



Coverity 2020.12 Release Notes Archive

For Coverity Analysis, Coverity Platform, and Coverity Desktop.

Copyright 2020 Synopsys, Inc. All rights reserved worldwide.

Table of Contents

1. Coverity 2020.09-5 Release Notes	1
1.1. Important information for 2020.09-5	1
1.2. Coverity Analysis 2020.09-5	1
2. Coverity 2020.09-4 Release Notes	2
2.1. Important information for 2020.09-4	2
3. Coverity 2020.09-3 Release Notes	3
3.1. Important information for 2020.09-3	3
3.2. Coverity Platform 2020.09-3	3
4. Coverity 2020.09-2 Release Notes	4
4.1. Important information for 2020.09-2	4
4.2. Coverity Analysis 2020.09-2	4
5. Coverity 2020.09-1 Release Notes	5
5.1. Important information for 2020.09-1	5
5.2. Coverity Platform 2020.09-1	5
6. Coverity 2020.09-1 Release Notes	6
6.1. Important information for 2020.09-1	6
6.2. Coverity Analysis 2020.09-1	6
7. Coverity 2020.09 Release Notes	7
7.1. Important information for 2020.09	7
7.2. Coverity Platform 2020.09	7
7.3. Coverity Analysis 2020.09	13
7.4. Coverity Desktop 2020.09	28
7.5. Coverity Documentation 2020.09	29
8. Coverity 2020.06-4 Release Notes	31
8.1. Important information for 2020.06-4	31
8.2. Coverity Platform 2020.06-4	31
9. Coverity 2020.06-3 Release Notes	32
9.1. Important information for 2020.06-3	32
9.2. Coverity Platform 2020.06-3	32
10. Coverity 2020.06-3 Release Notes	33
10.1. Important information for 2020.06-3	33
10.2. Coverity Analysis 2020.06-3	33
11. Coverity 2020.06-2 Release Notes	34
11.1. Important information for 2020.06-2	34
11.2. Coverity Platform 2020.06-2	34
12. Coverity 2020.06-2 Release Notes	35
12.1. Important information for 2020.06-2	35
12.2. Coverity Analysis 2020.06-2	35
13. Coverity 2020.06-1 Release Notes	36
13.1. Important information for 2020.06-1	36
13.2. Coverity Analysis 2020.06-1	36
14. Coverity 2020.06 Release Notes	37
14.1. Important information for 2020.06	37
14.2. Coverity Platform 2020.06	37
14.3. Coverity Analysis 2020.06	42
14.4. Coverity Desktop 2020.06	60

Coverity 2020.12 Release Notes Archive

14.5. Coverity Documentation 2020.06	62
15. Coverity 2020.03-8 Release Notes	64
15.1. Important information for 2020.03-8	64
15.2. Coverity Analysis 2020.03-8	64
16. Coverity 2020.03-7 Release Notes	65
16.1. Important information for 2020.03-7	65
16.2. Coverity Platform 2020.03-7	65
17. Coverity 2020.03-6 Release Notes	66
17.1. Important information for 2020.03-6	66
17.2. Coverity Analysis 2020.03-6	66
18. Coverity 2020.03-5 Release Notes	67
18.1. Important information for 2020.03-5	67
18.2. Coverity Platform 2020.03-5	67
19. Coverity 2020.03-4 Release Notes	68
19.1. Important information for 2020.03-4	68
19.2. Coverity Platform 2020.03-4	68
20. Coverity 2020.03-4 Release Notes	69
20.1. Important information for 2020.03-4	69
20.2. Coverity Analysis 2020.03-4	69
21. Coverity 2020.03-3 Release Notes	70
21.1. Important information for 2020.03-3	70
21.2. Coverity Analysis 2020.03-3	70
22. Coverity 2020.03-2 Release Notes	71
22.1. Important information for 2020.03-2	71
22.2. Coverity Platform 2020.03-2	71
22.3. Coverity Analysis 2020.03-2	72
23. Coverity 2020.03-1 Release Notes	73
23.1. Important information for 2020.03-1	73
23.2. Coverity Analysis 2020.03-1	73
24. Coverity 2020.03 Release Notes	74
24.1. Important information for 2020.03	74
24.2. Coverity Platform 2020.03	74
24.3. Coverity Analysis 2020.03	80
24.4. Coverity Desktop 2020.03	94
24.5. Coverity Documentation 2020.03	96
25. Coverity 2019.12-8 Release Notes	97
25.1. Coverity Analysis Coverity Commands	97
26. Coverity 2019.12-7 Release Notes	98
26.1. Coverity Platform bug fixes	98
27. Coverity 2019.12-6 Release Notes	99
27.1. Coverity Platform bug fixes	99
28. Coverity 2019.12-5 Release Notes	100
28.1. Coverity Platform bug fixes	100
29. Coverity 2019.12-4 Release Notes	101
29.1. Coverity Analysis bug fixes	101
30. Coverity 2019.12-3 Release Notes	102
30.1. Coverity Analysis bug fixes	102
30.2. Coverity Platform bug fixes	102

Coverity 2020.12 Release Notes Archive

31. Coverity 2019.12-2 Release Notes	103
31.1. Coverity Analysis bug fixes	103
32. Coverity 2019.12-1 Release Notes	104
32.1. Coverity Analysis bug fixes	104
33. Coverity 2019.12 Release Notes	105
33.1. Important information for 2019.12	105
33.2. Coverity Platform 2019.12	106
33.3. Coverity Analysis 2019.12	110
33.4. Coverity Desktop 2019.12	132
33.5. Coverity Report Generators 2019.12	136
33.6. Coverity Documentation 2019.12	137
34. Coverity 2019.09-6 Release Notes	138
34.1. Coverity Platform	138
35. Coverity 2019.09-5 Release Notes	139
35.1. Coverity Analysis	139
36. Coverity 2019.09-4 Release Notes	140
36.1. Coverity Analysis	140
36.2. Compiler configuration, Build capture, and Compiler Integration Toolkit (CIT) bug fixes ..	140
36.3. Coverity Platform	140
37. Coverity 2019.09-3 Release Notes	141
37.1. Coverity Platform bug fixes	141
38. Coverity 2019.09-2 Release Notes	142
38.1. Coverity Analysis bug fixes	142
38.2. Coverity Documentation Release Notes	142
39. Coverity 2019.09-1 Release Notes	143
39.1. Coverity Analysis bug fixes	143
40. Coverity 2019.09 Release Notes	144
40.1. Important information for 2019.09	144
40.2. Coverity Platform 2019.09	146
40.3. Coverity Documentation 2019.09	150
41. Coverity 2019.06-11 Release Notes	151
41.1. Coverity Platform	151
42. Coverity 2019.06-10 Release Notes	152
42.1. Coverity Analysis	152
43. Coverity 2019.06-9 Release Notes	153
43.1. Coverity Analysis	153
44. Coverity 2019.06-7 Release Notes	154
44.1. Coverity Platform bug fixes	154
45. Coverity 2019.06-6 Release Notes	155
45.1. Coverity Analysis bug fixes	155
46. Coverity 2019.06-5 Release Notes	156
46.1. Coverity Analysis bug fixes	156
47. Coverity 2019.06-1 Release Notes	157
47.1. Coverity Analysis bug fixes	157
48. Coverity 2019.06 Release Notes	158
48.1. Important information for 2019.06	158
48.2. Coverity Platform 2019.06	159
48.3. Coverity Analysis 2019.06	163

Coverity 2020.12 Release Notes Archive

48.4. Coverity Desktop 2019.06	181
48.5. Coverity Report Generators 2019.06	184
48.6. Coverity Documentation 2019.06	185
49. Coverity 2019.03-12 Release Notes	186
49.1. Coverity Platform bug fixes	186
50. Coverity 2019.03-11 Release Notes	187
50.1. Coverity Analysis bug fixes	187
51. Coverity 2019.03-10 Release Notes	188
51.1. Coverity Analysis bug fixes	188
52. Coverity 2019.03-9 Release Notes	189
52.1. Coverity Platform bug fixes	189
53. Coverity 2019.03-7 Release Notes	190
53.1. Coverity Analysis bug fixes	190
54. Coverity 2019.03-6 Release Notes	191
54.1. Coverity Analysis bug fixes	191
55. Coverity 2019.03-5 Release Notes	192
55.1. Coverity Analysis bug fixes	192
56. Coverity 2019.03-4 Release Notes	193
56.1. Coverity Platform bug fixes	193
57. Coverity 2019.03-3 Release Notes	194
57.1. Coverity Analysis bug fixes	194
58. Coverity 2019.03-2 Release Notes	195
58.1. Coverity Analysis bug fixes	195
58.2. Report Generators bug fixes	195
59. Coverity 2019.03-1 Release Notes	196
59.1. Coverity Analysis bug fixes	196
60. Coverity 2019.03 Release Notes	197
60.1. Important information for 2019.03	197
60.2. Coverity Platform 2019.03	198
60.3. Coverity Analysis 2019.03	202
60.4. Coverity Desktop 2019.03	221
60.5. Coverity Report Generators 2019.03	224
60.6. Coverity Documentation 2019.03	225
61. Coverity 2018.12-12 Release Notes	227
61.1. Coverity Analysis bug fixes	227
62. Coverity 2018.12-11 Release Notes	228
62.1. Coverity Platform bug fixes	228
63. Coverity 2018.12-10 Release Notes	229
63.1. Coverity Analysis bug fixes	229
64. Coverity 2018.12-9 Release Notes	230
64.1. Coverity Analysis bug fixes	230
65. Coverity 2018.12-8 Release Notes	231
65.1. Coverity Analysis bug fixes	231
66. Coverity 2018.12-7 Release Notes	232
66.1. Coverity Analysis bug fixes	232
67. Coverity 2018.12-6 Release Notes	233
67.1. Coverity Compiler Integration Toolkit (CIT) bug fixes	233
68. Coverity 2018.12-5 Release Notes	234

Coverity 2020.12 Release Notes Archive

68.1. Coverity Connect bug fixes	234
69. Coverity 2018.12-4 Release Notes	235
69.1. Coverity Compiler Integration Toolkit (CIT) bug fixes	235
70. Coverity 2018.12-3 Release Notes	236
70.1. Coverity Connect bug fixes	236
71. Coverity 2018.12-2 Release Notes	237
71.1. Coverity Analysis bug fixes	237
72. Coverity 2018.12-1 Release Notes	238
72.1. Coverity Analysis bug fixes	238
73. Coverity 2018.12 Release Notes	239
73.1. Important information for 2018.12	239
73.2. Coverity Platform 2018.12	240
73.3. Coverity Analysis 2018.12	244
73.4. Coverity Desktop 2018.12	269
73.5. Coverity Documentation 2018.12	273
74. Coverity 2018.09-15 Release Notes	274
74.1. Coverity Analysis bug fixes	274
75. Coverity 2018.09-14 Release Notes	275
75.1. Coverity Compiler Integration Toolkit (CIT) bug fixes	275
76. Coverity 2018.09-13 Release Notes	276
76.1. Coverity Analysis bug fixes	276
77. Coverity 2018.09-12 Release Notes	277
77.1. Coverity Analysis bug fixes	277
78. Coverity 2018.09-11 Release Notes	278
78.1. Coverity Compiler Integration Toolkit (CIT) bug fixes	278
79. Coverity 2018.09-9 Release Notes	279
79.1. Coverity Platform bug fixes	279
80. Coverity 2018.09-7 Release Notes	280
80.1. Coverity Analysis bug fixes	280
81. Coverity 2018.09-6 Release Notes	281
81.1. Coverity Connect bug fixes	281
82. Coverity 2018.09-5 Release Notes	282
82.1. Coverity Analysis bug fixes	282
83. Coverity 2018.09-4 Release Notes	283
83.1. Coverity Analysis bug fixes	283
84. Coverity 2018.09-3 Release Notes	284
84.1. Coverity Platform bug fixes	284
85. Coverity 2018.09-2 Release Notes	285
85.1. Coverity Analysis bug fixes	285
86. Coverity 2018.09-SP1 Release Notes	286
86.1. Coverity Analysis bug fixes	286
87. Coverity 2018.09 Release Notes	287
87.1. Important information for 2018.09	287
87.2. Coverity Platform 2018.09	289
87.3. Coverity Analysis 2018.09	293
87.4. Coverity Desktop 2018.09	302
87.5. Coverity Documentation 2018.09	304
A. Legal Notice	306

A.1. Legal Notice 306

Chapter 1. Coverity 2020.09-5 Release Notes

Table of Contents

1.1. Important information for 2020.09-5	1
1.2. Coverity Analysis 2020.09-5	1

1.1. Important information for 2020.09-5

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 Analysis 2020.09-5

This section provides release notes for Coverity Analysis components.

1.2.1. Coverity Compilers and Capture 2020.09-5

1.2.1.1. Bug fixes

CMPCPP-10951

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

Chapter 2. Coverity 2020.09-4 Release Notes

Table of Contents

2.1. Important information for 2020.09-4 2

2.1. Important information for 2020.09-4

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.

Chapter 3. Coverity 2020.09-3 Release Notes

Table of Contents

3.1. Important information for 2020.09-3	3
3.2. Coverity Platform 2020.09-3	3

3.1. Important information for 2020.09-3

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.

3.2. Coverity Platform 2020.09-3

This section provides release notes for Coverity Platform components.

3.2.1. Coverity Connect 2020.09-3

3.2.1.1. Bug fixes

IM-25401

Fixed incorrect 'occurrenceCount' value in WS API method 'getMergedDefectsForProjectScope'

Chapter 4. Coverity 2020.09-2 Release Notes

Table of Contents

4.1. Important information for 2020.09-2	4
4.2. Coverity Analysis 2020.09-2	4

4.1. Important information for 2020.09-2

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.

