SYNOPSYS®

Safety Manual for 2020.12 Coverity and Test Advisor

Table of Contents

Chapter 1. Introduction

This manual is of special interest to Coverity and Test Advisor customers who are developing safety-related software.

Coverity
> Coverity is a tool used to find, manage, and fix software issues within source code. Issues are found "statically" by Coverity, as a result of analysis on source code. Issues discovered by Coverity can be related to the general quality or security of the software that it analyzes. As such, Coverity should be considered an integral part of comprehensive testing and verification activities for safety-related development. Of note, Coverity can be used to determine whether the code is compliant with MISRA standards, including MISRA C 2004, MISRA C++ 2008, and MISRA C 2012. A list of MISRA rules and directives covered by Coverity is available in Appendix A of *Coverity 2020.12 Checker Reference* ("MISRA Rules and Directives").

Test Advisor
> Test Advisor is a software tool that prioritizes deficiencies in test coverage by combining code analysis with the output of standard code coverage tools. Prioritization is integral to developing safety-related software because it identifies areas of code that have a critical need to be tested. Test Advisor is documented in *Test Advisor 2020.12 User and Administrator Guide* .

Chapter 2. Third-party Software

Coverity and Test Advisor include third-party software described in Appendix A, *Legal Notice*.

> **Synopsys makes no claim about comprehensive correctness of the third-party software packages it includes with Coverity and Test Advisor**, though it performs extensive testing of these third-party packages within the context in which they are used. Additional third-party packages are included in the products as necessary, and updates to the third-party packages are made as a result of known enhancements from which they can benefit, or due to quality or security enhancements made in the third-party tools.

Test Advisor can also be dependent on third-party tools, such as the Cobertura code coverage tool. A list of these tools is available in *Coverity 2020.12 Installation and Deployment Guide* (see "Test Advisor supported compilers and platforms").

> **Synopsys makes no claims about the accuracy of code coverage tools, such as Cobertura, but has tested to its satisfaction that the precision of the code coverage results is adequate to the reporting requirements of Test Advisor.**

Chapter 3. Failure Modes

Table of Contents

Failure modes of Coverity and Test Advisor generally range from complete inoperability (most commonly from compiler configuration errors) to inaccurate reporting of issues due to the use of inappropriate settings or unsupported compilers.

Users of Coverity and Test Advisor tools (including developers, managers, and administrators) should be aware of potential misuses of the tool, which are described in the following sections.

3.1. Coverity SAVE Analysis Engine

- Use of inappropriate checker settings, which might lead the tool to ignore defects. For example, modfiying checker options or command line options might affect what the checkers report.

    The user is responsible for checking the correct checker settings.

    For details about settings available to individual checkers, see *Coverity 2020.12 Checker Reference*. For example, see the MISRA_CAST ⤴ checker options.

- Changes to analysis settings that might result in issues being falsely reported as no longer present. For example, you might introduce such changes by using models, directives, or code-line annotations to model an API; such changes can introduce false negatives or false positivies if you don't model the API correctly.

    The user is responsible for checking the correct analysis settings.

    The analysis settings must be kept constant over a period of time to maintain a clear baseline of issues. For details about these settings, see *Coverity 2020.12 Command Reference* ⤴ as well as *Coverity Analysis 2020.12 User and Administrator Guide* sections, such as "Analyzing source code from the command line" ⤴ and "Enabling Checkers" ⤴ for various analysis workflows.

- If you build your own checker using CodeXM or Extend, or if you use the customizable checkers (`TEXT.CUSTOM_CHECKER`, `DF.CUSTOM_CHECKER`, or `DC.CUSTOM_CHECKER`, you might get false positives or false negatives. Test your checker carefully.

    For information, see *Learning to write CodeXM Checkers* ⤴ .

- Inappropriate categorization of issues reported by the analysis, for example, marking a critical issue as *Intentional* as opposed to a *Bug*

    The user is responsible for the correct categorization of issues.

Inappropriate categorization can take place within Coverity Connect and the Coverity Desktop plugins to IDEs such as Eclipse, Visual Studio, and Intellij. See "Triaging issues" in *Coverity Platform 2020.12 User and Administrator Guide* 🔗 and the *Details view* sections within the Coverity Desktop guides.

- Execution of the tool against code compiler versions that are not supported by the product

  The tools must only be run against compiler versions that are listed in the supporting documentation.

  See sections "Coverity Analysis and Dynamic Analysis" 🔗 and "Coverity Test Advisor SCM and platform support" 🔗 in *Coverity 2020.12 Installation and Deployment Guide* for the lists of compilers that are supported by Coverity and Test Advisor.

3.2. Coverity Connect and Coverity Desktop

Coverity Connect and Coverity Desktop components should be installed and configured according to recommended options in their respective user guides.

- Coverity Platform User and Administrator Guide 🔗 (PDF 🔗)

- Coverity Desktop Analysis: User Guide 🔗 (PDF 🔗)

Chapter 4. Applicable Standards and Scope

Table of Contents

Coverity and Test Advisor shall not be used as the sole means of determining whether a product or system is safe.

The analysis takes place on source code only. It cannot detect issues that might arise dynamically as the program runs and then cause a safety hazard. The analysis is not guaranteed to find all software defects, nor will it find defects in third-party code for which source code is not made available.

4.1. Tool Error Detection

As in the case of all static code analysis tools, Coverity might report *False Positives*, which are issues that are not actual errors in the context of the relevant code. In addition, the tool might be subject to *False Negatives*, which are undiscovered, and therefore unreported, issues that are present in the code.

- The degree of confidence that a False Positive can be identified by the user is high (TD1).

- The degree of confidence that a False Negative can be identified by the user is low (TD3).

Examples of False Negatives are discussed in various sections of the *Coverity 2020.12 Checker Reference*, such as in "Modeling Sources of Untrusted (Tainted) Data" and "Modeling Methods to which Tainted Data Must Not Flow (Sinks)".

4.2. Standards Compliance

IEC 61508
    T2: Coverity and Test Advisor tools can be classified as T2.

ISO 26262
    The Tool Impact (TI) and Tool Error Detection (TD) shall be defined by each project using the tool.

ASIL
    ASIL D: The tools have been qualified to be used in safety-relevant development up to ASIL D.

Chapter 5. Documentation and Support

A comprehensive set of documentation is provided as part of the Coverity and Test Advisor installation package.

> **The Coverity 2020.12 Installation and Deployment Guide** should be consulted before the software is installed or used to analyze safety-critical software.

The full documentation set covers both the use of the analysis client and issue management software that will help users fully understand their use.

Synopsys provides Customer Support at www.synopsys.com as part of the products, and users of Coverity and Test Advisor are provided with information about defects within the products upon request.

Appendix A. Legal Notice

Table of Contents

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 acknowlegement: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)."

   Alternately, this acknowlegement may appear in the software itself, if and wherever such third-party acknowlegements 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 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. SYNOPSYS AND ITS SUPPLIERS SPECIFICALLY DISCLAIM ALL IMPLIED WARRANTIES, CONDITIONS AND REPRESENTATIONS ARISING OUT OF COURSE OF DEALING, USAGE OR TRADE.