



Coverity 2020.12 Release Notes

Copyright 2020 Synopsys, Inc.

Table of Contents

1. Coverity 2020.12 Release Notes	1
1.1. Important information for 2020.12	1
1.2. Coverity Platform 2020.12	1
1.3. Coverity Analysis 2020.12	6
1.4. Coverity Desktop 2020.12	23
1.5. Coverity Documentation 2020.12	25
A. Legal Notice	28
A.1. Legal Notice	28

Chapter 1. Coverity 2020.12 Release Notes

Table of Contents

1.1. Important information for 2020.12	1
1.2. Coverity Platform 2020.12	1
1.3. Coverity Analysis 2020.12	6
1.4. Coverity Desktop 2020.12	23
1.5. Coverity Documentation 2020.12	25

1.1. Important information for 2020.12

Support for this version of Coverity will be discontinued 18 months after the base version of this release.

All Coverity products, including the installers, support only ASCII characters for file and directory names. Non-ASCII characters, such as Japanese characters, are not supported for these names.

1.2. Coverity Platform 2020.12

This section provides release notes for Coverity Platform components.

1.2.1. Coverity Connect 2020.12

1.2.1.1. Deprecated products and features

SAT-29440

Use of the commit port (usually, port 9090) for the `cov-commit-defects` and `cov-run-desktop` command is deprecated. You should use the HTTPS port (or HTTP port for demo purposes) instead. See the `cov-commit-defects` documentation.

This also means that the following options of `cov-commit-defects` and `cov-run-desktop` are also deprecated, since they will be obsolete:

- `--dataport <port>`
- `--url commit://...`
- `--encryption <option>`

In addition, several other command-line options are deprecated, in favor of using the `--url` option:

- `--port`
- `--https-port`

- `--host`
- `--user >`
- `--password`

1.2.1.2. New or changed features

- The Coverity Connect silent installer option `--backup.destination` has been deprecated and replaced with `--backup.dir`. (COVDOCS-121)
- Added Bug Tracking System, Metrics and History, and Kerberos logging configuration via Web Services. (IM-23525)
- Coverity Connect has a new feature called Commit Over HTTPS. This feature lets the administrator provide all user services over the standard HTTPS port. Specifically, a commit (upload) of analysis results no longer requires a nonstandard port (typically, port 9090) on the Connect server. This simplifies deployment and makes Connect more compatible with typical firewall rules. This feature also improves commit reliability—a heartbeat between the client and server blocks proxies from terminating commit connections they presume are idle. (IM-25088)
- Added ability to retrieve issue occurrences using new REST APIs. (IM-25345)
- Added syntax highlighting for Go. (SAT-35405)
- When using both `cov-commit-defects` and a CIM server from this release, the `cov-commit-defects` client will, by default, send all data over the https port, and will not use the separate commit port, unless explicitly requested. (SAT-35621)

1.2.1.3. Bug fixes

COVDOCS-154

An issue was fixed for missing the `cov.css` (style sheet file) for the Web Service API reference.

IM-25271

Fixed a Coverity Connect File filter Issue.

IM-25315

Improved the replication of user data in Coverity Connect cluster. This should reduce the chances of replication not being able to proceed without manual intervention.

IM-25390

An issue was fixed that resulted from editing settings for a last snapshot.

IM-25400

The *Coverity Platform 2020.12 Web Services API Reference* has been updated to include a description of the Configuration Service's `userDataObj` complex type.

IM-25402

Fixed wrong occurrences count value in `occurrencemode`.

IM-25416

Fixed web services method `getStreamDefects` failing for some defects

INS-2955

An issue was resolved in which the backup directory parameter was not working for Coverity Connect silent installer.

RG-1482

For ease of use, we have added a message in the Connection page to show total issues for a particular snapshot.

1.2.1.4. Known issues and solutions

CPU-17

Downloading the binaries to update Java and/or PSQL for security fixes might fail on slow internet connections. Please make sure you have a fast internet connection and retry.

CPU-38

In order to use Coverity Connect with a mail server (https option) or Bugzilla (https option), and some other cases, the user has to import certificates into `cim/jre/lib/security/cacerts`. After running the updater, all of these certificates are gone.

IM-16076

Changing the summary metric name on a coordinator causes the summary metric to disappear from all reports on subscribers. To work around this issue, add the new summary metric back into the reports on subscriber

IM-17701

User and password information in `coverity_config.xml` do not override options specified on the command line.

IM-18707

Collisions might occur if triage data is deleted from a cluster (used for testing, for example), and then up-to-date triage data is imported from a production instance. This is because deleting triage stores does not delete related CIDs. It is recommended you rebuild the cluster from scratch using the production data.

IM-18710

In a cluster environment, deletion of triage data on the coordinator is not recommended unless it can be verified that there are no subscriber dependencies. Synchronization problems between subscribers and the coordinator might result.

IM-19048

The selected value is not displayed for a Coverity Connect field when using Chrome browser version 47.0.2526.80 on Windows 7.

IM-19685

Using a custom defect export handler script might on occasion create an error when attempting to export data to a bug tracking system.

IM-19690

To prevent database constraint violations on subscribers in a cluster, when a user is deleted, it is marked for deletion instead of being completely (hard) deleted. This status subsequently synchronizes across the cluster.

IM-23550

When configuring Coverity Connect to connect to an LDAP server, you must specify (in the Host Name field) the hostname of the machine hosting the LDAP server. Using the IP address of the LDAP server is not supported. For more information, refer to the section "Configuring LDAP server settings" in the *Coverity Platform 2020.03 User and Administrator Guide*.

IM-23755

The *Coverity Platform Web Services API Reference* has been clarified to point out that the `snapshotScope` parameter to the Defect Service's `getMergedDefectsForStreams` operation is optional.

IM-23994

Internet Explorer 11 breaks on functionalities using file upload.

IM-25194

Translations for standard attribute descriptions that are displayed when an issue is selected are not provided in this release.

IM-25329

When upgrading from a database in 2020.03 or 2020.06, two columns are shown for PCI DSS info: 1) Standard: Payment Card Industry Data Security Standard (PCI DSS) 2018 and 2) PCI DSS 2018. Use the information in the PCI DSS 2018 column for correct results.

INS-1274

Although the *Upgrade Guide* states that 32-bit to 64-bit in-place database format upgrades are not permitted, some will succeed, yielding valid results. Because in-place upgrade is preferable to backup-and-restore upgrade, we recommend that you try your upgrade in-place and, if it fails, fallback to backup-and-restore upgrade.

INS-1477

If Java 1.7.0_xx is used, and even if the system has a large amount of available RAM, using Java 1.7.0_xx and older, `Out of Memory` errors might occur despite having sufficient/available RAM. The workaround is to use the Java version shipped with Coverity Platform, or to specify a max heap setting for `cov-im-daemon`.

INS-2133

Due to a Red Hat Enterprise Linux issue (Bug 1484079), the Coverity Platform installer on Centos7 or RHEL v7.4 might fail due to an `ArrayIndexOutOfBoundsException` error and a stack trace indicating an error with fonts. This can be resolved by installing the `dejavu-serif-fonts` package.