4.2. Coverity Analysis 2020.09-2

This section provides release notes for Coverity Analysis components.

4.2.1. Coverity Compilers and Capture 2020.09-2

4.2.1.1. Bug fixes

CMPCPP-10804

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

Chapter 5. Coverity 2020.09-1 Release Notes

Table of Contents

5.1. Important information for 2020.09-1	5
5.2. Coverity Platform 2020.09-1	5

5.1. Important information for 2020.09-1

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.

5.2. Coverity Platform 2020.09-1

This section provides release notes for Coverity Platform components.

5.2.1. Coverity Connect 2020.09-1

5.2.1.1. Bug fixes

IM-24483

Mitigated a bug that was sometimes causing in-memory cache of assigned roles to become inconsistent with Coverity Connect DB until Coverity Connect is restarted.

IM-25373

An issue has been resolved that resulted in a difference in defect count between CIM and `summary.txt`

Chapter 6. Coverity 2020.09-1 Release Notes

Table of Contents

6.1. Important information for 2020.09-1	6
6.2. Coverity Analysis 2020.09-1	6

6.1. Important information for 2020.09-1

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.

6.2. Coverity Analysis 2020.09-1

This section provides release notes for Coverity Analysis components.

6.2.1. Coverity Checkers 2020.09-1

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*.

6.2.1.1. Bug fixes

SAT-36241

Both built-in and user-written C++ CodeXM checkers were not being applied to CUDA source files. This has been fixed.

6.2.2. Coverity Compilers and Capture 2020.09-1

6.2.2.1. Bug fixes

CMPSWIFT-460

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

Chapter 7. Coverity 2020.09 Release Notes

Table of Contents

7.1. Important information for 2020.09	7
7.2. Coverity Platform 2020.09	7
7.3. Coverity Analysis 2020.09	13
7.4. Coverity Desktop 2020.09	28
7.5. Coverity Documentation 2020.09	29

7.1. Important information for 2020.09

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.

7.2. Coverity Platform 2020.09

This section provides release notes for Coverity Platform components.

7.2.1. Coverity Compliance Solution 2020.09

Coverity Compliance Solution helps quality managers and architects manage coding standards projects, those using MISRA, CERT, or AUTOSAR standards, which typically surface large numbers of findings. Using Compliance Solution, developers can focus on the most important issues and even prioritize these.

If you use coding standards and find you have a larger number of defects than you can comfortably handle, the Compliance Solution will let you do the following:

- Visualize large numbers of findings and make decisions about how to handle them
- Use those decisions to filter findings, excluding all that are not of interest now
- Upload only the interesting findings to Coverity Connect

Compliance Solution is now in a beta phase. For documentation, tutorials, and information about the beta program, please see the solution's Community page: <https://community.synopsys.com/s/coverity-compliance-solution>

7.2.1.1. Known issues and solutions

COMP-351

The threshold control on the Filter Policies/Threshold page does not work on some versions of Microsoft Edge browser. Workaround: Use a different browser.

COMP-386

The installer that the bootstrap script runs quits if you press Enter too many times during the display of the End User License Agreement. You can work around this by pressing 'q' once while the EULA is displayed.

COMP-420

When you run `cov-upload-findings`, please ignore the warning message that says `no EndPointIdentificationAlgorithm has been configured for SslContextFactory`.

COMP-507

The installer executed by the bootstrap script fails to read the EULA agreement and quits if you select `y` to read the EULA. You can work around this by selecting `n` to read the EULA.

7.2.2. Coverity Connect 2020.09

7.2.2.1. New or changed features

- Line number and triage comments information are now displayed for an exported CSV file. (IM-21055)
- Preview issues are now included in Coverity Connect URL construction query output. (IM-24420)
- Made the following additions to the Defect service of the Coverity Web Services API: Added these complex objects: `standardAttributeIdDataObj`, `standardAttributeValueFilterMapDataObj`, `standardAttributeValueIdDataObj`. Added a `standardAttributeValueFilterMapList` field to the `filterSpec` parameter of the `getMergedDefectsForProjectScope`, `getMergedDefectsForSnapshotScope`, and `getMergedDefectsForStreams` operations. (IM-25015)
- Made the following additions to the Configuration service of the Coverity Web Services API: Added these operations: `getStandardAttribute`, `getStandardAttributes`. Added these complex objects: `standardAttributeDataObj`, `standardAttributeIdDataObj`, `standardAttributeValueDataObj`, `standardAttributeValueIdDataObj`. (IM-25127)

7.2.2.2. Bug fixes

COVDOCS-95

Coverity Connect and Coverity Policy Manager support for Linux was inadvertently omitted from the Coverity Installation and Deployment Guide in releases 2019.06 and 2019.09. Support information for Linux was reinstated in the documentation in release 2019.12. Support for Linux Kernel v2.6.32 or higher, glibc 2.12 or higher, and GTK2+ or higher has in fact been continuous through this period and continues in the current release.

COVDOCS-98

CIM Web Services example code has been fixed.

IM-20518

Documentation has been updated to clarify the column names in Status Reports view.

IM-23626

A description for the `getOutputFileForSnapshot` operation has been added to the "Coverity Platform Web Services API Reference".

IM-24410

A bug was fixed for the Checker View on the Dashboard - all the checker names in the Bar type chart appeared to be truncated.

IM-24562

The `getDeveloperStreamsProjects` operation was added to the Configuration service in the Coverity Platform Web Services API v8 as part of Coverity release 7.0.1.1 but was inadvertently omitted from the documentation. The Coverity Platform Web Services API Reference has been updated with documentation for this method.

IM-24734

A solution was provided for a situation in which Connect was unable to establish the certificate chain of trust to the mail server.

IM-24871

Improved the performance of the following methods of the Defect Service Web Service:
`getMergedDefectsForProjectScope`, `getMergedDefectsForStreams`,
`getMergedDefectsForSnapshotScope`.

This resulted in improving the performance of the `cov-manage-im` command. In some internal tests the latencies were reduced from about 120 seconds to about 8 seconds and from about 13 hours to about 15 minutes.

Increased the maximum allowed value of the `pageSize` field of the `pageSpec` parameter of the aforementioned methods. As a result, the maximum allowed value of the `--page` parameter of the `cov-manage-im` command also increased. While this change contributes to the aforementioned performance improvements, it is not the major one, but simply one visible to a user.

IM-24977

An issue was fixed with the Coverity Server not Responding to `cov-commit-defects`.

IM-25006

The server can now handle a heavier load of incoming commit-related connections.

IM-25056

Fixed an issue where filtering or grouping by standard attribute returns an error if the column is not visible.

IM-25092

An issue has been fixed which now shows Standard Attribute information that was missing from Issues by Snapshot Views as well as from Policy Manager Reports.

IM-25132

An issue was fixed that prevented users from changing their password from the Preferences dialog.

IM-25164

Updated documentation to show that the component views does not include data from outdated streams.

IM-25273

Port redirection is not supported with the default self-signed certificate `server/base/conf/server.xml`.

INS-2959

Fixed an issue that required the user to type password twice to `cov-commit-defects` command.

