code 2. Scan the deployed/build output 3. Upload both scan results 4. Submit the application results