INS-2307

For customers upgrading their Coverity Platform server from unsupported Coverity versions (such as version 5.x), we recommended that you upgrade to a supported intermediate version (such as 2018.03) before upgrading to 2018.06. We also recommended that you perform a backup of your data beforehand with the Upgrade Preparation feature.

INS-2648

All Coverity installers for Linux have a known issue related to missing fonts.

If you are installing a Coverity product on Linux from the command line, the installer might fail before asking for user input if the target system does not have access to the fonts required by the installer. Stack traces vary, but usually reference "fonts". You can work around this issue by installing the `fontconfig` package.

For example, this command uses the `apt-get` package manager to install `fontconfig`:

```
apt-get install fontconfig
```

This command uses the `yum` package manager to install `fontconfig`:

```
yum install fontconfig
```

1.2.2. Coverity Report Generators 2020.12

1.2.2.1. Bug fixes

COVDOCS-156

In the *Platform User and Administrator Guide*, the description of the OWASP Top Ten generator now documents all the `--password` options.

COVDOCS-167

In the *Platform User and Administrator Guide*, the description of the OWASP Top Ten generator now documents options that have been added, including `--auth-key-file`, an alternative to `--password`.

COVP-2284

Documentation has been updated to remove information about Synopsys certification for Integrity Report ratings because this service is no longer offered.

RG-1484

The 10,000 limit on reported issues has been raised to 50,000.

RG-1487

To resolve an issue where the report generator did not recognise CERT-Java data in the stream, we added support for CERT-Java analysis.

RG-1497

Fixed an issue where `CERT_POS**_C` Checkers were missing in CERT report.

RG-1500

Fixed a link to 2019 CWE/SANS that should go to 2020 CWE/SANS.

RG-1518

Documentation was updated to correct a command syntax error.

1.2.2.2. Known issues and solutions

RG-1128

For ATP-based systems, you might receive an error message during report generation. If you do receive an error message, you are likely missing these libraries: `libgl1`, `libgl1-mesa-dri`, and `libgl1-mesa-glx`. You can install the missing libraries by using the following command syntax:
`apt-get install libgl1, apt-get libgl1-mesa-dri, and apt-get libgl1-mesa-glx.`

RG-1142

During report generation, you might receive the following error: "Loading library `prism_es2` from resource failed: `java.lang.UnsatisfiedLinkError`:"

If you encounter this error message, please install these missing libraries: `apt-get install libgl1, apt-get libgl1-mesa-dri, and apt-get libgl1-mesa-glx.`

RG-1260

In the Security Report, "Issues Without CWE Numbers" has been renamed "Non-security Issues" to address a complaint about a mismatch between the reported count of issues without CWE numbers and Coverity Connect output sorted by `outstanding defects`.

RG-1271

The Security Report now points to BDBA instead of Protecode SC.

1.3. Coverity Analysis 2020.12

This section provides release notes for Coverity Analysis components.

1.3.1. Coverity Checkers 2020.12

For a summary of checkers that have been added or changed in this release, refer to the "Coverity Checker Change History" table in the *Coverity Checker Reference*.

1.3.1.1. New or changed features

- The `RISKY_CRYPTO` checker now supports Python. (SAT-21637)
- The `URL_MANIPULATION` checker now supports Python. (SAT-25956)
- The `HEADER_INJECTION` checker now supports Python. (SAT-25957)
- The `INSECURE_RANDOM` checker now supports Python. (SAT-25958)
- The `UNENCRYPTED_SENSITIVE_DATA` checker now supports Visual Basic. (SAT-26267)
- The new C/C++ `WRITE_CONST_FIELD` checker detects certain writes to const-qualified fields of structures, classes, or unions. (SAT-28173)

- Security checkers now support Python 3.x. (SAT-28958)
- CodeXM now supports GO. (SAT-29963)
- The `XPATH_INJECTION` checker now supports Go. (SAT-30368)
- Added additional `TRUST_BOUNDARY_VIOLATION` checker support for Spring Framework. (SAT-31438)
- Removed `NULL_RETURNS` false positives in `With me` construct of `VB.NET`. (SAT-31470)
- Added support for user modeling for Go. (SAT-33104)
- Added example compliance config for all compliance standards, where all the standard rules are listed in deviations. For instance, the config file containing all CERT-C standard rules as deviations can be found at `configs/cert-c/cert-c-all-deviations.config`. This will enable enforcement of specific rules by deleting their entries from these all-deviations sample config files. (SAT-34014)
- The `RISKY_CRYPTO` checker now reports defects when usage of weak TLS cipher suites is detected. (SAT-34086)
- Improved Spring Framework programmatic view resolution configuration support. Programmatically setting an XML dispatch servlet configuration file path(s) and a resolver prefix/suffix in Java is now supported by analysis. (SAT-34830)
- Modeled CUDA runtime memory management API. (SAT-34978)
- The `INSUFFICIENT_LOGGING` checker now supports Python. (SAT-35342)
- Spring `@ExceptionHandler` annotation is now supported as an entry point. (SAT-35501)
- CUDA specifier inconsistencies across declarations within a single translation unit are now reported under the `parse_warnings` subcategory of `CUDA.SPECIFIERS_INCONSISTENCY`. (SAT-35805)
- The `CONFIG.ENABLED_DEBUG_MODE` checker now supports Python. (SAT-36057)
- The new `MISSING_PASSWORD_VALIDATOR` checker finds cases where no password validators are set in the Django configuration file. (SAT-36160)
- The description of how to use Security Directives, both the JSON syntax used by these directives, and the semantics of the various directives themselves, has now been moved from an appendix in the *Checker Reference* to a book of its own, *Security Directive Reference*. (SAT-36162)

The contents of `parse_warnings.conf.sample` have been removed from the *Checker Reference* altogether, as this file is independently provided with each installation of Coverity Analysis. (SAT-36162)
- The new `HOST_HEADER_VALIDATION_DISABLED` checker finds cases where host header validation list is set to allow access from all hosts. (SAT-36248)
- A new library reference has been added to describe CodeXM support for the Go language. (SAT-36271)

- The new `CONFIG.WEAK_SECURITY_CONSTRAINT` checker finds cases where `<security-constraint>` elements for Java servlets in XML configuration files do not have proper authorization. (SAT-36297)
- The `WEAK_URL_SANITIZATION` checker now supports Java applications. (SAT-36298)
- The `INSECURE_COOKIE` checker now supports Python applications. (SAT-36371)
- The `WEAK_PASSWORD_HASH` checker now supports Python applications. (SAT-36372)
- The new `WEAK_XML_SCHEMA` checker finds insecure settings in XSD (XML Schema Definition) schema files. (SAT-36376)
- Checker options in the `CSRF` checker have been renamed: `url_whitelist` option is now `suppress_for_url`; `http_method_whitelist` option is now `suppress_for_http_method`; `http_method_blacklist` option is now `ignore_filters_for_http_method`. (SAT-36386)
- Changed the name of `whitelist` option of the `CALL_SUPER` checker to `use_must_call_list`. (SAT-36388)
- Changed the name of the `BLACKLIST_FOR_AUTHN` checker to `DENY_LIST_FOR_AUTHN`. Changed the name of `auth_blacklist` event to `auth_deny_list`. Changed the name of `csrf_blacklist` event to `csrf_deny_list`. (SAT-36424)
- The `INSECURE_COOKIE` checker now supports Python applications. (SAT-36427)
- JavaScript JSHint analysis has been upgraded to v2.12.0 (SAT-36434)
- The new Java `CONFIG.SPRING_BOOT_ADMIN_ACCESS_ENABLED` checker finds cases where the admin features are enabled in configuration files of Spring Boot applications. (SAT-36449)
- Extended the `CUDA.INVALID_MEMORY_ACCESS` checker to report on cases where a virtual method of a polymorphic object is called on a host or device where the object was not created. (SAT-36451)
- Substantive information has been added about handling tainted data issues in Chapter 6.8 "Tainted Data Overview" of the *Checker Reference Guide*. (SAT-36467)
- Substantive information has been added about handling sensitive data issues in Chapter 6, Section 9, "Sensitive Data Overview" of the *Checker Reference Guide*. (SAT-36468)
- The `HARDCODED_CREDENTIALS` checker now supports Python 3 Django applications. (SAT-36603)
- The new Python `CONFIG.DJANGO_CSRF_PROTECTION_DISABLED` checker finds cases where a `CsrfViewMiddleware` plugin is not enabled. (SAT-36612)
- The `CONFIG.HARDCODED_CREDENTIALS_AUDIT` checker now supports .Net applications. (SAT-36763)
- The `XPATH_INJECTION` checker now supports Go. (SAT-36774)