7.2.2.3. 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
```

7.2.3. Coverity Report Generators 2020.09

7.2.3.1. New or changed features

- The Scan Date / Time has been added in all PDF reports - Security, Software Integrity, Cert C/C++, OWASP, PCI DSS, and CVSS Report. (RG-1421)

7.2.3.2. Bug fixes

IM-25092

An issue has been fixed which now shows Standard Attribute information that was missing from Issues by Snapshot Views as well as from Policy Manager Reports.

RG-1408

A reference to Coverity as a "Synopsys company" has been fixed in Coverity Security Reports.

RG-1418

It is now possible to view the Coverity Severity and Severity level from Security report together.

RG-1422

Documentation was updated to note that value passed on the command line for project name will override the config file setting.

RG-1423

An issue was fixed wherein the project description field was cropped in Integrity Report.

7.2.3.3. 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 Poretcode SC.

7.3. Coverity Analysis 2020.09

This section provides release notes for Coverity Analysis components.

7.3.1. Coverity Architecture Analysis 2020.09

7.3.1.1. Deprecated products and features

COVDOCS-125

Support for all operating systems is deprecated as of 2020.06 and will be removed in a future release.

7.3.1.2. Bug fixes

CMPJ-997

The Java compiler now supports the `--system=none` option value to suppress the inclusion of system modules for Java 9+ builds.

SAT-33665

Fixed broken links in documentation.

SAT-34724

Fixed broken links in documentation.

7.3.2. Coverity Checkers 2020.09

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*.

7.3.2.1. New or changed features

- The `HEADER_INJECTION` checker now supports VB.NET (SAT-26272)
- The new `C/C++ Y2K38_SAFETY` checker points out two potential issues with the rollover of the 32-bit signed integer counter of seconds since epoch in the UNIX `time_t` type. (SAT-27539)
- Enhanced `NO_EFFECT` checker to detect useless `continue` statements, that is, any `continue` statements that are the last statement executed in a loop. (SAT-31515)
- The new `CUDA.SHARE_FUNCTION` checker searches for violations of specific function calls to the device-only function in a host execution space, and vice-versa. (SAT-31595)
- The new `CUDA.CUDA.SPECIFIERS_INCONSISTENCY` checker looks for inconsistencies in CUDA execution space and kernel function specifiers across function declarations. (SAT-31603)
- The `--field-offset-escape` option no longer affects checkers such as `UNINIT_CTOR` that care about writes but not frees. (SAT-32483)

- Improved `DEADCODE` reporting when the dead code is due to a condition on a variable, and that variable is always assigned a constant when reaching that condition. (SAT-32861)
- Added support for MISRA-C: 2012 Amendment 2. (SAT-33918)
- The `SCRIPT_CODE_INJECTION` checker now supports android taints for Java. (SAT-34011)
- The `UNSAFE_JNI` checker now supports Android taints for Java. (SAT-34060)
- The `ANDROID_CAPABILITY_LEAK` checker now supports configurable Android API levels (SAT-34663)
- The `TAINTED_SCALAR` checker can now treat assembly swap instructions as source of tainted data. (SAT-34820)
- For the `FORWARD_NULL` checker, updated description of the boolean option `aggressive_null_sources`. (SAT-34889)
- Added support for these CERT C POSIX rules: POS30-C (v.79), POS33-C (v.101), POS34-C (v.126), POS35-C (v.86), POS36-C (v.67), POS37-C (v.79), POS38-C (v.35), POS39-C (v.51), POS44-C (v.23), POS47-C (v.58), POS49-C (v.24), POS50-C (v.17), POS52-C (v.23), POS54-C (v.32) (SAT-34936)
- Added Go language support to all options for the `OPEN_REDIRECT` checker. (SAT-34977)
- Updated the `INSECURE_COMMUNICATION` checker for Java to support Spring Roo configuration files. (SAT-34987)
- The `HEADER_INJECTION` checker now supports Visual Basic. (SAT-34996)
- Added modeling for the Boost `property_map` API. (SAT-35017)
- The new Java `CONFIG.SPRING_SECURITY_WEAK_PASSWORD_HASH` checker finds cases that create an instance of a class implementing the `PasswordEncoder` interface using weak hashing algorithms or no hashing algorithm at all. (SAT-35033)
- Added a new option to `cov-analyze` command. The `--resolve-calls-to-all-delegates` option might allow reporting more defects involving calls to delegates, notably `LOCK_INVERSION` defects. It might cause a higher false positive rate. (SAT-35036)
- The new Java `CONFIG.SPRING_SECURITY_DEPRECATED_XSS_HEADER` checker finds cases where the soon to be deprecated X-XSS-Protection header is explicitly enabled. (SAT-35062)
- The `XML_INJECTION` checker now supports VB.NET. (SAT-35132)
- The new Java `VERBOSE_ERROR_REPORTING` checker finds cases where an application has been configured to allow exception information or stack traces to be displayed in an error page. (SAT-35134)
- The new C/C++ `Y2K38_SAFETY` checker points out two potential issues with the rollover of the 32-bit signed integer counter of seconds since epoch in the UNIX `time_t` type. (SAT-35167)

- Added two options to the `ANDROID_CAPABILITY_LEAK` checker:
`ANDROID_CAPABILITY_LEAK:default_targetSdk:<integer>`; (sets which Android API level the application targets), and
`ANDROID_CAPABILITY_LEAK:detect_targetSdk:<boolean>`; (sets whether the analysis will auto-detect the Android API level that the application targets). (SAT-35196)
- Added support of SEI CERT C coding standard for Clang-based compilers. (SAT-35313)
- The new `CUDA.SHARE_OBJECT_STREAM_ASSOCIATED` checker finds instances when managed global variables associated with a stream are accessed from a different stream. (SAT-35390)
- Added security models for the `libpq` library. (SAT-35399)
- Added support for 3 more rules in AUTOSAR C++14 standard on both EDG-based and Clang-based compilers: A1-1-1, A14-5-2, A15-0-7 (SAT-35468)
- Updated the Java `CONFIG.SPRING_SECURITY_SESSION_FIXATION` checker to support cases where the session fixation protection is explicitly disabled in the source code. (SAT-35498)
- The new Java `CONFIG.SPRING_BOOT_SSL_DISABLED` checker finds cases when SSL is disabled in configuration files of Spring Boot applications. (SAT-35505)
- The `INSECURE_COOKIE` checker now supports C# applications. (SAT-35528)
- Added a new option to the `NO_EFFECT'` checker. The `NO_EFFECT:report_useless_continue:false` option is supported for C, C++, Objective-C, Objective-C++; it reports the `continue` statements that can be removed without affecting code execution. (SAT-35541)
- The new Java `CONFIG.SPRING_SECURITY_CSRF_PROTECTION_DISABLED` checker finds cases where the Spring Security cross-site request forgery (CSRF) protection is explicitly disabled. (SAT-35542)
- The new `CUDA.INVALID_MEMORY_ACCESS` checker finds cases where pointers into host or device memory are used incorrectly. (SAT-35560)
- The new Java `CONFIG.JAVAAEE_MISSING_SERVLET_MAPPING` checker finds cases where a deployment descriptor XML configuration file contains a servlet entry without a corresponding servlet mapping, enabling dangerous implicit mapping. (SAT-35562)
- The `INSECURE_COMMUNICATION` checker now supports configuration files for Java projects. (SAT-35605)
- Added support for 2 new CERT-JAVA rules: CERT IDS16-J and CERT IDS17-J. (SAT-35687)
- Brakeman version has been upgraded to 4.8.2. (SAT-35697)

7.3.2.2. Bug fixes

COVDOCS-100

Reference to the renamed checker `__CONFIG.SPRING_SECURITY_DEBUG_MODE_JAVA` has been removed.

SAT-25046

Fixed `TAINTED_SCALAR` checker to report different defects on different fields of the same variable on a given path.

SAT-28360

The `CSRF` checker now detects Spring CSRF protection enablement using the Spring version and handles the `<csrf disabled="true">` tag correctly.

SAT-31900

A false negative for the `FLOATING_POINT_EQUALITY` checker (vector types containing floats) has been fixed.

SAT-32024

Add resource leak primitive for `URLConnection::getInputStream` and `URLConnection::getOutputStream`.

SAT-32466

A false positive was fixed for the `NO_EFFECT` checker.

SAT-32624

Added model for `g_slice_free_chain_with_offset` to fix false positive for the `ALLOC_FREE_MISMATCH` checker.

SAT-32797

A false positive was fixed for the `UNINIT_CTOR` checker.

SAT-32886

Fixed a false positive in `BAD_OVERRIDE` when the overriding function differed only in `const/volatile` qualifiers on the outermost level of the parameter type specification.

SAT-32914

The option `suppress_under_related_conditional` to the `NULL_RETURNS` checker was not actually effective for C, C++, Objective-C, and Objective-C++. The behavior has now been implemented, and is enabled by default.

SAT-32953

An issue was fixed for a `NO_EFFECT` false positive when comparing an unsigned integer with 0 within an impossible condition.

SAT-33449

An issue was fixed: The `UNCAUGHT_EXCEPT` checker now flags situations in which a `bad_alloc` exception is thrown when the string function fails to allocate storage.

SAT-33691

Added string equality models to fix false positives for the `TAINTED_SCALAR` checker on Visual Studio.

SAT-33823

An issue was fixed that produced a false negative for the `SQLI` checker for PHP source.

SAT-33868

Fixed an issue in the `OVERLAPPING_COPY` checker that could cause a recoverable failure if a negative number was supplied for the `size` argument.

SAT-34147

Fixed a source of `UNINIT` false positives when multiple members of a struct are initialized together using a function such as `memset` given the address of the first of those members.

SAT-34832

Fixed performance regression in C/C++ security checkers by removing unnecessary taint tracking.

SAT-34864

CSRF False Positives have been fixed in those cases where users have implemented a homemade `ActionFilter`.

SAT-34869

Fixed a false positive for the `CUDA.INACTIVE_THREAD_AT_COLLECTIVE_WARP` checker.

SAT-34914

Fixed a source of `OVERRUN` false positives when a pointer argument is cast after having an offset added in a callee.

SAT-35014

Fixed a recoverable analysis crash with message "While generating WUP for per-TU reports (...) assertion failed: Invalid index" when analyzing an intermediate directory with no `define` functions using HIS metrics.

SAT-35266

Fixed a recoverable analysis crash with message "Disjoining != NULL"

SAT-35281

A false positive for the `CONFIG.ATS_INSECURE` checker has been fixed.

SAT-35353

A false positive for the `UNINIT_CTOR` checker has been fixed.

SAT-35445

An issue has been fixed with Brakeman Pro when `HOME` is a relative path.

SAT-35489

A false negative was fixed for Javascript DOM XSS.

SAT-35509

CodeXM documentation has been updated with information about regular expressions, as well as descriptions of the `allFunctionCode` and `allFunctionsAndGlobalVariableCode` patterns. In the Learning CodeXM document, the use of patterns to match loops has been clarified.

SAT-35644

Fixed some inconsistent analysis results from Kotlin security analysis with mutually recursive functions. Analysis results will be more consistent across analysis runs. However, Kotlin analysis results might gain or lose a small number of defects compared to the previous release.

SATW-3074

The event message for CERT SIG30-C is now correctly translated in Japanese.

SATW-3763

Fixed a false positive of MISRA C++-2008 Rule 6-6-5 regarding `statementExpression`.

SATW-3766

Fixed a false positive of AUTOSAR C++14 M3-2-3 about template function declarations.

SATW-3773

Fixed a false positive of CERT EXP37-C where `__set_psw(unsigned char)` is called.

SATW-3780

Fixed a false positive of MISRA C-2012 Rule 13.2 about two function calls with unrelated side effects.

SATW-3788

Fixed false positives of AUTOSAR C++14 A4-7-1 related to casting already checked variables and `constexpr` variables.

SATW-3790

Fixed a false positive of AUTOSAR C++14 A12-1-5 when there were no other constructors to delegate to.

SATW-3798

Fixed a false positive of AUTOSAR C++14 A5-2-2 where a function was cast to be used as a template parameter.

SATW-3803

Fixed a false positive of CERT INT31-C about `sizeof` operator.

SATW-3810

Fixed a false positive of AUTOSAR C++14 A3-9-1 where the type was dependent on template arguments.

SATW-3821

Fixed a false positive of AUTOSAR C++14 M6-4-2 where a `throw` operator was wrapped in an `else` statement.

SATW-3847

Fixed a false negative of CERT CON39-C for C++ source files.

7.3.2.3. 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.

7.3.3. Coverity Commands 2020.09

7.3.3.1. New or changed features

- Added a new option to the `cov-manage-emit` command. The `--tu-sort` option specifies the sort order for TU output. (CMPG-3355)
- Added a new option to the `cov-configure` command. The `--coverity-response-file=<response_file>` specifies a response file that contains a list of additional command line arguments, such as a list of input files. (CMPG-3357)
- Added `cov-archive` support of importing to coordinator. See the description of the `--cluster-config` option in the `cov-archive` documentation. (IM-25048)
- Added the new `--brakeman-aggressiveness-level` option to `cov-analyze`. This option allows users to tune the aggressiveness of Brakeman Pro to only report defects above a certain confidence level. (SAT-34008)
- For the `cov-manage-emit` command, under the "Translation unit pattern matching" section, added `all` to regular expression values. (SAT-34919)
- The `cov-run-fortran` command now uses response files to communicate with the underlying analysis. This removes an earlier limitation due to the maximum command-line size (approx. 2¹⁵ bytes on Windows; 2¹⁷ bytes on other platforms). (SAT-35004)
- For the `cov-manage-emit` command, the following language patterns were added to the "Translation unit pattern matching" section: .NET bytecode, Fortran, Go, HTML, JSX, JVM bytecode, Kotlin, Python 2, Python 3, TypeScript, Vue.js SFC. (SAT-35187)
- Coverity client tools will now accept all SSL/TLS cipher suites, allowing more flexibility in server configuration, in particular with reverse proxies. (SAT-35801)

7.3.3.2. Bug fixes

CMPCPP-10203

Errors related to in-class initializers and incomplete type errors have been fixed.

CMPCPP-10248

Fixed an issue for `cov-emit`: error #135 has no member "type".

CMPCPP-10258

An issue resulting from seeing assertion error when using `cov-build` has been fixed.

CMPCPP-9135

Fix made for 2020.09 release: `cov-internal-emit-clang` now generates xrefs for variable templates used within namespaces.

CMPCSH-1399

And issue with the `cov-build` command in Visual Studio has been fixed.

IM-22843

Logging was improved for `cov-admin-db` by incorporating verbose mode.

IM-25145

Fixed `cov-manage-im` checker filter option.

INS-2959

Fixed an issue that required the user to type password twice to `cov-commit-defects` command.

SAT-32766

An issue was fixed for a situation in which `cov-run-desktop` would not take the `-use-jshintrc` option.

SAT-34401

Fixed an issue causing a server certificate to be rejected by `cov-commit-defects` with the message "ASN CA path length larger than signer error" if the issuing root CA has a path length limit of 0.

SAT-34712

An issue was fixed where `cov-run-fortran` crashed for large projects. An internal command-line buffer was limited to 1500 characters. This limitation has been removed.

SAT-34833

Fixed a `cov-commit-defects` crash with message "Expected a value to be present for optional integer" when using an HTTP redirect to an HTTPS address with no port specified.

SAT-34881

In the results from `cov-run-fortran`, certain syntax errors (defects) were not being associated with any function. To track defects through line number changes, Coverity Platform requires a function name as part of the defect identifier. Such syntax errors are now attributed to `<module>%.MAIN.` if within a module, and to `.MAIN.` otherwise.

SAT-34911

The `cov-run-fortran` command is now providing friendlier messages for abnormal exits.

SAT-35381

An issue with `cov-format-errors` has been fixed.

SAT-35586

Fixed an issue for `cov-commit-defects` failure due to long file names.

SAT-7240

Fixed an issue that could cause results to change depending on compilation order, when multiple compilations of the same file were not within a single `cov-build` command.

7.3.3.3. 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.

7.3.4. Coverity Compilers and Capture 2020.09

7.3.4.1. End-of-life products

CMPG-3251

Support for OpenJDK 13 is dropped as of 2020.09. Support for Oracle JDK 13 is dropped as of 2020.09.

CMPG-3258

Coverity support for Go 1.11 and 1.12 has reached end of life and is dropped in Coverity 2020.09 release.

CMPG-3260

Support for Apple Clang 6.0 and 6.2 has been dropped as of 2020.09.

CMPG-3269

Coverity Analysis support for Ruby 2.3 and 2.4 is dropped as of 2020.09.

CMPG-3282

Support for IBM XLC versions 8–12 on AIX is dropped as of 2020.09.

CMPG-3283

Support for Linux versions of Intel C++ older than version 17 is dropped as of 2020.09.

CMPG-3284

Support for Keil Arm compiler RVCT 3.1, 4.0 for uVision is dropped as of 2020.09.

CMPG-3376

Support for .Net Core 3.0 has been dropped as of 2020.09

COVP-2281

We no longer support Extend SDK on FreeBSD.

7.3.4.2. Deprecated products and features

CMPG-3252

Support for Swift 5.2 is deprecated as of 2020.09

7.3.4.3. New or changed features

- Added support for clang-cl 9.0 on Windows. (CMPCPP-10059)
- Added support for ARM Clang 6.13.1. (CMPCPP-10064)
- Added support for GNU GCC and G++ 10.1.0 compiler. (CMPCPP-10088)
- Added support for the Texas Instruments ARM version 18.12.5 compiler on Windows. (CMPCPP-10157)
- Added support for Microchip XC8 version 2.20 compiler on Windows and Linux. (CMPCPP-10261)
- Added support for the Qualcomm Kalimba C version 2.06 compiler on Windows. (CMPCPP-10289)
- Added support for the QNX 7 C++ compiler as a host compiler for the CUDA nvcc compiler. Use the `cov-configure --cuda` command to configure support for this compiler combination. (CMPCPP-10344)
- We now support `-arch arm64, -mcpu=cortex-a7` for the clang compiler. (CMPCPP-4449)

- Added support for the .NET Core C# compiler on linux64. (CMPG-3388)
- Added support for Java 14 language features. (CMPJ-1215)
- Added a capability to the JavaScript front end to suppress secondary capture (capture of files imported by previously captured source files) of files that reside under a `node_modules` directory. Contact Coverity support to enable this functionality. (CMPJS-733)
- Added support for Open JDK 14. (COVP-2271)
- Added support for Oracle JDK 14. (COVP-2272)

7.3.4.4. Bug fixes

CAP-1644

Docs have been updated to correct a discrepancy in the way that Maven versions were specified.

CMPCPP-10270

We are now flagging any usage of variadic macros (which contain ... ellipses representing multiple arguments), to be a violation of MISRA C 2004 Rule 1.1. We do this because variadic macros were introduced in the C99 standard, while Rule 1.1 requires adherence to C89. We are making an exception for variadic macros defined in system header files, since the customer might not be able to change any of these.

CMPCPP-10307

Keil ARM(`armcc`) compiler can recognize `--C99(Uppercase C)`, which `cov-emit` can't.

CMPCPP-10366

The `CIT nvcc:msvc` configuration for the CUDA `nvcc` compiler with Microsoft Visual C++ as the host compiler was corrected to ensure that CUDA-predefined macros are detected and emulated.

CMPCPP-10439

Fixed error when parsing non-zero nontype template arguments for Microsoft compilers.

CMPCPP-10521

Fixed an issue where Coverity failed with undefined `__float128` for `gcc`.

CMPCPP-9047

An issue was fixed in which template function declarations resulted in two declaration locations.

CMPCPP-9135

Fix made for 2020.09 release: `cov-internal-emit-clang` now generates xrefs for variable templates used within namespaces.

CMPCPP-9468

Fixed a crash affecting clang compilers when a C++11 `{{static_assert}}` declaration was used as the body of a selection or loop statement.

CMPCPP-9740

Resolved an issue that resulted from an exception signature mismatch in system headers.

CMPCSH-1153

An issue was fixed when a dll is seen in two locations in the same build, which could cause problems merging other idir's that use the same dll.

CMPCSH-1378

An issue was fixed when a local function was being used in a lambda expression in a generic method.

CMPCSH-1399

And issue with the `cov-build` command in Visual Studio has been fixed.

CMPCSH-1402

A `cov-analyze` issue with custom ValueTypes with a user defined conversion has been resolved.

CMPFG-410

Fixed an issue where `cov-emit-java` crashed on long command line switches.

CMPFG-420

Kotlin front end now properly handles compiler plugins in maven.

CMPFG-424

Fixed a crash in Kotlin front end related to `coroutines` and objects.

CMPG-3378

Removed references to the HPUX environment in documentation because this environment is no longer supported.

CMPJ-1176

Upgraded webapp archive compilation to support JSP files that reference precompiled class files with class format versions up to 58.0, which corresponds to Java 14.

CMPJ-1335

A `cov-emit-java` hang that could occur when emitting the exoplayer project has been fixed.

CMPSWIFT-405

An assertion failure for a Coverity Swift compiler has been fixed.

SAT-35355

Fixed a recoverable error when analyzing static Spring MVC request handlers.

SAT-7240

Fixed an issue that could cause results to change depending on compilation order, when multiple compilations of the same file were not within a single `cov-build` command.

7.3.4.5. 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=esnnext`. As a result, Coverity tools cannot currently emit top-level `awaits` which are built using `module=esnnext`.

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` invokations at the OS level.

7.3.5. Coverity Dynamic Analysis 2020.09

7.3.5.1. Deprecated products and features

COVDOCS-126

Support for all operating systems is deprecated as of 2020.06 and will be removed in a future release.

7.3.5.2. Bug fixes

SAT-35461

An issue has been fixed in which the `cov-run-desktop` command now uses the same process to choose the latest emit of multiple compilations as the `cov-analyze` command.

7.3.5.3. 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.

7.3.6. Coverity Test Advisor 2020.09

7.3.6.1. Known issues and solutions

TADE-2033

The use of "`--cs-coverage opencover`" with Test Advisor may 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. This can be worked around by manually registering the OpenCover profiler DLLs and passing "`--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 on this, see the documentation of `cov-build`'s "`--cs-no-register-profiler`" switch in the 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.

7.3.7. Coverity Wizard 2020.09

7.3.7.1. 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>/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.

7.4. Coverity Desktop 2020.09

This section provides release notes for Coverity Desktop components.

7.4.1. Coverity Desktop for Android Studio 2020.09

7.4.1.1. 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.

7.4.2. Coverity Desktop for Eclipse 2020.09

7.4.2.1. New or changed features

- Added support for Eclipse 2020-06 (4.16) (PRD-12189)

7.4.2.2. 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/`.

7.4.3. Coverity Desktop for IntelliJ IDEA 2020.09

7.4.3.1. 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.

7.5. Coverity Documentation 2020.09

This section provides release notes for Coverity Documentation components.

7.5.1. Coverity Documentation 2020.09

7.5.1.1. New or changed features

- The Coverity CodeXM C/C++ Library Reference documents added support for the CUDA platform. (SAT-34702)

7.5.1.2. Bug fixes

COVDOCS-109

Incorrect examples have been fixed in the "Learning CodeXM" document.

COVDOCS-90

Corrected misuse of "analyses" throughout doc set.

IM-24003

The "Coverity Platform Web Services API Reference" has been updated to reflect the fact that the `updateSignInConfiguration` operation no longer accepts an `enableSessionTimeout` parameter and that the `signInSettingsDataObj` complex type no longer contains an `enableSessionTimeout` component.

7.5.1.3. 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*.

Chapter 8. Coverity 2020.06-4 Release Notes

Table of Contents

8.1. Important information for 2020.06-4	31
8.2. Coverity Platform 2020.06-4	31

8.1. Important information for 2020.06-4

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.

8.2. Coverity Platform 2020.06-4

This section provides release notes for Coverity Platform components.

8.2.1. Coverity Connect 2020.06-4

8.2.1.1. Bug fixes

IM-25276

Fixed user discrepancies in the triage panel owner autocomplete

Chapter 9. Coverity 2020.06-3 Release Notes

Table of Contents

9.1. Important information for 2020.06-3	32
9.2. Coverity Platform 2020.06-3	32

9.1. Important information for 2020.06-3

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.

9.2. Coverity Platform 2020.06-3

This section provides release notes for Coverity Platform components.

9.2.1. Coverity Connect 2020.06-3

9.2.1.1. Bug fixes

IM-25006

The server can now handle a heavier load of incoming commit-related connections.

IM-25185

Add access control check when fetching user information based on ID.

Chapter 10. Coverity 2020.06-3 Release Notes

Table of Contents

10.1. Important information for 2020.06-3	33
10.2. Coverity Analysis 2020.06-3	33

10.1. Important information for 2020.06-3

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.

10.2. Coverity Analysis 2020.06-3

This section provides release notes for Coverity Analysis components.

10.2.1. Coverity Checkers 2020.06-3

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*.

10.2.1.1. Bug fixes

SAT-35709

Fixed a source of `OVERRUN` false positives when a pointer argument is cast after having an offset added in a callee.

10.2.2. Coverity Compilers and Capture 2020.06-3

10.2.2.1. Bug fixes

CMPSWIFT-438

Fixed an issue where swift emit rate was lowered by mishandling missing types.

Chapter 11. Coverity 2020.06-2 Release Notes

Table of Contents

11.1. Important information for 2020.06-2	34
11.2. Coverity Platform 2020.06-2	34

11.1. Important information for 2020.06-2

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.

11.2. Coverity Platform 2020.06-2

This section provides release notes for Coverity Platform components.

11.2.1. Coverity Connect 2020.06-2

11.2.1.1. Bug fixes

IM-25181

Fixed persistent XSS issue involving display of user names. Fixed access control issue for web services backup configuration APIs.

Chapter 12. Coverity 2020.06-2 Release Notes

Table of Contents

12.1. Important information for 2020.06-2	35
12.2. Coverity Analysis 2020.06-2	35

12.1. Important information for 2020.06-2

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.

12.2. Coverity Analysis 2020.06-2

This section provides release notes for Coverity Analysis components.

12.2.1. Coverity Compilers and Capture 2020.06-2

12.2.1.1. Bug fixes

CMPSWIFT-435

An assertion failure for a Coverity Swift compiler has been fixed.

Chapter 13. Coverity 2020.06-1 Release Notes

Table of Contents

13.1. Important information for 2020.06-1	36
13.2. Coverity Analysis 2020.06-1	36

13.1. Important information for 2020.06-1

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.

13.2. Coverity Analysis 2020.06-1

This section provides release notes for Coverity Analysis components.

13.2.1. Coverity Compilers and Capture 2020.06-1

13.2.1.1. Bug fixes

CMPCPP-10312

Corrected configuration of the nvcc compiler when used with Microsoft Visual C++ as the host compiler, to ensure that the implicitly included `cuda_runtime.h` header file is correctly found.

CMPCPP-10313

Corrected implicit instantiation of function templates when compiling CUDA code to match the behavior of the nvcc compiler more closely. This avoids parse errors due to unexpected instantiations that don't occur with the nvcc compiler.

CMPFG-423

Kotlin front end now properly handles compiler plugins in maven.

Chapter 14. Coverity 2020.06 Release Notes

Table of Contents

14.1. Important information for 2020.06	37
14.2. Coverity Platform 2020.06	37
14.3. Coverity Analysis 2020.06	42
14.4. Coverity Desktop 2020.06	60
14.5. Coverity Documentation 2020.06	62

14.1. Important information for 2020.06

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.

14.2. Coverity Platform 2020.06

This section provides release notes for Coverity Platform components.

14.2.1. Coverity Compliance Solution 2020.06

Coverity Compliance Solution helps quality managers and architects manage coding standards projects, those using MISRA, CERT, or AUTOSAR standards, which typically surface large numbers of findings. Using Compliance Solution, developers can focus on the most important issues and even prioritize these.

If you use coding standards and find you have a larger number of defects than you can comfortably handle, the Compliance Solution will let you do the following:

- Visualize large numbers of findings and make decisions about how to handle them
- Use those decisions to filter findings, excluding all that are not of interest now
- Upload only the interesting findings to Coverity Connect

Compliance Solution is now in a beta phase. For documentation, tutorials, and information about the beta program, please see the solution's Community page: <https://community.synopsys.com/s/coverity-compliance-solution>

14.2.1.1. Bug fixes

COMP-417

A bug was fixed where if Findings Manager host's `hostname` command returns a name that is not known to DNS, Findings Manager does not know what streams are available from Coverity Connect, even though Coverity Connect is connected to the message bus.

COMP-441

A bug was fixed where files or directories were displayed under incorrect parent node when you drilled down by path in certain cases.

COMP-451

A bug was fixed for cases in which findings did not match upon drilling down in certain cases.

14.2.1.2. Known issues and solutions

COMP-351

The threshold control on the Filter Policies/Threshold page does not work on some versions of Microsoft Edge browser. Workaround: Use a different browser.

COMP-386

The installer that the bootstrap script runs quits if you press Enter too many times during the display of the End User License Agreement. You can work around this by pressing 'q' once while the EULA is displayed.

COMP-420

When you run `cov-upload-findings`, please ignore the warning message that says no `EndPointIdentificationAlgorithm` has been configured for `SslContextFactory`.

COMP-507

The installer executed by the bootstrap script fails to read the EULA agreement and quits if you select `y` to read the EULA. You can work around this by selecting `n` to read the EULA.

COMP-514

If a user chooses to undo changes on the Scoring Policies page and clicks on Cancel, the Included/ Excluded label doesn't get updated. Clicking on Save after that updates the score that was cancelled. You can work around this by not clicking Save after Cancel and refreshing the page.

14.2.2. Coverity Connect 2020.06

14.2.2.1. End-of-life products

IM-24692

Dropped support for Web Services API version 6, 7, and 8.

INS-2887

The Coverity Platform Updater has reached end of life, and has been removed from the Coverity Connect installer.

14.2.2.2. New or changed features

- It is now possible to automatically disable Coverity users and their access tokens when such users are disabled in LDAP. (IM-24188)
- Coverity Connect now shows you which issues are associated with particular standards. The following standards are covered: AUTOSAR C++14, CERT C and CERT C++, DISA-STIG V4R3, DISA-STIG

V4R3 Severity, DISA-STIG V4R10, DISA-STIG V4R10 Severity, ISO TS17961 2016, OWASP Mobile Top Ten 2016, OWASP Web Top Ten 2017, Payment Card Industry Data Security Standard (PCI DSS) 2018 (IM-24846)

- Added filtering and segmentation by coding standards and vulnerability reports into the policy manager reports. (IM-24858)
- Component views now exclude data from streams marked as "outdated". (IM-24905)

14.2.2.3. Bug fixes

IM-21017

Fixed an issue with duplicating projects.

IM-23481

Fixed the issue that the `remoteHost` field of the `WebAccessEvent` in the `usageLog.log` was not being populated.

IM-24007

A bug was fixed to make Last snapshot column for preview defects blank.

IM-24047

An optional caching feature was added to speed up the rendering of component views.

IM-24387

A bug was fixed that resulted in incorrect log messages being shown in `usageLog.log`.

IM-24477

A bug was fixed: When using Chrome, users were unable to add a filter to 'Outstanding issue count' Summary Metric.

IM-24627

A bug was fixed that prevented an authentication key file from being downloaded for IE and Edge.

IM-24712

`getComponentMetricsForProject` now works as expected.

IM-24767

A bug was fixed for slow commit performance with a large `function_instance` table.

IM-24778

The documentation was updated to indicate that the `cov-admin-db check-integrity` command supports only the embedded database. It does not support an external database.

IM-24807

A bug was fixed for problems arising when configuring Jira Cloud BTS and an API key was used for authentication.

IM-24877

Fixed absent column names for the security standards in email notifications

INS-2757

Add `--autostart={true|false}` argument to enable or disable automatically starting Coverity Connect after a fresh install. The default is `--autostart=true`.

14.2.2.4. 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-23994

Internet Explorer 11 breaks on functionalities using file upload.

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 Java1.7.0_xx and older, `OutOfMemory` 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
```