- Added support for 2 new SEI CERT C rules: ERR34-C and EXP47-C. (SAT-36826)
- The new `CONFIG.ANDROID_GRADLE_OBFUSCATION_NOT_ENABLED` checker finds cases where an Android application is configured to not enable code shrinking or code obfuscation in the release build. (SAT-36845)
- Added support for part of SEI CERT CPP rules for Clang-based compilers. (SAT-36910)

1.3.1.2. Bug fixes

CMPCPP-10825

A false positive has been fixed for AUTOSAR C++14 M0-1-1.

CMPJ-1464

References to `this` captured by Java and C++ lambda functions are now represented in event messages as `this` rather than `__coverity_captured_this`.

COVDOCS-134

The description of `issueType` in "The checker-definition" section of the CodeXM Syntax reference has been corrected and expanded in order to clarify how to set this value. A description of the `defineIssueType()` function has been added.

PRD-12149

An issue was fixed where analysis of a project in AS 3.6 failed.

SAT-26595

The `CSRF` checker now detects Spring `CSRF` protection enablement using the `WebSecurityConfig.configure` method.

SAT-31206

Fixed False Positive in `CONSTANT_EXPRESSION_RESULT` related to calls to methods with names containing string `equal` but having an opposite meaning like `notEqual`.

SAT-33324

The *Coverity Checker Reference* includes a new appendix for the Payment Card Industry Data Security Standard.

SAT-33355

Fixed a false positive for the `FORWARD_NULL` checker.

SAT-33424

Fixed a false positive for the `FORWARD_NULL` checker.

SAT-33440

Fixed a false positive for the C# `FORWARD_NULL` checker.

SAT-33561

Fixed a false positive for the `UNREACHABLE` checker not reporting on `__builtin_unreachable()`.

SAT-33670

An issue was fixed with the handling of url-patterns for `CSRF`.

SAT-34287

Fixed a false negative for golang: the `INFINITE_LOOP` checker was unable to identify an infinite loop for `golang: len` call in condition.

SAT-34968

Fixed False Positives in `UNSAFE_DESERIALIZATION` related to deserialization when using the Jackson framework.

SAT-34980

Fixed an issue by allowing `PASS_BY_VALUE` thresholds to trigger different errors based on specified sizes.

SAT-35860

Fixed a false negative for the `OVERFLOW_BEFORE_WIDEN` checker due to a sign extension issue.

SAT-35906

Fixed some formatting and spelling errors in `useless_continue` events produced by the `NO_EFFECT` checker.

SAT-35997

Neither built-in nor user-written C++ CodeXM checkers were being applied to CUDA source files. This has been fixed.

SAT-36094

Filtering capability of dynamic table in Chapter 2 of the "Checker Reference" has been fixed.

SAT-36097

A false negative for the `HARDCODED_CREDENTIALS` C# checker has been eliminated.

SAT-36158

Fixed an issue that could cause calls from bytecode to source not to be resolved when they involved C# generics, in particular when using `--resolve-calls-to-all-delegates true`.

SAT-36165

Fixed an issue in CodeXM with `--fnptr-models`.

SAT-36182

Fixed a false positive for `FORWARD_NULL` with `vector::resize`,

SAT-36186

Fixed a recoverable error with the message `assertion failed: endIndex > startIndex`.

SAT-36223

Fixed a false positive for `FORWARD_NULL` with `vector::resize`.

SAT-36292

False negatives in multiple checkers when using cookie values in GO have been fixed.

SAT-36304

An issue has been fixed for performance degradation when enabling `UNENCRYPTED_SENSITIVE_DATA` checker.

SAT-36305

An issue has been fixed for performance degradation when enabling `UNENCRYPTED_SENSITIVE_DATA`, `WEAK_GUARD`, `HARDCODED_CREDENTIALS`, or `WEAK_PASSWORD_HASH` for C/C++ checkers.

SAT-36336

Fixed a commit error from the `TAINT_ASSERT` checker.

SAT-36595

Fixed a false positive with AUTOSAR compliance rules on CUDA code, where we incorrectly claimed a symbol to be duplicated.

SAT-36610

Fixed a `cov-analyze` assertion because of unhandled events in the `INSUFFICIENT_LOGGING` checker for JavaScript.

SAT-36754

An issue has been fixed that caused a `cov-analyze` assertion when handling missing events in the `RISKY_CRYPT` checker for Go.

SAT-36900

Fixed a stack overflow in `CONFIG.SPRING_BOOT_SSL_DISABLED`.

SAT-37022

An issue that resulted in a `cov-analyze` crash when `HARDCODED_CREDENTIALS_BUDA_CPP` is used with `--enable-fnptr` has been fixed.

SAT-37052

Fixed an unrecoverable `cov-analyze` crash with some C++ constructs when some MISRA rules were enabled.

SATW-3744

Fixed a false positive of AUTOSAR C++14 A20-8-4 about ownership sharing as the return value of a function call.

SATW-3849

Fixed a false positive of MISRA C-2012 Rule 11.8 while casting a pointer to `const` to a pointer to a `const` array.

SATW-3875

Fixed a false positive of AUTOSAR C++14 A7-1-8 about `const` and `volatile`.

SATW-3877

Fixed a false positive of AUTOSAR C++14 A8-4-9 about non `const` in-out parameter not modified in one path.

SATW-3901

Fixed a false positive of AUTOSAR C++14 A5-16-1 when the ternary conditional operator had a constant conditional expression.

SATW-3902

Fixed a false positive of AUTOSAR C++14 A13-2-2 about overloading stream operators.

SATW-3909

Fixed a false positive of AUTOSAR C++14 A7-6-1 about `throw` operator with `--enable-exceptions` option in `cov-analyze` command.

SATW-3913

Fixed a false positive of MISRA C-2012 Rule 14.2 where there was only one side-effect in the first clause of a `for` loop, which was to set the loop counter.

SATW-3922

Fixed a false positive of AUTOSAR C++14 A9-6-1 while giving an alias to the underlying type of `enum`.

SATW-3925

Fixed a false positive of CERT STR30-C about casting a `string` literal to a pointer to a `const non-char` type.

SATW-3926

Fixed a false positive of AUTOSAR C++ A5-2-3 and CERT EXP55-CPP about type traits.

SATW-3945

Fixed a false positive in CERT STR51-CPP where a pointer had been checked against `NULL` in a callee.

SATW-3946

Fixed a false positive of AUTOSAR C++14 A15-4-4 about `extern "C"`.

SATW-3948

Fixed a false positive of CERT ERR08-J about `multi-catch` clause.

SATW-3955

Fix a false positive of MISRA C-2012 Rule 10.3 and MISRA C-2012 Rule 10.4 to take typedef names such as `BOOL_T` as essentially `boolean` types.

SATW-3957

Fixed a false positive of CERT EXP39-C about casting between base class and derived class.

SATW-3960

Fixed a false positive of CERT EXP37-C about casting `pointerType` to `referenceType`.

SATW-4003

Fixed a false negative of AUTOSAR C++14 A5-2-3 when applying to CUDA source files.

1.3.1.3. Known issues and solutions

BLC-833

When using Buildless Capture with JavaScript projects, in some cases analysis might yield a large number of false positives for the `EXPLICIT_THIS_EXPECTED` checker. In such cases, we

recommend disabling this checker using the `--disable EXPLICIT_THIS_EXPECTED` option for the `cov-analyze` command.

SAT-17490

Churn for the preview `INTEGER_OVERFLOW` checker might be higher in this release compared to churn for other checkers.

SAT-34445

The latest version of the integrated SpotBug software has a documented bug: `FE_FLOATING_POINT_EQUALITY` defects won't be reported

SAT-7224

The `XSS` checker can report multiple occurrences of the same local defect under certain circumstances.

1.3.2. Coverity Commands 2020.12

1.3.2.1. Deprecated products and features

COVP-2315

Support for FreeBSD 11.3 is deprecated as of 2020.12 and will be removed in a future release.

COVP-2317

Support for FreeBSD 12.0 is deprecated as of 2020.12 and will be removed in a future release.

COVP-2320

Support for Linux glibc versions 2.12-2.16 is deprecated and will be removed in a future release.

SAT-29440

Use of the commit port (usually, port 9090) for the `cov-commit-defects` and `cov-run-desktop` command is deprecated. You should use the HTTPS port (or HTTP port for demo purposes) instead. See the `cov-commit-defects` documentation.

This also means that the following options of `cov-commit-defects` and `cov-run-desktop` are also deprecated, since they will be obsolete:

- `--dataport <port>`
- `--url commit://...`
- `--encryption <option>`

In addition, several other command-line options are deprecated, in favor of using the `--url option`:

- `--port`
- `--https-port`

- `--host`
- `--user >`
- `--password`

1.3.2.2. New or changed features

- Added support for FreeBSD 11.4 (COVP-2312)
- Option added to `cov-make-library` to enable CGO files when creating models. (SAT-32475)
- When using both `cov-commit-defects` and a CIM server from this release, the `cov-commit-defects` client will, by default, send all data over the https port, and will not use the separate commit port, unless explicitly requested. (SAT-35621)

1.3.2.3. Bug fixes

CMPCPP-10166

We have fixed a false positive for a data race condition in `cov-emit`.

CMPCPP-10704

We have fixed an issue with `cov-emit` where the front end would abort on a nested generic lambda.

CMPJ-1391

Fixed `NullPointerException` leading to catastrophic failure in missing types mode in the Java front end.

IM-25351

Improved `cov-archive` import logic for users and user stream role assignments.

INS-2984

An issue has been fixed in which the `cov-install-updates` command was still reporting that there were updates needed for the latest version.

SAT-34738

Fixed output from the `cov-configure --help` command.

SAT-35475

Fixed an issue for `cov-analyze` not finishing analysis.

SAT-35911

Fixed an issue where the `cov-commit-defects` command would incorrectly accept self-signed certificates even when they were expired.

SAT-35922

In the context of a `SELECT ... CASE` statement, Fortran analysis was flagging a type mismatch between an integer selector expression and case expressions consisting of `BOZ` constants. This false positive pattern has been removed.

1.3.2.4. Known issues and solutions

CAP-332

If you receive the following error message when using `cov-build`, you can work around this issue by using the `--instrument` option.

```
[WARNING] Compilations that use 32-bit Java tools running on 64-bit Windows were detected during this build. Such compilations are not supported at the moment; analysis might be incomplete or invalid because of that.
```

Workaround: `> cov-build --dir t1 --instrument ant`

CAP-812

If you have KB2919355 (<http://support.microsoft.com/kb/2919355>) installed on Windows 2012 system, you might encounter the build hanging under `cov-build` if MSBuild is used. When this happens, the process tree will show MSBuild still running under `cov-build`, even though there will be no output or progress from MSBuild. To work around this issue, you can do one of the following: Uninstall KB2919355, or Add the `--instrument` flag to your `cov-build` invocation; for example:

```
> cov-build --dir dir --instrument msbuild ..
```

CMPCPP-4764

On Windows, when preprocessing a file with `cov-emit` to the Windows console, `cov-emit` might fail with a catastrophic error if the character encoding of the preprocessed output is not compatible with the console encoding. This error can be avoided by redirecting the preprocessed output to a file.

PRD-7595

When in the Test Prioritization workflow, on the View Results page, clicking the **Open in System Editor** button might not work for some older Linux distributions.

SAT-12174

Running `cov-emit-java` to emit a web application (with `--war --findears` or similar) might fail if the number of JAR files in its classpath (including those found with `--findjars`) exceeds the operating system's per-process file limit. To work around this case, either increase the per-process open file limit or remove unnecessary JARs from the classpath.

1.3.3. Coverity Compilers and Capture 2020.12

1.3.3.1. End-of-life products

CMPG-3253

Support for Swift 5.2 has been dropped as of 2020.12.

CMPG-3254

Support for Clang 3.0-3.6 has been dropped as of 2020.12

CMPG-3271

Support for GNU GCC and G++ version 3 has been dropped as of 2020.12

CMPG-3420

Support for Solaris 10 has been dropped as of 2020.12

CMPG-3483

Support for Apple Clang 6.1 has been dropped as of 2020.12

COVP-2295

Support for Solaris 10 has been dropped as at 2020.12

COVP-2321

Support for Windows Server 2012 has been dropped as at 2020.12

1.3.3.2. Deprecated products and features

CMPG-3255

Support for LLVM 3.7 is now deprecated.

CMPG-3311

Support for Python 2.7 is deprecated as of 2020.12 and will be removed in a future release.

CMPG-3423

Support for Renesas C/C++ M32R 5.01 is deprecated as of 2020.12

CMPG-3459

Support for PHP 5.x has been deprecated.

CMPG-3476

Support for Apple Clang 7.0–7.2 is deprecated as of 2020.12 and will be removed in a future release.

CMPG-3505

Support for Xbox One is now deprecated.

CMPG-3506

Support for ARM C/C++ for Nintendo 3DS is now deprecated.

CMPG-3507

Support for Freescale Codewarrior 10.5 for MSC815x is now deprecated.