14.2.3. Coverity Report Generators 2020.06

14.2.3.1. New or changed features

- The CIR Report can now use low impacts for calculating defect density based on user input. (RG-1371)

- Increased "issue-cutoff-count" value from 5000 to 10000 for security report. (RG-1392)

14.2.3.2. Bug fixes

IM-24866

A bug was fixed: When exporting Outstanding Issues to CSV from Coverity server, the values of the OWASP Top 10 and PCI DSS columns always remain 0, even when Coverity Connect displays data.

INS-2932

Cov reports now have documentation.

RG-1346

A bug was fixed to remove the limit of 500 streams.

RG-1385

A bug was fixed where by Snapshot-ID-Date Report gets generated for the whole project.

RG-1387

A bug was fixed where project-contact-email was not a valid email.

14.2.3.3. 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 Poretcode SC.

14.3. Coverity Analysis 2020.06

This section provides release notes for Coverity Analysis components.

14.3.1. Coverity Architecture Analysis 2020.06

14.3.1.1. New or changed features

- We now support Clion 2020.01. (PRD-12096)

14.3.2. Coverity Checkers 2020.06

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*.

14.3.2.1. New or changed features

- Improved support for multidimensional arrays in the UNINIT checker. (SAT-1483)
- Improved the UNINIT checker to handle some cases where an uninitialized variable's address is taken but no initialization happens. (SAT-15899)
- Improved the UNINIT checker to handle some cases where an uninitialized variable's address is taken but no initialization happens. (SAT-19020)
- The INSECURE_RANDOM checker now supports Visual Basic. (SAT-26269)
- The SCRIPT_CODE_INJECTION checker now supports Visual Basic. (SAT-26270)
- Added support for tracking unions to the UNINIT checker. (SAT-2631)
- The RISKY_CRYPTO checker now considers TLS1.2 insecure by default. (SAT-30256)
- Added 'include-files' and 'exclude-files' options for `cov-commit-defects` and `cov-blame` commands. (SAT-31428)
- The new `CUDA.DIVERGENCE_AT_COLLECTIVE_OPERATION` checker looks for calls to a collective thread synchronization operation or a collective warp operation (for pre-Compute Capability 7.0 versions) and checks whether they are diverged on thread index. (SAT-31586)
- The new `CUDA.INACTIVE_THREAD_AT_COLLECTIVE_WARP` checker looks for defects caused when a warp synchronization function is called and the mask does not match the set of participating threads in the warp. It also looks for defects when a warp shuffle function is called and the non-mask arguments are inconsistent with the set of participating threads in the warp. (SAT-31589)
- The new `CUDA.COLLECTIVE_WARP_SHUFFLE_WIDTH` checker looks for calls to collective shuffle operations and checks whether they are passed an incorrect width parameter. (SAT-31594)
- Enabled intraprocedural integer tracking in `RISKY_CRYPTO` to resolve key size values stored in variables. (SAT-31606)
- The `SQLI` checker now supports Kotlin. (SAT-31723)
- The `UNSAFE_DESERIALIZATION` checker now supports Kotlin. (SAT-31738)
- The `UNRESTRICTED_ACCESS_TO_FILE` checker now supports Kotlin. (SAT-31759)

- The `URL_MANIPULATION` checker now supports Kotlin. (SAT-31763)
- The `WEAK_PASSWORD_HASH` checker now supports Kotlin. (SAT-31764)
- The `XML_EXTERNAL_ENTITY` checker now supports Kotlin. (SAT-31765)
- The `MOBILE_ID_MISUSE` checker now supports Kotlin. (SAT-31767)
- The `OS_CMD_INJECTION` checker now supports Kotlin. (SAT-31768)
- The `PATH_MANIPULATION` checker now supports Kotlin. (SAT-31769)
- The `RISKY_CRYPTO` checker now supports Kotlin. (SAT-31771)
- The `ANDROID_CAPABILITY_LEAK` checker now supports Kotlin. (SAT-31802)
- The `EXPOSED_PREFERENCES` checker now supports Kotlin. (SAT-31804)
- The `IMPLICIT_INTENT` checker now supports Kotlin. (SAT-31805)
- The `INSECURE_COMMUNICATION` checker now supports Kotlin. (SAT-31806)
- The `MISSING_PERMISSION_FOR_BROADCAST` checker now supports Kotlin. (SAT-31807)
- The `PREDICTABLE_RANDOM_SEED` checker now supports Kotlin. (SAT-31808)
- Added support for OpenSSL's `ssl` API to the `RISKY_CRYPTO` checker. (SAT-32083)
- The `INSECURE_RANDOM` checker now supports Kotlin. `INSECURE_RANDOM` defects of the subcategory `insecure_random_value` have their impact changed from Medium to Low (SAT-32461)
- Added a new option to the `UNINIT_CTOR` checker. The boolean option `report_on_default_constructor_without_private_member`, false by default, and activated at high aggressiveness. This causes the checker to report defects when a compiler-generated default constructor fails to initialize some members, even when no members of the class or struct are private. (SAT-32484)
- Added a new option to the `RISKY_CRYPTO` checker, `usage_report`, that allows gathering information about all cryptographic algorithms used in a codebase into a CSV file. (SAT-32506)
- The `RISKY_CRYPTO` checker now reports 3DES/DES EDE insecure by default. (SAT-32589)
- The `UNENCRYPTED_SENSITIVE_DATA` checker now supports Kotlin (SAT-32677)
- The `PW.PRINTF_ARG_MISMATCH` checker has been disabled by default; please use `PW.PRINTF_ARGS` instead. (SAT-32901)
- The new `ANDROID_WEBVIEW_FILEACCESS` checker finds cases where an Android application allows JavaScript code of files loaded through the `file:///` protocol to load other local files without maintaining the WebView's sandbox. (SAT-33427)
- Upgraded SpotBugs to version 4.0.0. (SAT-33432)

- Added a new event for Go in LOCK checker when a `defer` statement is called. (SAT-33435)
- The new UNSAFE_BUFFER_METHOD checker finds cases where a segment of allocated memory is uninitialized (not zeroed-out), because its content could leak sensitive data from system memory. (SAT-33474)
- New MISSING_HEADER_VALIDATION checker finds cases where the Netty HTTP header validation is disabled, which makes the code vulnerable to HTTP response splitting attacks. (SAT-33475)
- Added an option, disabled by default, `follow_virtual_destructor_classes` that allows the ALLOC_FREE_MISMATCH checker to track instances of classes that have virtual destructors. Use of this option might cause some false positive defect reports when derived classes overload `new` or `delete`. (SAT-33544)
- The new DISABLED_ENCRYPTION checker finds cases where the `noOpText()` method is used, which creates an encryptor object that does not perform any encryption and thus might leak sensitive data. (SAT-33550)
- The new INSECURE_ACL checker finds cases where access control lists (ACLs) are set too permissively in cloud provider configuration. (SAT-33663)
- Table in Chapter 3 of the *Checker Reference* has been updated to show those cases when `--webapp-security` option would enable the checker. (SAT-33693)
- Brakeman version upgraded to 4.8.0. (SAT-33728)
- The new CONFIG.HARDCODED_CREDENTIALS_AUDIT checker finds hardcoded credentials in configuration files in Java, JavaScript, and TypeScript applications. (SAT-33835)
- The HEADER_INJECTION checker now supports Kotlin. (SAT-33897)
- The INSECURE_COMMUNICATION checker now supports Java. (SAT-33903)
- The new checker LDAP_NOT_CONSTANT was added for Java, C#, and Visual Basic. (SAT-33937)
- The new UNLIMITED_CONCURRENT_SESSIONS checker finds cases where the maximum number of concurrent sessions is unlimited. (SAT-33978)
- Added support for the Go web frameworks Echo and Gin. (SAT-34000)
- The new INSECURE_REMEMBER_ME_COOKIE checker finds cases of insecure configuration of the RememberMe cookie, which can be accessed over an HTTP channel. (SAT-34009)
- The new CONFIG.SPRING_SECURITY_UNSAFE_AUTHENTICATION_FILTER checker finds cases where Spring Security frameworks allows credentials to be accepted in a GET request. (SAT-34012)
- The HEADER_INJECTION checker now support Android taints for Java. (SAT-34059)
- Improved reporting in the RISKY_CRYPTO checker when using the `forbid:****` option. Now the checker will include all the crypto parameters it knows about in the event message. (SAT-34083)

- The new `CONFIG.SPRING_SECURITY_EXPOSED_SESSIONID` checker finds cases where the session ids are configured to be sent in URLs. (SAT-34160)
- Updated `CONFIG.SPRING_SECURITY_DEBUG_MODE` checker now flags additional ways in which debug mode has been enabled.. (SAT-34161)
- Support has been added for a few high impact CERT-JAVA rules. (SAT-34176)
- Support has been added for a few high impact CERT-C Recommendation rules on EDG-based compilers. (SAT-34178)
- Improved the quality of the results of the `UNINIT` checker. (SAT-34191)
- Increased the default value of `UNRESTRICTED_ACCESS_TO_FILE:api_level` from 15 to 19 (SAT-34218)
- The new `CONFIG.SPRING_BOOT_SENSITIVE_LOGGING` checker finds cases where a Spring Boot application has been configured to log request cookies or HTTP request details. (SAT-34243)
- The new `CONFIG.SPRING_SECURITY_LOGIN_OVER_HTTP` checker finds cases where the login form of a Spring application is not forced to be accessed over HTTPS. (SAT-34244)
- Added models for the Boost Log library. (SAT-34288)
- Added models for the Boost Icl library. (SAT-34295)
- Extended modeling for the glib library. (SAT-34304)
- Extended modeling for the appweb library. (SAT-34307)
- The new `CUDA.CUDEVICE_HANDLES` checker reports cases where an integer value is used in place of a CUdevice object. (SAT-34336)
- The new `CUDA.ERROR_INTERFACE` checker looks for missing checks of return values from CUDA API functions that might return an error code. (SAT-34338)
- The new `CUDA.DEVICE_DEPENDENT_CALLBACKS` checker reports cases where a device-dependent operation is executed, a kernel is launched, or a managed storage object is ODR-used inside a CUDA callback function. (SAT-34339)
- The new `CUDA.DEVICE_DEPENDENT` checker reports cases where a device-dependent operation is executed, a kernel is launched, or a managed storage object is ODR-used before program initiation has completed, or after program termination has started. (SAT-34340)
- The new `CUDA.FORK` checker reports cases where a CUDA library interface is called, or an object residing in storage allocated by CUDA library interfaces or a managed storage duration object is accessed, between a call to `fork` and a subsequent call to `exec`. (SAT-34344)
- The new `INSECURE_HTTP_FIREWALL` checker finds cases where the HTTP firewall is configured insecurely within the Spring Security framework. (SAT-34434)

- The new `WEAK_URL_SANITIZATION` checker finds cases where weak sanitization of URLs occurs. (SAT-34551)
- C/C++ security checkers like `TAINTED_STRING` now distinguish between taint on pointers and their contents and have improved handling of write and assign operations. (SAT-4838)
- Improved the `UNINIT` checker to handle some cases where an uninitialized variable's address is taken but no initialization happens. (SAT-5286)

14.3.2.2. Bug fixes

CMPCPP-10028

Fixed a false positive of MISRA C-2012 Directive 4.9 on clang-based compilers where `inline` functions couldn't be used to initialize constants.

CMPCPP-10065

Fixed a false positive of MISRA C-2012 Rule 8.3 about identical redefinition using `typedef` on clang-based compilers.

CMPCPP-10111

Fixed a false positive of MISRA C-2012 Directive 4.9 for macros containing only text string substitution.

CMPCPP-8584

Fixed a false positive of MISRA C-2012 Rule 20.7 when macro expansion was not a complete expression.

IM-24998

Fixed a localization issue for some checker and event messages which now shows more translated strings in all three languages.

SAT-14060

Fixed a source of `UNINIT` false positives involving nested field accesses.

SAT-14526

Fixed some false positives with the `UNINIT` checker with the `enable_write_context` option and with MISRA C-2012 Rule 9.1 when initializing arrays in a loop in a callee.

SAT-16111

Fixed some false positives with the `UNINIT` checker with the `enable_write_context` option and with MISRA C-2012 Rule 9.1 when initializing arrays in a loop in a callee.

SAT-19569

Fixed a source of `UNINIT` false positives when computing field offsets.

SAT-20127

Fixed a false positive on the `OVERRUN` checker on `msgsnd` and `msgrcv`.

SAT-20307

Fixed a false positive on the `OVERRUN` checker on `msgsnd` and `msgrcv`.

SAT-21246

Fixed a false positive on the `OVERRUN` checker on `msgsnd` and `msgrcv`.

SAT-26839

Fixed a false positive on the `UNINIT` checker.

SAT-28103

Fixed a source of `UNINIT` false positives when a value is initialized using a cast of an address.

SAT-28489

Fixed a false positive on the `OVERRUN` checker on `msgsnd` and `msgrcv`.

SAT-29161

Fixed some false positives with the `UNINIT` checker with the `enable_write_context` option and with MISRA C-2012 Rule 9.1 when initializing arrays in a loop in a callee.

SAT-30890

Fixed a `USE_AFTER_FREE` false positive when code was compiled for the C++-17 standard.

SAT-30905

Fixed `OVERRUN` checker to update bounds for `disequalities` against a constant.

SAT-30922

Fixed a false negative in the `OVERRUN` checker involving memory buffers of size 1 byte allocated with `malloc()`.

SAT-30953

Fixed an `OVERRUN` false negative where the length of a string was not adequately propagated through `strcpy` calls.

SAT-31580

Eliminated a false positive report from `FORWARD_NULL` when certain constructs involving multiple identical dynamic casts appeared.

SAT-32681

Fixed a false positive for the `RESOURCE_LEAK` checker.

SAT-32872

Fixed a false positive on the `OVERRUN` checker on `msgsnd` and `msgrcv`.

SAT-32885

Updated memory allocations/free functions in glib API models to use `system malloc/free`.

SAT-33356

Fixed an inconsistency in the way the `OVERRUN` checker option `allow_arrays_of_uniform_structs` treats direct accesses vs. pointer accesses.

SAT-33428

Fixed a source of `MISSING_COPY_OR_ASSIGN` false positives when the assignment operator or copy constructor is explicitly deleted in a base class.

SAT-33450

Fixed a false negative for the `SLEEP` checker.

SAT-33485

Fixed a false positive for the `CTOR_DTOR_LEAK` checker.

SAT-33589

Fixed a false positive on the `OVERRUN` checker on `msgsnd` and `msgrcv`.

SAT-33737

Improved the `UNINIT` checker to avoid incorrectly detecting some assembly code as initializing variables that it does not actually reference.

SAT-33881

Added support for the `javax.net.ssl.SSLContext` API to the `RISKY_CRYPTO` checker.

SAT-33908

A false positive for the `CONFIG.ATS_INSECURE` checker has been fixed.

SAT-33979

Fixed an issue that was causing the analysis to fail in certain cases when the `XML_EXTERNAL_ENTITY` checker was enabled.

SAT-34108

Fixed a source of `FORWARD_NULL` false positives when accessing an outer class member in an inner class with a delegating constructor.