CMPG-3508

Support for Analog Devices Compiler version 8.5 and any versions prior to 8.12 on Blackfin are now deprecated.

COVP-2298

Support for Oracle JDK 14 has been deprecated as of 2020.12 and will be removed in a future release.

COVP-2300

Support for Open JDK 14 has been deprecated as of 2020.12 and will be removed in a future release.

1.3.3.3. New or changed features

- Added support for Microchip XC16 version 1.50 compiler on Windows and Linux. (CMPCPP-10101)
- Arguments for the `nvcc -Xcompiler` and `--compiler-options` options are now shell expanded to match `nvcc` behavior. (CMPCPP-10143)
- Added support for Green Hills 68K/ColdFire compiler 2012.1. (CMPCPP-10218)
- Added support for HighTec Tricore compiler 4.9.3.0. (CMPCPP-10340)
- Added support for the Texas Instruments C7000 version 1.2.0 compiler on Linux. (CMPCPP-10420)
- Added support for AVR MCU targets of Microchip XC8 version 2.20 compiler on Windows and Linux. (CMPCPP-10475)
- Added support for the CEVA-XC12 version b480 compiler. (CMPCPP-9813)
- Added support for the Clang 11 C/C++ compiler. (CMPG-3401)
- Java support added for the `--record-with-source` and `--replay-from-emit` options to the `cov-build` command. (CMPJ-1309)
- The JavaScript front end now supports all ES11 (ECMAScript 2020) syntax. (CMPJS-768)
- The JavaScript front end now supports ES11 dynamic import expressions. (CMPJS-804)
- The JavaScript front end now supports ES11 dynamic import expressions. (CMPJS-810)
- Build capture for .NET Core is now supported. (COVDOCS-169)
- Added SCM support for Accurev 7.4. (COVP-2231)
- Added support for NetBSD 9.0. (COVP-2291)
- Added support for Swift 5.3. (SATPLAN-223)

1.3.3.4. Bug fixes

CMPCPP-10309

The addition of support for C++20 Concepts fixes various parsing errors that would happen with code that defined and used `Concepts`.

CMPCPP-10375

Resolved an issue where an alias template could be used in the wrong context causing a crash.

CMPCPP-10548

Fixed an issue where Coverity failed to translate `-mv` options for the Qualcomm Hexagon clang compiler.

CMPCPP-10593

Fixed a bug where a variable in a GNU statement expression could not be marked `constexpr`.

CMPCPP-10606

Basic support has been added for ARM SVE types for Clang based compilers.

CMPCPP-10608

Resolved a problem when two distinct header files generated the same unique ID, resulting in one of them not being included.

CMPCPP-10612

Fixed "Encountered an unexpected error node" assertion failure in c++17 mode due to exception specification incorrectly treated as a `SFINAE` context.

CMPCPP-10626

Command-line translation actions passed to `cov-configure --xml-option` that apply to the translate and pre-translate phases of command line translation are now correctly processed when compiling for CUDA.

CMPCPP-10627

Fixed an issue where `cov-emit` can't recognize short enums in specified targets for the Qualcomm Kalimba C compiler.

CMPCPP-10647

Support for 128-bit integers is now unconditionally enabled for the `nvcc` compiler and arguments to the `-Xcompiler` and `--compiler-options` options that begin with `-std=` and contain additional host compiler options are no longer misinterpreted.

CMPCPP-10657

Fixed an assertion failure when initializing an object with an initializer list within a copy constructor.

CMPCPP-10695

Fixed a segmentation fault that happened in clang front-end while parsing a program with function templates that have errors.

CMPCPP-10781

Fixed an issue where `cov-wizard` can't select compiler type for the Qualcomm Kalimba C compiler.

CMPCPP-10788

Arguments to the `nvcc -Xcompiler` and `--compiler-options` options that begin with `-std=` and contain additional host compiler options are no longer misinterpreted.

CMPCPP-10806

Fixed failure to emulate gcc with language standard c++03 when compiling templates with a default argument.

CMPCPP-10807

Fixed performance issue with compiling boost headers introduced in 2020.09.

CMPCPP-10822

Fixed a build hang.

CMPCPP-10919

Fixed a false positive of MISRA C 2004 Rule 8.9 where `_asm()` is called.

CMPCPP-10957

Fixed `cov-emit` Internal error #2688, assertion failure in "lower_name.c", line 9863 in `record_substitution_for_type`.

CMPCPP-4599

Fixed compatibility with GCC where `__INCLUDE_LEVEL__` predefined macro was not set.

CMPCPP-8606

Fixed an issue where if a designated initializer and a function pointer are involved in the same function, a crash occurs.

CMPCPP-8719

Translation of C++17 structured bindings and C99 VLAs used in C++11 range-based for statements has been corrected for Clang based compilers. Previously, these combinations resulted in assertion failures during translation.

CMPCPP-9769

An assertion failure that occurred when two C++ class template partial specializations for the same primary class template were given the same mangled name was relaxed to prevent translation failures.

CMCFG-431

Fixed a Kotlin front-end crash related to Kotlin serialization compiler plugin.

CMCFG-434

Fixed a Kotlin front-end crash related to method and class names containing spaces.

CMFJ-1336

Improved support for compiling modular Java projects, including the Android Open Source Project.

CMFJ-1431

Improved performance of emitting the JDK 9+ system image on Windows.

CMPOCCPP-225

Computation of line counts for Clang's block literals was corrected. Previous computation could result in a negative line count.

CMPSWIFT-449

Fixed assertion failure causing low emit rate on Coverity Swift 5.2 compiler.

COVDOCS-152

Documentation for the `<file_exclude_pattern>` and `<file_include_pattern>` tags has been added to section "5.3.1 The `<compiler>` tags" in the *Coverity Analysis 2020.12 User and Administrator Guide*.

INS-2982

An issue was fixed for uninstalling CC on Linux.

SAT-35921

Some compilers accept array initializers with character constants of varying length. Coverity Fortran Syntax Analysis has been modified to emulate this behavior in Intel Fortran Compilers version 15 and later.

SAT-36392

An issue has been fixed that caused a `cov-analyze` stack overflow crash with Swift infinitely recursive class hierarchy.

1.3.3.5. Known issues and solutions

CAP-1176

`cov-build --instrument` has a known issue when running the `xdcmake.exe` tool of VisualStudio 2010 when launched from a 32-bit process on Windows 10. This will currently fail with a `System.BadImageFormatException` exception. To work around this issue you can do one of the following: Modify the build such that `xdcmake.exe` is run from a 64-bit process, or ignore the `xdcmake.exe` process by adding `--capture-ignore xdcmake.exe` to your `cov-build` invocation.

CAP-1650

When using JDK 14 on mac OS 10.14 or 10.15 `cov-build` might miss capturing Java source. In this situation, please use buildless capture (`cov-capture`) to capture your Java source.

CMPG-3115

Casts of ISO/IEC TR 18037 fixed point types are incorrectly rejected in code compiled in C++ mode for Clang based compilers. This issue is known to affect the Synopsys MetaWare ccac compiler.

CMPG-3156

The new build system introduced in Xcode 10 is not supported with Clang compilers. See the section "Building projects that use Xcode 10's new build system" in the "Coverity Analysis User and Administrator Guide" for details on how to work around this issue.