SAT-34408

Fixed a recoverable analysis crash mentioning "`CHECKED_ARGUMENT_FOR_MISRA`" when running MISRA checkers on some code involving floating point operations.

SAT-34505

Fixed a case where `FORWARD_NULL` would report a false positive when a null reference was reassigned within a lambda function that captured the reference.

SAT-34792

Fixed a source of `STRAY_SEMICOLON` false positives when using the C++ `if constexpr` construct and a clang-based compiler.

SATW-2170

Fixed a false negative of MISRA C-2012 Directive 4.14 regarding standard document cases.

SATW-3276

Fixed a false positive of CERT INT30-C about a multiply operation after casting into a bigger-size integer.

SATW-3391

Fixed false positives of CERT INT30-C and CERT INT32-C about an addition operation after casting into a bigger-size integer.

SATW-3403

Fixed a false negative of CERT INT31-C about invalid conversion related to types with qualifiers.

SATW-3487

Fixed a false positive of AUTOSAR C++14 A8-4-5 where `std::exchange` was used on scalar types.

SATW-3510

Fixed a false positive of MISRA C-2012 Rule 10.3 about defects in a standard header file. Also fixed a false positive of MISRA C-2012 Rule 17.7 about compound expressions.

SATW-3527

Fixed a false positive of MISRA C-2012 Rule 9.1 about `__gettimeofday`.

SATW-3528

Fixed a false positive for MISRA C-2012 Rule 9.1.

SATW-3532

Fixed a false positive of MISRA C-2012 Rule 10.6 about variables declared as `const`.

SATW-3533

Fixed a false positive in MISRA C-2012 Rule 9.1/UNINIT with "enable_write_context" when a write was predicated on 2 pointers being non-null.

SATW-3554

Fixed a false positive of AUTOSAR C++14 A12-1-3 where not all data members were initialized with `const` values.

SATW-3555

Fixed a false positive of CERT INT31-C about bit fields.

SATW-3556

Fixed a false positive of CERT INT31-C about casts in parameters and the return value when calling compiler-generated functions.

SATW-3557

Suppressed AUTOSAR C++14 A8-4-7 reporting on a template class member function.

SATW-3573

Fixed a false positive of MISRA C++-2008 Rule 5-0-15 about using array indexing on an array parameter.

SATW-3574

Fixed a false positive of CERT EXP37-C when the called function was prototyped.

SATW-3576

Suppressed AUTOSAR C++14 A8-4-11 and A8-4-13 reporting on template classes and functions.

SATW-3578

Fixed an AUTOSAR C++14 A8-4-9 false positive where the "in-out" parameter was used to initialize a class non-const reference member.

SATW-3585

Fixed a false positive of CERT INT30-C about adding an integer literal after a right-shift operation.

SATW-3587

Fixed a false positive of CERT DCL37-C about using a reserved identifier in a standard header file.

SATW-3589

Fixed a false positive of CERT MEM55-CPP where an overloaded function has no throw specifier and does not throw exceptions.

SATW-3607

Fixed a false positive of MISRA C-2012 Directive 4.7 where a function call was tested by the equality operator.

SATW-3613

Fixed the presentation issue of MISRA C-2004 Rule 2.2 for Chinese characters.

SATW-3614

Fixed a false positive of AUTOSAR C++14 A9-6-1 where a type was redefined.

SATW-3618

Fixed a false positive of CERT INT30-C about subtract operation after left-shifting constant bits.

SATW-3620

Fixed a false positive of CERT INT31-C about casting type after sufficient right-shift operation.

SATW-3625

Fixed a false positive of MISRA C-2012 Rule 11.1 caused by the wrapped type.

SATW-3633

Fixed a false positive of MISRA C++-2008 Rule 0-1-2 where equality operator was used in `if` statement.

SATW-3671

Fixed a false positive of MISRA C++-2008 Rule 0-1-10 about overloading `new` operator.

SATW-3720

Fixed a false positive of CERT MEM52-CPP about compiler generated variables.

14.3.2.3. 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.

14.3.3. Coverity Commands 2020.06

14.3.3.1. New or changed features

- Added limited support of `cov-archive` in a cluster Coverity Connect installation. The limitation is: importing to a subscriber node is forbidden. (IM-24896)

14.3.3.2. Bug fixes

INS-2861

Due to a mismatch between creation timestamps and version numbers in some Incremental Release packages, it was possible for `cov-commit-defects` to report that an update was available when in fact no applicable update existed. This test now uses the actual installed version for improved accuracy.

PRD-12087

Fixed `cov-wizard` result view link error

SAT-33483

Fixed a crash in `cov-format-errors` that happened when the `--json-output-v7` option was used in combination with a missing source location in a SpotBugs report.

SAT-33906

`cov-make-library` calls specifying a compiler and additional `--compiler-opt` options could fail with an incorrect command line error. This has been fixed.

SAT-34182

The `--ticker` mode option of the `cov-run-desktop` command has been removed to fix a bug.

14.3.3.3. 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.

14.3.4. Coverity Compilers and Capture 2020.06

14.3.4.1. End-of-life products

COVP-2223

Support for FreeBSD 11.2 has been dropped.

14.3.4.2. Deprecated products and features

CMPG-3236

Support for LLVM Clang 3.0–3.6.x is deprecated as of 2020.06 and will be removed in a future release.

CMPG-3257

Deprecated support for Go 1.12.x.

CMPG-3259

Deprecated support for Apple Clang 6.0 (Xcode 6.0–6.2) and Apple Clang 6.1 (Xcode 6.3–6.4)

CMPG-3268

Deprecated support for Ruby 2.3 and 2.4.

CMPG-3273

Support for IBM XLC versions 8–12 are deprecated as of 2020.06 and will be removed in a future release.

CMPG-3279

Support for Linux versions of Intel C++ older than version 17 are deprecated as of 2020.06 and will be removed in a future release.

CMPG-3280

Support for Keil Arm compiler RVCT 3.1, 4.0 for uVision is deprecated as of 2020.06 and will be removed in a future release.

COVP-2222

Support for mac OS 10.13 has been deprecated as of 2020.06 and will be removed in a future release.

COVP-2227

Support for Oracle JDK 13 has been deprecated as of 2020.06 and will be removed in a future release.

COVP-2229

Support for OpenJDK 13 has been deprecated as of 2020.06 and will be removed in a future release

14.3.4.3. New or changed features

- Added support for clang-cl 7.0 on Windows. (CMPCPP-3662)
- Added support for the MetaWare ccac Q-2019.12 compiler. (CMPCPP-9048)
- Added support for the TASKING TriCore version 6.0r1 compiler. (CMPCPP-9497)
- Added support for the Renesas C/C++ RX version 3.01 compiler. (CMPCPP-9622)
- Added support for the GNU GCC and G++ version 9.2.0 compiler. (CMPCPP-9821)
- Added support for Kotlin 1.3.71. (CMPFG-394)
- The JavaScript front end now supports all ES10 (ECMAScript 2019) syntax. (CMPG-3220)
- Added support for Go 1.13 and 1.14. (CMPG-3221)
- Added support for TypeScript 3.8. (CMPG-3228)
- Added support for IBM JDK 7 through 8. (CMPG-3231)
- Clarification added to documentation that Coverity only supports Kotlin projects that are targeted to JVM or Android, not other platforms. For multiplatform projects, Coverity only captures Kotlin source files that are targeted to the supported platforms. (CMPG-3239)
- Added support for Apple Clang 11 (Xcode 11.4) (CMPG-3247)
- Added support for LLVM Clang version 10.0 (CMPG-3272)
- Support has been added for the CUDA programming language and the NVIDIA nvcc compiler when used with GNU gcc or Microsoft Visual C++ as the host compiler. (CMPG-3291)

- Added support for Go 1.13-1.14.x. (CMPGO-151)
- `cov-emit-go` now properly handles Go modules. (CMPGO-90)
- Optional chaining syntax is now supported by the TypeScript front end. (CMPJS-775)
- Xcode 11 is now supported (CMPSWIFT-301)
- Added support for macOS 10.15. (COVP-2146)
- SCM support for Accurev 7.3 has been added. (COVP-2192)
- Support for IBM JDK 7, 7.1 and 8 has been added. (COVP-2225)
- To improve performance third party license files will now be contained in a zip file at `doc/licenses/coverty-thirdparty-licenses.zip`. (INS-2851)

14.3.4.4. Bug fixes

CMPCPP-10003

A bug was fixed in the case when `cov-build` did not work with `gcc-10`.

CMPCPP-10038

Fixed an issue where `cov-emit` can't recognize `va_list` for the IAR ARM compiler.

CMPCPP-10068

Assertion "Trying to access type `std::array<unsigned char, 1ul>` as a array type" has been eliminated.

CMPCPP-10074

Fixed an issue where Coverity failed with an invalid redeclaration of type name `__istate_t` for the IAR STM8 compiler.

CMPCPP-10139

An issue was fixed in the compilation of code that uses Qt and protocol buffers using 2020.03.

CMPCPP-8034

Special support for the `WARN_ON`, `BUG_ON`, and `BUG` macros present in Linux kernel development is no longer in effect when not compiling Linux kernel code.

CMPCPP-8573

Fixed a bug in MISRA C 2012 Directive 4.9 checker where a macro with properties that qualify it as not function-like was being flagged as a violation when used as an argument to another macro.

CMPCPP-9778

A bug was fixed that created errors when the underlying type of `enum` is not `scalar`.

CMPCPP-9830

Fixed an issue where `cov-emit` didn't set macro `__ARM_FEATURE_CMSE` correctly with `--cmse` option for the IAR ARM compiler.

CMPCPP-9942

Fixed a bug where MISRA 2008 C++ compliance rule 0-1-5 reported a FP defect due to unused lambda closure class type

CMPCPP-9993

Fixed an issue where `cov-emit` can't support `static_assert` in c mode for the IAR ARM compiler.

CMPCSH-1334

An issue was fixed for the capture of a C# project .

CMPCSH-1360

Addressed an issue in the emit where large logical chains of C# dynamic types could negatively impact performance.

CMPCSH-1370

A bug has been fixed that caused stalled C# builds.

CMPCSH-1379

An issue has been addressed that resulted in increased build times for C# code.

CMPCFG-388

Kotlin front end now properly supports Kotlin serialization compiler plugin.

CMPCG-2822

A bug was fixed for numerous parse warnings that were due to Coverity preprocessing removing whitespace incorrectly.

CMPJ-1252

We can now handle some error cases encountered during Java file-system capture more gracefully so that these cases do not result in a failure to emit entire sets of source files.

CMPJ-1272

We can now handle an error case encountered during Java build capture more gracefully so that this case does not result in a failure to emit entire sets of source files.

CMPJ-1319

Fixed a problem in which Java builds with a `--release` value below 9 were compiled with modular type-resolution semantics.

CMPJS-744

A bug was fixed that caused TypeScript recursive function type declarations to cause stack overflow.

CMPJS-773

The Coverity front end for TypeScript does not have support for the `esnext` module type, and has historically fallen back to emitting such TUs with the default module type. It will now fall back to emitting those TUs with the `commonjs` module type instead.

CMPJS-805

The logging output produced by the JavaScript/TypeScript front end has been improved to provide a consistent presentation and more context.

INS-2898

A bug was fixed that led to no files getting emitted and no files getting analyzed for the Scala compiler.

SAT-32079

Fixed an issue (on a 64-bit platform compiling using `-m32`) that caused Coverity to think that `msgrcv()` is overrunning the stated buffer size by four bytes.

14.3.4.5. 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.

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.

14.3.5. Coverity Dynamic Analysis 2020.06

14.3.5.1. 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.

14.3.6. Coverity Test Advisor 2020.06

14.3.6.1. End-of-life products

COVP-2233

Support for Mercurial 3.1 and 3.2 has been dropped.

COVP-2235

Support for Perforce 2016.1 has been dropped.

14.3.6.2. Deprecated products and features

COVP-2232

Support for Perforce 2016.2 is deprecated as of 2020.06 and will be removed in a future release.

14.3.6.3. New or changed features

- SCM support for Git 2.26 has been added. (COVP-2230)

14.3.6.4. Known issues and solutions

TADE-2033

The use of `--cs-coverage opencover` with Test Advisor may 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. This can be worked around by manually registering the OpenCover profiler DLLs and passing `--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 on this, see the documentation of `cov-build`'s `--cs-no-register-profiler` switch in the 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.

14.3.7. Coverity Wizard 2020.06

14.3.7.1. Bug fixes

PRD-12114

A bug was fixed to make Buildless capture Settings Page scrollable.

14.3.7.2. 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.

14.4. Coverity Desktop 2020.06

This section provides release notes for Coverity Desktop components.

14.4.1. Coverity Desktop for Android Studio 2020.06

14.4.1.1. New or changed features

- Added support to Android Studio 3.6 (PRD-12056)

14.4.1.2. 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.

14.4.2. Coverity Desktop for Eclipse 2020.06

14.4.2.1. Deprecated products and features

PRD-12107

Support for Eclipse 4.7 has been deprecated

14.4.2.2. New or changed features

- Added support for Eclipse 2020-03 (4.15) (PRD-12089)

14.4.2.3. 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.exe` 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/`.

14.4.3. Coverity Desktop for IntelliJ IDEA 2020.06

14.4.3.1. New or changed features

- Added support for IntelliJ 2020.1 (PRD-12102)
- Added support for Pycharm 2020.1 (PRD-12103)
- Added support for WebStorm 2020.1 (PRD-12104)
- Added support for Rubymine 2020.1 (PRD-12105)
- Added support for PhpStorm 2020.1 (PRD-12106)

14.4.3.2. 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.

14.5. Coverity Documentation 2020.06

This section provides release notes for Coverity Documentation components.

14.5.1. Coverity Documentation 2020.06

14.5.1.1. New or changed features

- The list of supported compilers for Dynamic Analysis has been moved to "Appendix A. Coverity Dynamic Analysis for Java". (SAT-33715)

14.5.1.2. Bug fixes

COVDOCS-59

Information about the `projects` view type has been added to section 5.2.1 of the *Coverity Platform Web Services API Reference*.

COVDOCS-68

Updated the fonts used in PDF output for Chinese Simplified documentation. This fixed an issue by which some characters did not render properly.

COVDOCS-70

The broken hyperlinks in section "2.3.4. Troubleshooting for FLEXnet licensing" of the *_Coverity 2020.06 Deployment and Installation Guide* have been replaced with a single working hyperlink.

COVDOCS-79

Table 8.10. "Frameworks supported by Coverity" in the *Coverity 2020.03 Deployment and Installation Guide* incorrectly listed "Sprint boot". That listing has been corrected to "Spring boot".

COVDOCS-80

A missing image was restored to *Coverity Desktop Analysis Guide*.

14.5.1.3. 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*.

Chapter 15. Coverity 2020.03-8 Release Notes

Table of Contents

15.1. Important information for 2020.03-8	64
15.2. Coverity Analysis 2020.03-8	64

15.1. Important information for 2020.03-8

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.

15.2. Coverity Analysis 2020.03-8

This section provides release notes for Coverity Analysis components.

15.2.1. Coverity Compilers and Capture 2020.03-8

15.2.1.1. Bug fixes

CMPCPP-10872

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

Chapter 16. Coverity 2020.03-7 Release Notes

Table of Contents

16.1. Important information for 2020.03-7	65
16.2. Coverity Platform 2020.03-7	65

16.1. Important information for 2020.03-7

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.

16.2. Coverity Platform 2020.03-7

This section provides release notes for Coverity Platform components.

16.2.1. Coverity Connect 2020.03-7

16.2.1.1. Bug fixes

IM-25471

Fixed user discrepancies in the triage panel owner autocomplete

Chapter 17. Coverity 2020.03-6 Release Notes

Table of Contents

17.1. Important information for 2020.03-6	66
17.2. Coverity Analysis 2020.03-6	66

17.1. Important information for 2020.03-6

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.

17.2. Coverity Analysis 2020.03-6

This section provides release notes for Coverity Analysis components.

17.2.1. Coverity Commands 2020.03-6

17.2.1.1. Bug fixes

IM-25150

Fixed a bug in `cov-archive` that in rare cases prevented importing archives with "unique constraint violation" error.

Chapter 18. Coverity 2020.03-5 Release Notes

Table of Contents

18.1. Important information for 2020.03-5	67
18.2. Coverity Platform 2020.03-5	67

18.1. Important information for 2020.03-5

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.

18.2. Coverity Platform 2020.03-5

This section provides release notes for Coverity Platform components.

18.2.1. Coverity Connect 2020.03-5

18.2.1.1. Bug fixes

IM-25097

Fixed persistent XSS issue involving display of user names. Fixed access control issue for web services backup configuration APIs.

Chapter 19. Coverity 2020.03-4 Release Notes

Table of Contents

19.1. Important information for 2020.03-4	68
19.2. Coverity Platform 2020.03-4	68

19.1. Important information for 2020.03-4

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.

19.2. Coverity Platform 2020.03-4

This section provides release notes for Coverity Platform components.

19.2.1. Coverity Connect 2020.03-4

19.2.1.1. Bug fixes

IM-24970

Improved the performance of the following methods of the Defect Service Web Service: `getMergedDefectsForProjectScope`, `getMergedDefectsForStreams`, `getMergedDefectsForSnapshotScope`. This resulted in improving the performance of the `cov-manage-im` command. In some internal tests the latencies were reduced from about 120 seconds to about 8 seconds.

//This RN is a subset of the RN specified in <https://jira-sig.internal.synopsys.com/browse/IM-24871> for 2020.09 (Upland).

Chapter 20. Coverity 2020.03-4 Release Notes

Table of Contents

20.1. Important information for 2020.03-4	69
20.2. Coverity Analysis 2020.03-4	69

20.1. Important information for 2020.03-4

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.

20.2. Coverity Analysis 2020.03-4

This section provides release notes for Coverity Analysis components.

20.2.1. Coverity Compilers and Capture 2020.03-4

20.2.1.1. Bug fixes

CMPFG-418

Fixed an issue where `cov-emit-java` crashed on long command line switches.

CMPJ-1332

We can now handle an error case encountered during Java build capture more gracefully so that this case does not result in a failure to emit entire sets of source files.

CMPJS-815

A bug was fixed that caused TypeScript recursive function type declarations to cause stack overflow.

Chapter 21. Coverity 2020.03-3 Release Notes

Table of Contents

21.1. Important information for 2020.03-3	70
21.2. Coverity Analysis 2020.03-3	70

21.1. Important information for 2020.03-3

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.

21.2. Coverity Analysis 2020.03-3

This section provides release notes for Coverity Analysis components.

21.2.1. Coverity Checkers 2020.03-3

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*.

21.2.1.1. Bug fixes

CMPCPP-10206

Fixed a false positive of MISRA C-2012 Rule 20.7 when macro expansion was not a complete expression.

Chapter 22. Coverity 2020.03-2 Release Notes

Table of Contents

22.1. Important information for 2020.03-2	71
22.2. Coverity Platform 2020.03-2	71
22.3. Coverity Analysis 2020.03-2	72

22.1. Important information for 2020.03-2

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.

22.2. Coverity Platform 2020.03-2

This section provides release notes for Coverity Platform components.

22.2.1. Coverity Compliance Solution 2020.03-2

Coverity Compliance Solution helps quality managers and architects manage coding standards projects, those using MISRA, CERT, or AUTOSAR standards, which typically surface large numbers of findings. Using Compliance Solution, developers can focus on the most important issues and even prioritize these.

If you use coding standards and find you have a larger number of defects than you can comfortably handle, the Compliance Solution will let you do the following:

- Visualize large numbers of findings and make decisions about how to handle them
- Use those decisions to filter findings, excluding all that are not of interest now
- Upload only the interesting findings to Coverity Connect

Compliance Solution is now in a beta phase. For documentation, tutorials, and information about the beta program, please see the solution's Community page: <https://community.synopsys.com/s/coverity-compliance-solution>

22.2.1.1. Bug fixes

COMP-441

Files or directories displayed under incorrect parent node when you drill down by path in certain cases.

COMP-451

Findings don't match upon drilling down in certain cases.

22.3. Coverity Analysis 2020.03-2

This section provides release notes for Coverity Analysis components.

22.3.1. Coverity Checkers 2020.03-2

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*.

22.3.1.1. Bug fixes

SATW-3702

Fixed a false positive of MISRA C-2012 Rule 11.1 caused by the wrapped type.

SATW-3703

Fixed a false positive of AUTOSAR C++14 Rule 9-6-1 where a type was redefined.

22.3.2. Coverity Compilers and Capture 2020.03-2

22.3.2.1. Bug fixes

CMPJS-808

The logging output produced by the JavaScript/TypeScript front end has been improved to provide a consistent presentation and more context.

Chapter 23. Coverity 2020.03-1 Release Notes

Table of Contents

23.1. Important information for 2020.03-1	73
23.2. Coverity Analysis 2020.03-1	73

23.1. Important information for 2020.03-1

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.

23.2. Coverity Analysis 2020.03-1

This section provides release notes for Coverity Analysis components.

23.2.1. Coverity Commands 2020.03-1

23.2.1.1. Bug fixes

SAT-33934

`cov-make-library` calls specifying a compiler and additional `--compiler-opt` options could fail with an incorrect command line error. This has been fixed.

23.2.2. Coverity Compilers and Capture 2020.03-1

23.2.2.1. Bug fixes

CMPCSH-1375

Fixed a bug in which unexpected Exception (C# Code) were suppressed

CMPJ-1280

We are now handling some error cases encountered during Java file-system capture more gracefully so that these cases do not result in a failure to emit entire sets of source files.

Chapter 24. Coverity 2020.03 Release Notes

Table of Contents

24.1. Important information for 2020.03	74
24.2. Coverity Platform 2020.03	74
24.3. Coverity Analysis 2020.03	80
24.4. Coverity Desktop 2020.03	94
24.5. Coverity Documentation 2020.03	96

24.1. Important information for 2020.03

Support for this version of Coverity will be discontinued 18 months after the 2020.12 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.

24.2. Coverity Platform 2020.03

This section provides release notes for Coverity Platform components.

24.2.1. Coverity Compliance Solution 2020.03

Coverity Compliance Solution helps quality managers and architects manage coding standards projects, those using MISRA, CERT, or AUTOSAR standards, which typically surface large numbers of findings. Using Compliance Solution, developers can focus on the most important issues and even prioritize these.

If you use coding standards and find you have a larger number of defects than you can comfortably handle, the Compliance Solution will let you do the following:

- Visualize large numbers of findings and make decisions about how to handle them
- Use those decisions to filter findings, excluding all that are not of interest now
- Upload only the interesting findings to Coverity Connect

Compliance Solution is now in a beta phase. For documentation, tutorials, and information about the beta program, please see the solution's Community page: <https://community.synopsys.com/s/coverity-compliance-solution>

24.2.1.1. Bug fixes

COMP-350

The Help page does not contain documentation or tutorial links. Workaround: You can get the documentation and tutorials from the Solution's Synopsys Community page.

COMP-401

The `cov-upload-findings` command is slow for large intermediate directories, that is, those with more than about 100,000 findings.

COMP-429

Improvement of page load performance on Visualize Page upon adding a filter policy.

24.2.1.2. Known issues and solutions

COMP-351

The threshold control on the Filter Policies/Threshold page does not work on some versions of Microsoft Edge browser. Workaround: Use a different browser.

COMP-386

The installer that the bootstrap script runs quits if you press Enter too many times during the display of the End User License Agreement. You can work around this by pressing 'q' once while the EULA is displayed.

COMP-417

If Findings Manager host's `hostname` command returns a name that is not known to DNS, Findings Manager does not know what streams are available from Coverity Connect, even though Coverity Connect is connected to the message bus. Workaround: Edit the file `compliance-solution-2020.03-CSPBETA-####/config/environment`. On the line that starts `KAFKA_HOST=`, replace the `hostname` with the host's IP address. Then restart the compliance solution with the following command: `cd ../bin; message-bus up && findings-manager up`

COMP-420

When you run `cov-upload-findings`, please ignore the warning message that says `no EndPointIdentificationAlgorithm has been configured for SslContextFactory`.

COMP-454

When you run `cov-upload-findings`, please ignore the warning message that says `no EndPointIdentificationAlgorithm has been configured for SslContextFactory`.

24.2.2. Coverity Connect 2020.03

24.2.2.1. End-of-life products

IM-24759

Dropped support for Windows 7.

24.2.2.2. New or changed features

- A new filter and column named Score is available on views of type "Issues: By Snapshot". This filter/column represents the score assigned to findings by Compliance Solution scoring policies. These scores are retained when Coverity Connect converts the findings to issues. For more information, refer to the *Coverity Compliance Solution Guide*. (COVDOCS-36)

Coverity Compliance Solution is currently part of the Compliance Solution Beta program and is available only to program participants. For more information about the Compliance Solution Beta program, see the section "Compliance Solution". (COVDOCS-36)

- The `cov-archive` command can now delete exported streams (for example, `cov-archive export-streams --remove --stream s1 --project p1 --archive ../s1_p1.covarch`) thus supporting archiving semantics rather than only exporting as it was in the previous release. See *Coverity Command Reference* for more details. (IM-23992)
- Added build success information to the WS API `getSnapshotInformation` method (IM-24103)
- Added an ability to configure Trend ETL process scheduling separately from Status ETL process scheduling. (IM-24368)
- Added triage-store information to the triage event notifications (IM-24569)
- Exporting via `cov-archive` no longer requires putting Coverity Connect into maintenance mode. (IM-24590)
- Documented the ability to provide event-driven triage notification. The administrator can use event-driven triage notifications to send notifications to issue owners about all triage event changes as they happen. These notifications are filterable and configurable. (IM-24592)
- The Coverity Connect silent installer option `--admin.password` has been deprecated in favour of two new options `--admin.password.env` and `--admin.password.file`, which don't require the password to be entered directly on the command line. (INS-2860)

24.2.2.3. Bug fixes

IM-20661

Documentation has been updated to describe the defect status `Absent` `Dismissed`.

IM-22002

Documentation has been updated to warn against setting up more than one LDAP configuration.

IM-22830

Fixed a bug that prevented component map import.

IM-23166

Docs have been updated to reflect the fact that a project manager can preview commits to a stream.

IM-23659

The View API `views` method now reports shared views.

IM-23754

Documentation was amended in the Japanese version to warn that admins can use external DB only if they are an experienced database administrator.

IM-23938

Coverity Connect again supports keystores in pkcs12 format.

IM-23942

A bug was fixed for a situation in which repeatedly soft-deleting users caused user names so long that DB errors were raised.

IM-24142

Fixed a bug in the functionality of linking a dynamic stream to a project. As a result, importing dynamic streams via the `cov-archive` command is now supported.

IM-24285

A bug has been fixed whereby a user who lacked appropriate permissions was able to access data.

IM-24299

Fixed a time stamp inconsistency issue when the Triage Store was exported and imported between different time zones.

IM-24451

The `getMergedDefectsForProjectsScope` API can now retrieve Impact & other information on Preview Defects .

IM-24546

A bug was fixed in which autocompleted users query was using too much working memory.

IM-24619

Documentation has been updated to note the need to set `java.io.tmpdir` and/or `jna.tmpdir` when file systems are mounted `noexec`.