CMPG-3322

Coverity Swift front end does not support Mac Catalyst apps in 2020.06 release.

CMPJ-368

The default `charset` for Java 1.8 VM on Mac appears to be UTF-8 if a `charset` has not been explicitly set. The Coverity Java compiler does not emulate this behavior. Make sure to explicitly set the character encoding by setting a locale using the `LANG` or `LC_CTYPE` environment variables

CMPJS-286

The JavaScript front end no longer supports nameless function statements. (Nameless function expressions are supported as before.) A function statement without a declared name is a syntax error according to the ECMAScript standard, but may be used in JavaScript source files with some frameworks.

CMPJS-796

The Coverity front end for TypeScript does not presently respect `module=esnext`. As a result, Coverity tools cannot currently emit top-level awaits which are built using `module=esnext`.

CMPSCA-187

Scala Macro Paradise compiler plugin can be incompatible between different Scala 2.12.x patch versions and might cause emit failures.

COVDOCS-124

If you are building ant projects from Netbeans, there is a failure to capture emitted files when launching Netbean build inside of `cov-build`. To fix the issue set Ant Javac Task `fork` attribute to `yes/true`. This tells ant to execute the OS level compiler externally. By default this is set to `no`, which means that it compiles intrinsically, and as a result, `cov-build` wont see javac invocations at the OS level.

1.3.4. Coverity Dynamic Analysis 2020.12

1.3.4.1. Bug fixes

SAT-36678

Fixed a crash when running the security dynamic analysis on an Android application.

1.3.4.2. Known issues and solutions

JDA-681

If Dynamic Analysis reports defects in classes that were compiled without debugging information, or classes that contain mangled information due to misbehaving code coverage or AOP tool, the defect report might contain nonsensical line numbers or file names.

JDA-694

Specifying certain combinations of the `instrument-arrays`, `instrument-collections`, `detect-races`, and `detect-deadlocks` options to the Dynamic Analysis agent causes unexpected behavior. In particular, Dynamic Analysis still reports races on arrays and collections according to the `instrument-arrays` and `instrument-collections` options when `detect-races` is false and `detect-deadlocks` is true. However, if both `detect-races` and `detect-deadlocks` are false, Dynamic Analysis reports races on neither collections nor arrays.

JDA-720

If you do not specify a class in the `cov-start-da-brokerclasspath` option, the corresponding source file isn't committed, even if the source file is present on the source path.

1.3.5. Coverity Test Advisor 2020.12

1.3.5.1. End-of-life products

COVP-2306

Support for Perforce 2016.2 has been dropped.

1.3.5.2. Deprecated products and features

COVP-2305

Support for Perforce 2017.1 is deprecated as of 2020.12 and will be removed in a future release.

TADE-2077

Support for Test Advisor Dev Edition is deprecated as of 2020.12 and will be removed in a future release.

1.3.5.3. New or changed features

- SCM support for Mercurial 5.2–5.5 has been added. (COVP-2301)
- Added SCM support for Perforce 2019.2, 2020.1 (COVP-2302)
- Added SCM support for SVN 1.13 - 1.14. (COVP-2303)

1.3.5.4. Known issues and solutions

TADE-2033

The use of `--cs-coverage opencover` with Test Advisor might fail to capture any tests or coverage data on some versions of Windows if the user's account has Administrator permissions, .NET Framework 4.8 is installed, and user account control (UAC) is disabled.

Workaround: manually register the OpenCover profiler DLLs and pass `--cs-no-register-profiler` to your `cov-build --test-capture` invocation. This manual registration must be performed systemwide; your `regsvr32` invocations must be run *without* the `/i:user` argument. For more details, see the documentation of `cov-build`'s `--cs-no-register-profiler` switch in the *Coverity Command Reference*.

TADE-2043

When using `--java-coverage jacoco`, Test Advisor might consider lines that never run to completion, but instead always generate exceptions, to be uncovered.

1.3.6. Coverity Wizard 2020.12

1.3.6.1. Known issues and solutions

PRD-11727

`cov-wizard` might not emit Java successfully with the default version that is installed in Ubuntu 18.04. (See <https://bugs.launchpad.net/ubuntu/+source/openjdk-lts/+bug/1796027>) To fix this issue, install a different version of Java and set it as the default Java version.

PRD-5290

In the Coverity Wizard Policy Editor, the *Link to Editor* icon in the Outline View might be toggled as enabled, even though the editor is not actually linked with the Outline View. To enable outline linking, toggle the **Link to Editor** button to disabled, and back to enabled again.

PRD-5387

Not all the Preference dialog text is translated into Japanese on the syntax coloring dialog.

PRD-5770

In Coverity Wizard, after automatically configuring the compilers in the Configure Compilers screen, the status indicator for the Configure Compilers screen might not update from the exclamation mark icon to the check mark icon, which will appear as though the auto-configuration was unsuccessful.

However, clicking anywhere in the Coverity Wizard window or changing pages will cause the indicator to update to the check mark icon.

PRD-6760

The *Guided Test Advisor Policy Creation Wizard* uses Java regex validation instead of the Perl regex validation that Coverity Analysis Test Advisor uses. This should not cause any issues for most users, but if there is a difference, go to the more advanced *Test Prioritization Policy Editor and Debugger* to enter the proper regex.

PRD-6832

The guided policy creation wizard **Documentation** link fails to open properly on Linux. Open the *Coverity Wizard 2019.12 User Guide* separately to view this documentation.

PRD-8227

After upgrade, Coverity Wizard can sometimes give a `ReferenceMap NullPointerException` application error on startup. To work around this issue, delete the `.orphan` file in the `<install_dir_sa>/jars/cwiz/configurations/org.eclipse.core.runtime` folder.

PRD-8453

When using a self-signed certificate, if the user chooses not to trust a certificate, they might be prompted multiple times (asking to trust the certificate). If a user does not want to trust a self-signed certificate, they should change their Coverity Connect server settings to avoid the prompts. But just keep pressing **no** (to not trust the certificate), to get through the multiple prompts.

PRD-9208

Coverity Wizard now warns the user every time they select the `Test Prioritization'` workflow, even if they did not first work with the regular analysis workflow. This can be safely ignored

PRD-9245

Using the **Duplicate** button for configuring compilers in Coverity Wizard does not work.

1.4. Coverity Desktop 2020.12

This section provides release notes for Coverity Desktop components.

1.4.1. Coverity Desktop for Android Studio 2020.12

1.4.1.1. Bug fixes

PRD-12166

Fixed an issue with Android Studio 3.6.

1.4.1.2. Known issues and solutions

PRD-12153

Even though we have added support for Android Studio 3.6, Coverity Desktop plugin will fail to scan Android projects. It will work for Java and Gradle projects that are not Android based.

PRD-7991

Android Studio does not show the proper `scope` in the Issues view for local analysis. It just always says "External output file" currently when in local analysis mode.

PRD-8042

Currently any source generated by Gradle Android projects will not be captured by the build process, and will be reported as "Uncaptured" by the IntelliJ and Android Studio IDEs. These files can be ignored by the "Uncaptured Source Files Dialog" or through the "File Exclusions" settings page. Auto-generated Gradle source files are captured when using Android Studio 3+.

PRD-8397

Coverity markers in the editor gutter can sometimes be shown in duplicate with the IntelliJ/AndroidStudio Coverity Desktop plug-in.

1.4.2. Coverity Desktop for Eclipse 2020.12

1.4.2.1. End-of-life products

PRD-12270

Support for Eclipse 4.7 has been dropped in this release.

1.4.2.2. New or changed features

- Added support for Eclipse 2020-09 (4.17) (PRD-12191)

1.4.2.3. Known issues and solutions

PRD-10694

For OXS 10.14 users with JDK-8136913 installed, using the `hostname_regex` in the `coverity.conf` file caused a 5 to 30 second delay. We've provided a workaround to fix this issue in our documentation.

PRD-10711

Eclipse customers using Plastic SCM might see a failure during Analyze Modified Files, as Eclipse is unable to locate their `cm` executable file. This occurs when the `cm.exe` file is located in `/usr/local/bin/` rather than `/usr/bin/` and can be resolved by adding a link to the executable in `/usr/bin/`.

1.4.3. Coverity Desktop for IntelliJ IDEA 2020.12

1.4.3.1. Deprecated products and features

PRD-12271

Support for IntelliJ 2017.2 has been deprecated in this release.

1.4.3.2. New or changed features

- Documentation has been updated to reflect a change in the implementation of Android Studio support: With this new implementation, we no longer rely on the Coverity Gradle plugin to support Android Studio. (PRD-12269)

1.4.3.3. Known issues and solutions

PRD-10076

When using whole program checkers in IntelliJ, a warning about missing class files might be displayed in the console, which indicates missing class files with incorrect paths. Even if the paths do not seem correct, this should not affect analysis results

PRD-10553

For Coverity Connect users using the Japanese locale, the **Apply** button in the triage panel was disabled unless the Owner was changed. To work around this, the IDE locale should be the same as the user account locale on the Coverity Connect server. Since IntelliJ currently only supports English, the user account locale on Coverity Connect must be set to English as well

PRD-7453

Coverity Connect attributes and usernames in the Coverity Desktop plug-in are cached on start up, and not refreshed until IntelliJ is restarted. If you are missing a new username, or some other triage attribute, try restarting IntelliJ.

PRD-7980

The Coverity Desktop plug-in does not currently work for the Alloy IDEA theme.

PRD-7991

Android Studio does not show the proper `scope` in the Issues view for local analysis. It just always says "External output file" currently when in local analysis mode.

PRD-8038

The triage view will not resize while the History section is expanded. Collapsing the history section will cause the view contents to resize.

PRD-8042

Currently any source generated by Gradle Android projects will not be captured by the build process, and will be reported as "Uncaptured" by the IntelliJ and Android Studio IDEs. These files can be ignored by the "Uncaptured Source Files Dialog" or through the "File Exclusions" settings page. Auto-generated Gradle source files are captured when using Android Studio 3+.

PRD-8397

Coverity markers in the editor gutter can sometimes be shown in duplicate with the IntelliJ/AndroidStudio Coverity Desktop plug-in.

1.5. Coverity Documentation 2020.12

This section provides release notes for Coverity Documentation components.

1.5.1. Coverity Documentation 2020.12

1.5.1.1. New or changed features

- In the Command Reference, new options for `cov-run-desktop` have been added. (COVDOCS-151)

- Help Center contents have been rewritten and reorganized. The document now begins with an overview of Coverity and static analysis. A second part lists and describes all current Coverity documentation. (COVDOCS-155)
- In the *Command Reference*, the description of `cov-make-library` has been updated to include the new `--enable-cgo-for-go-models` option. In the *Checker Reference*, chapter 5 and various checker descriptions now describe newly available models and primitives for Go-language source. (COVDOCS-160)
- In the *Coverity Analysis User and Administrator Guide*, the table of Language Support has been updated to show enhanced support for CUDA programs. (COVDOCS-173)
- The description of how to use Security Directives, both the JSON syntax used by these directives, and the semantics of the various directives themselves, has now been moved from an appendix in the *Checker Reference* to a book of its own, *Security Directive Reference*. (SAT-36162)

The contents of `parse_warnings.conf.sample` have been removed from the *Checker Reference* altogether, as this file is independently provided with each installation of Coverity Analysis. (SAT-36162)

1.5.1.2. Bug fixes

COVDOCS-147

The documentation has been updated to clarify that Coverity Connect supports TLS v1.2 and that client tools that invoke Coverity Connect should also be TLS v1.2-compliant. For example, OpenSSL requires version 1.0.1 or newer, and cURL requires version 7.3.4.0 or newer.

COVDOCS-149

The *Coverity Platform 2020.12 User and Administrator Guide* has been updated to clarify that when configuring integration with Jira, you must enter an API token into the Password field rather than a password. Consult the Atlassian Jira documentation for information on how to create an API token.

COVDOCS-152

Documentation for the `<file_exclude_pattern>` and `<file_include_pattern>` tags has been added to section "5.3.1 The `<compiler>` tags" in the *Coverity Analysis 2020.12 User and Administrator Guide*.

COVDOCS-99

The *Platform User and Administrator Guide* has been updated to show an introductory screen shot for those report generators that have a graphical interface: Coverity CERT, Coverity MISRA, Coverity Security, and Synopsys Software Integrity.

IM-23753

An issue was fixed in which the wrong version was shown for the *Coverity Analysis Administration Guide* in the Japanese version.

IM-25467

The "cov-admin-db — Administer Coverity Connect" section in the *2020.12 Command Reference* has been updated to clarify that when performing backups and restores, files are both faster and more convenient to work with than directories.

SAT-35438

Fixed an issue in Japanese version of *Coverity Checker Reference* which did not display Issues/Impacts/Checkers table in Chapter 2.

SAT-36577

A link to the *Command Reference* has been restored.

SAT-36659

The *Command Reference* was updated by removing five instances of the name "Prevent", which was the previous name for Coverity.

SAT-36887

In *CodeXM Syntax Reference*, the description of the `globalset-type` has been updated for greater completeness and clarity.

SAT-37118

In *CodeXM Syntax Reference*, the description of the `globalset-type` has been updated for greater completeness and clarity.

1.5.1.3. Known issues and solutions

SAT-26758

No HTML files are available for the following: *Coverity_CodeXM_C_C++_Library_Reference.pdf*, *Coverity_CodeXM_QuickStart_Tutorial.pdf*, *Coverity_CodeXM_Syntax_Reference_Guide.pdf*, *fortran_syntax_analysis_guide.pdf*.

Appendix A. Legal Notice

Table of Contents

A.1. Legal Notice	28
-------------------------	----

A.1. Legal Notice

The information contained in this document, and the Licensed Product provided by Synopsys, are the proprietary and confidential information of Synopsys, Inc. and its affiliates and licensors, and are supplied subject to, and may be used only by Synopsys customers in accordance with the terms and conditions of a license agreement previously accepted by Synopsys and that customer. Synopsys' current standard end user license terms and conditions are contained in the `cov_EULM` files located at `<install_dir>/doc/en/licenses/end_user_license`.

Portions of the product described in this documentation use third-party material. Notices, terms and conditions, and copyrights regarding third party material may be found in the `<install_dir>/doc/en/licenses` directory.

Customer acknowledges that the use of Synopsys Licensed Products may be enabled by authorization keys supplied by Synopsys for a limited licensed period. At the end of this period, the authorization key will expire. You agree not to take any action to work around or override these license restrictions or use the Licensed Products beyond the licensed period. Any attempt to do so will be considered an infringement of intellectual property rights that may be subject to legal action.

If Synopsys has authorized you, either in this documentation or pursuant to a separate mutually accepted license agreement, to distribute Java source that contains Synopsys annotations, then your distribution should include Synopsys' `analysis_install_dir/library/annotations.jar` to ensure a clean compilation. This `annotations.jar` file contains proprietary intellectual property owned by Synopsys. Synopsys customers with a valid license to Synopsys' Licensed Products are permitted to distribute this JAR file with source that has been analyzed by Synopsys' Licensed Products consistent with the terms of such valid license issued by Synopsys. Any authorized distribution must include the following copyright notice: **Copyright © 2020 Synopsys, Inc. All rights reserved worldwide.**

U.S. GOVERNMENT RESTRICTED RIGHTS: The Software and associated documentation are provided with Restricted Rights. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in subparagraph (c)(1) of The Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of Commercial Computer Software – Restricted Rights at 48 CFR 52.227-19, as applicable.

The Manufacturer is: Synopsys, Inc. 690 E. Middlefield Road, Mountain View, California 94043.

The Licensed Product known as Coverity is protected by multiple patents and patents pending, including U.S. Patent No. 7,340,726.

Trademark Statement

Coverity and the Coverity logo are trademarks or registered trademarks of Synopsys, Inc. in the U.S. and other countries. Synopsys' trademarks may be used publicly only with permission from

Synopsys. Fair use of Synopsys' trademarks in advertising and promotion of Synopsys' Licensed Products requires proper acknowledgement.

Microsoft, Visual Studio, and Visual C# are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Microsoft Research Detours Package, Version 3.0.

Copyright © Microsoft Corporation. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or affiliates. Other names may be trademarks of their respective owners.

"MISRA", "MISRA C" and the MISRA triangle logo are registered trademarks of MISRA Ltd, held on behalf of the MISRA Consortium. © MIRA Ltd, 1998 - 2013. All rights reserved. The name FindBugs and the FindBugs logo are trademarked by The University of Maryland.

Other names and brands may be claimed as the property of others.

This Licensed Product contains open source or community source software ("**Open Source Software**") provided under separate license terms (the "**Open Source License Terms**"), as described in the applicable license agreement under which this Licensed Product is licensed ("**Agreement**"). The applicable Open Source License Terms are identified in a directory named `licenses` provided with the delivery of this Licensed Product. For all Open Source Software subject to the terms of an LGPL license, Customer may contact Synopsys at `software-integrity-support@synopsys.com` and Synopsys will comply with the terms of the LGPL by delivering to Customer the applicable requested Open Source Software package, and any modifications to such Open Source Software package, in source format, under the applicable LGPL license. Any Open Source Software subject to the terms and conditions of the GPLv3 license as its Open Source License Terms that is provided with this Licensed Product is provided as a mere aggregation of GPL code with Synopsys' proprietary code, pursuant to Section 5 of GPLv3. Such Open Source Software is a self-contained program separate and apart from the Synopsys code that does not interact with the Synopsys proprietary code. Accordingly, the GPL code and the Synopsys proprietary code that make up this Licensed Product co-exist on the same media, but do not operate together. Customer may contact Synopsys at `software-integrity-support@synopsys.com` and Synopsys will comply with the terms of the GPL by delivering to Customer the applicable requested Open Source Software package in source code format, in accordance with the terms and conditions of the GPLv3 license. No Synopsys proprietary code that Synopsys chooses to provide to Customer will be provided in source code form; it will be provided in executable form only. Any Customer changes to the Licensed Product (including the Open Source Software) will void all Synopsys obligations under the Agreement, including but not limited to warranty, maintenance services and infringement indemnity obligations.

The Cobertura package, licensed under the GPLv2, has been modified as of release 7.0.3. The package is a self-contained program, separate and apart from Synopsys code that does not interact with the Synopsys proprietary code. The Cobertura package and the Synopsys proprietary code co-exist on the same media, but do not operate together. Customer may contact Synopsys at `software-integrity-support@synopsys.com` and Synopsys will comply with the terms of the GPL by delivering to Customer the applicable requested open source package in source format, under the GPLv2 license. Any Synopsys proprietary code that Synopsys chooses to provide to Customer upon its request will be provided in object form only. Any changes to the Licensed Product will void all

Coverity obligations under the Agreement, including but not limited to warranty, maintenance services and infringement indemnity obligations. If Customer does not have the modified Cobertura package, Synopsys recommends to use the JaCoCo package instead.

For information about using JaCoCo, see the description for `cov-build --java-coverage` in the *Command Reference*.

LLVM/Clang subproject

Copyright © All rights reserved. Developed by: LLVM Team, University of Illinois at Urbana-Champaign (<http://llvm.org/>). Permission is hereby granted, free of charge, to any person obtaining a copy of LLVM/Clang and associated documentation files ("Clang"), to deal with Clang without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of Clang, and to permit persons to whom Clang is furnished to do so, subject to the following conditions: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution. Neither the name of the University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from Clang without specific prior written permission.

CLANG IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH CLANG OR THE USE OR OTHER DEALINGS WITH CLANG.

Rackspace Threading Library (2.0)

Copyright © Rackspace, US Inc. All rights reserved. Licensed under the Apache License, Version 2.0 (the "License"); you may not use these files except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

SIL Open Font Library subproject

Copyright © 2020 Synopsys Inc. All rights reserved worldwide. (www.synopsys.com), with Reserved Font Name fa-gear, fa-info-circle, fa-question.

This Font Software is licensed under the SIL Open Font License, Version 1.1. This license is available with a FAQ at <http://scripts.sil.org/OFL>.

Apache Software License, Version 1.1

Copyright © 1999-2003 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgement: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)."

Alternately, this acknowledgement may appear in the software itself, if and wherever such third-party acknowledgements normally appear.

4. The names "The Jakarta Project", "Commons", and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache" nor may "Apache" appear in their names without prior written permission of the Apache Group.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License Version 2.0, January 2004 <http://www.apache.org/licenses/>
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at: <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Results of analysis from Coverity and Test Advisor represent the results of analysis as of the date and time that the analysis was conducted. The results represent an assessment of the errors, weaknesses and vulnerabilities that can be detected by the analysis, and do not state or infer that no other errors, weaknesses or vulnerabilities exist in the software analyzed. Synopsys does NOT guarantee that all errors, weakness or vulnerabilities will be discovered or detected or that such errors, weaknesses or vulnerabilities are discoverable or detectable.

SYNOPSYS AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, CONDITIONS AND REPRESENTATIONS, EXPRESS, IMPLIED OR STATUTORY, INCLUDING THOSE RELATED

TO MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, SATISFACTORY QUALITY, ACCURACY OR COMPLETENESS OF RESULTS, CONFORMANCE WITH DESCRIPTION, AND NON-INFRINGEMENT. SYNOPSIS AND ITS SUPPLIERS SPECIFICALLY DISCLAIM ALL IMPLIED WARRANTIES, CONDITIONS AND REPRESENTATIONS ARISING OUT OF COURSE OF DEALING, USAGE OR TRADE.