IM-24644

A bug was fixed where the Coordinator was running out of memory after a 2019.12 upgrade.

IM-24765

Drastically improved performance of the `cov-archive import-streams` command. In our tests importing times went from days to about an hour. Performance of the `cov-archive export streams` command was slightly improved.

INS-2852

MacOS analysis installer now checks for space character in path.

24.2.2.4. 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-23994

Internet Explorer 11 breaks on functionalities using file upload.

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 Java1.7.0_xx and older, `OutOfMemory` 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
```

24.2.3. Coverity Report Generators 2020.03

24.2.3.1. New or changed features

- The Report Generator now refers to the 2019 version of the sans25 Application Security Risks. (RG-1368)

24.2.3.2. Bug fixes

RG-1324

Fixed a bug in which `cov-generate-cvss-report` did not count `WEAK_GUARD` (CWE-291) as a CWE/SANS Top 25 vulnerability.

RG-1359

Fixed an error about an invalid configuration file that complained about missing properties when those properties were not actually relevant to the given report.

24.2.3.3. 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 install libgl1-mesa-dri`, and `apt-get install 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 Poretcode SC.

24.3. Coverity Analysis 2020.03

This section provides release notes for Coverity Analysis components.

24.3.1. Coverity Checkers 2020.03

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*.

24.3.1.1. Deprecated products and features

SAT-33177

The `FILE_OPEN_MODE` sink type was removed.

24.3.1.2. New or changed features

- Added issue view columns for standards (AUTOSAR C++14, CERT C, CERT C++, DISA-STIG V4R3, ISO TS17961 2016, OWASP Mobile Top Ten 2016, OWASP Web Top Ten 2017, PCI DSS 2018). (IM-24530)
- The `SENSITIVE_DATA_LEAK` checker now supports C and C++. (SAT-26090)
- A checker option, `allow_array_of_uniform_structs`, has been added to the `OVERRUN` checker. This option suppresses defect reports when filling an entire array of structures from a pointer to one structure. (SAT-27510)
- The `REVERSE_INULL` checker will no longer report on pointers that have been asserted as non-NULL. (SAT-29174)
- For languages that implicitly initialize member references to null in a constructor, we now explicitly treat those references as initialized to null, rather than unknown/uninitialized. (SAT-31288)
- Defects that were previously reported as `BUFFER_SIZE_WARNING` are now reported as `BUFFER_SIZE`, with a distinguishing subcategory. (SAT-31373)
- The `ATOMICITY` checker now supports Go. (SAT-31554)
- The `LOCK` checker now supports Go. (SAT-31555)

- The `GUARDED_BY_VIOLATION` checker now supports Go. (SAT-31556)
- The `LOCK_INVERSION` checker now supports Go. (SAT-31557)
- The `SLEEP` checker now supports Go. (SAT-31558)
- The `HEADER_INJECTION` checker now supports C, C++, Objective C, and Objective C++. (SAT-31576)
- The `HARDCODED_CREDENTIALS` checker now supports Kotlin. (SAT-31766)
- The `SENSITIVE_DATA_LEAK` checker now supports Kotlin. (SAT-31772)
- The `MISSING_PERMISSION_ON_EXPORTED_COMPONENT` checker now supports Kotlin. (SAT-31795)
- The `ANDROID_DEBUG_MODE` checker now supports Kotlin. (SAT-31796)
- The `CONFIG.ANDROID_BACKUPS_ALLOWED` checker now supports Kotlin. (SAT-31797)
- The `CONFIG.ANDROID_OUTDATED_TARGETSDKVERSION` now supports Kotlin. (SAT-31798)
- The `CONFIG.ANDROID_UNSAFE_MINSDKVERSION` now supports Kotlin. (SAT-31799)
- The `BAD_CERT_VERIFICATION` checker now supports Kotlin. (SAT-31803)
- The `INSECURE_REFERRER_POLICY` checker for JavaScript and TypeScript finds cases where the `Referer-Policy` HTTP header is set to certain values that might leak `Referer` header across origins. (SAT-32198)
- The new `MULTER_MISCONFIGURATION` JavaScript and TypeScript checker finds different cases of insecure configuration of the module `multer`. (SAT-32248)
- You can now use code annotations in C/C++ to completely suppress a defect instead of only triaging it. (SAT-32272)
- Coverity now supports Detekt analysis (SAT-32309)
- The `CONFIG.MISSING_GLOBAL_EXCEPTION_HANDLER` checker now supports JavaScript and TypeScript. (SAT-32398)
- The new `DNS_PREFETCHING` checker for JavaScript and TypeScript finds situations where DNS prefetching is enabled by setting the `allow` property explicitly to `true` in the `dnsPrefetchControl()` function of the `helmet` middleware or in the configuration of the `dns-prefetch-control` middleware. (SAT-32402)
- The new `HPKP_MISCONFIGURATION` checker for JavaScript and TypeScript finds cases of the HTTP Public Key Pinning (HPKP) insecure configuration using modules `helmet` and `hpkp`. (SAT-32432)
- The new `FILE_UPLOAD_MISCONFIGURATION` checker for JavaScript and TypeScript finds cases where the `express-fileupload` plugin for an Express application is misconfigured and might allow a denial of service attack. (SAT-32512)

- The `CONFIG.ENABLED_DEBUG_MODE` checker now supports JavaScript and TypeScript. (SAT-32515)
- The new `ANGULAR_SCE_DISABLED` JavaScript and TypeScript checker finds cases where Strict Contextual Escaping (SCE) is explicitly disabled. (SAT-32659)
- The `INSECURE_COMMUNICATION` checker now supports JavaScript and TypeScript. (SAT-32779)
- The `BAD_CERT_VERIFICATION` checker now supports JavaScript and TypeScript. (SAT-33061)
- The new `AWS_VALIDATION_DISABLED` checker for JavaScript and TypeScript finds cases where the `aws-sdk` middleware disables parameters or credentials validation globally. (SAT-33068)
- The `CONFIG.UNSAFE_SESSION_TIMEOUT` checker now supports JavaScript and TypeScript. (SAT-33071)
- Added support for the A rules (Ax-x-x, where x is literal number, for example A0-1-1) in AUTOSAR C++14 compliance standard for Clang-based compilers. (SAT-33078)
- For languages that implicitly initialize member references to null in a constructor, we now explicitly treat those references as initialized to null, rather than unknown/uninitialized. (SAT-33117)
- The `RISKY_CRYPTO` checker now finds insecure SSL ciphers in JavaScript and TypeScript. (SAT-33124)
- The name of the `AWS_INSUFFICIENT_PRESIGNED_URL_TIMEOUT` checker was changed to `INSUFFICIENT_PRESIGNED_URL_TIMEOUT` and now supports AWS and Google cloud providers. (SAT-33134)
- The new `AWS_SSL_DISABLED` checker for JavaScript and TypeScript finds cases where the `sslEnabled` property is set to `false` in AWS configuration. (SAT-33135)
- The new `TEMPORARY_CREDENTIALS_DURATION` checker for JavaScript and TypeScript finds cases where cloud providers create temporary credentials that last longer than necessary. (SAT-33137)
- The `CONFIG.CORDOVA_EXCESSIVE_LOGGING` and `CONFIG.CORDOVA_PERMISSIVE_WHITELIST` checkers are now enabled when the `cov-analyze --webapp-security` option is specified. (SAT-33138)
- The new `CONFIG.HARDCODED_TOKEN` checker for JavaScript and TypeScript finds tokens and keys stored directly in configuration files. (SAT-33148)
- Added modeling for the `bzip2` library. (SAT-33179)
- Enhanced our modeling of the `GoAhead` library. (SAT-33209)
- Added modeling for `libcurl`. (SAT-33210)
- Added models for `boost/container` library. (SAT-33211)
- SpotBugs 3.1.10 has been patched to support the analysis of Java 13 classes. (SAT-33322)

- Added a new checker, `ODR_VIOLATION`, which finds code that violates the C++ "one definition rule". Also greatly improved results for MISRA C++-2008 Rule 3-2-2 (SAT-7075)

24.3.1.3. Bug fixes

CMPCPP-9413

Fixed false positive defects for compliance checker MISRA C++2008 Rule 14-6-2 for copy constructors and copy assignment operators generated by the compiler.

SAT-23344

Documentation for checker options by aggressiveness level has been updated.

SAT-29151

Fixed a source of `DEADCODE` false positives when a function whose semantics is `min` has `max` in its name (for example, `clamp_max`)

SAT-30441

Fixed a class of `USE_AFTER_FREE` false negatives, where calls of `free` depend on an allocation policy.

SAT-30875

Added models for the glib functions `g_variant_print()` and `g_variant_print_string()` so that a `FORWARD_NULL` is reported when these functions are called with a `NULL` pointer in the first parameter.

SAT-31548

Fixed a false positive for the `NO_EFFECT` Checker when the second argument to `memset` fits a signed character.

SAT-32508

A model was added for the QNX `pthreads` extension `pthread_mutex_trylock_monotonic()`. This avoids false positives in the `LOCK` checker due to the formerly unrecognized mutex locking function.

SAT-32701

Fixed a source of `OVERRUN` false positives when a function accesses a buffer using a minimum value multiplied by a constant.

SAT-32708

A false positive for the `SQLI` checker has been fixed.

SAT-32740

A model was added for the QNX `pthreads` extension `pthread_mutex_trylock_monotonic()`. This avoids false positives in the `LOCK` checker due to the formerly unrecognized mutex locking function.

SAT-32776

Fixed an error message issue for the `NULL_RETURN` checker when the checker's `stat_threshold` option is set to zero and the `allow_unimpl_and_unchecked` option is set to `true`.

SAT-32858

Fixed a false positive on the `FORWARD_NULL` checker when calling a method on a nil pointer.

SAT-32897

The `OVERRUN` checker now gives better messages when an array is accessed at a constant element index in a callee; it now prints the element index rather than just the byte index.

SAT-32912

Fixed a source `NULL_RETURNS` false positives with null-returning operator `new` and value-initialization, for instance `new (nothrow) char[N]()`.

SAT-32958

A bug was fixed in which `org.apache.jsp.tag.web.checklistGraph_tag` could not be resolved to a type.

SAT-33006

Fixed a crash in the `SENSITIVE_DATA_LEAK` checker, which was caused by incorrect handling of array expressions.

SAT-33075

Fixed some serious performance issues affecting analysis of C++ codebases with compliance standards.

SAT-33163

Fixed a recoverable error with message `Cannot get TULinks from TU with no ASTs when analyzing with CERT coding standards`.

SAT-33399

Fixed a source of false positive reports `RISKY_CRYPTO` that complained about an insecure block mode when not using a block cipher.

SAT-33472

Fixed a recoverable failure in the `INFINITE_LOOP` checker when analyzing Visual Basic code under certain circumstances.

SATW-2804

Fixed a false positive in MISRA C++-2008 Rule 7-1-1 where a variable was modified in `try-catch` blocks.

SATW-2950

Fixed a false positive of MISRA C 2012 Rule 13.1 about taking the address of a volatile variable in initializer-list.

SATW-3048

Fixed a false positive in AUTOSAR C++14 A16-0-1 about conditional file inclusion.

SATW-3065

Fixed a false positive of MISRA C-2012 Directive 4.3 where an `assembly` statement was mixed with a variable declaration or a `return` statement.

SATW-3142

Fixed a false positive of CERT ERR30-C about not resetting `errno` for out-of-band error indicator returning functions.

SATW-3254

Fixed a false positive of MISRA C++-2008 Rule 0-1-6 about reassignment of a variable.

SATW-3263

Fixed a false positive of CERT STR34-C where a `char` character is compared against another `char` character.

SATW-3267

Fixed a false positive of CERT INT36-C where a variable of `uintptr_t` type was cast to a pointer.

SATW-3289

Fixed false positives of MISRA C-2012 Rule 10.1 and MISRA C-2012 Rule 10.4 where `false` and `0` were in the same plain macro.

SATW-3318

Fixed a false positive of AUTOSAR C++14 A4-7-1 about increasing size without data loss.

SATW-3360

Fixed a false positive of MISRA C++-2008 Rule 12-8-1 when other rules were enabled at the same time.

SATW-3400

Fixed false positives of AUTOSAR C++14 M9-3-3 and AUTOSAR C++14 A8-4-5 when `lambda` function was used.

SATW-3401

Fixed a false positive of AUTOSAR C++14 A7-1-5 where member functions used `auto` specifier and trailing return type syntax in class template.

SATW-3407

Fixed a false positive of AUTOSAR C++14 A3-1-1 related to member initializations in class definitions.

SATW-3412

Fixed a false positive of MISRA C-2012 Rule 10.3 where Boolean `true` literal was used in `struct` initializers.

SATW-3414

Fixed a false positive of MISRA C++-2008 Rule 0-1-6 when a value is used in arguments to a `new` expression.

SATW-3416

Fixed a false positive of MISRA C-2012 Rule 10.3 where there were casts in macros to define `bool` literals.

SATW-3436

Fixed a false positive of MISRA C++-2008 Rule 12-8-2 about compiler-generated copy assignment operators.

SATW-3447

Fixed a false positive of CERT INT31-C about left shift operation and bitwise operation.

SATW-3455

Fixed false positive of CERT ERR59-CPP when the library function declaration explicitly indicates that it will not throw an exception.

SATW-3468

Fixed a false positive of MISRA C-2012 Rule 10.1 where an `enum` constant was used as an operand of `operator []`.

SATW-3478

Fixed a false positive of AUTOSAR C++14 A7-1-2 about declaration of a non-trivial destructible `const` object.

SATW-3479

Fixed a false positive of AUTOSAR C++14 A7-1-5 where a lambda function without `auto` specifier was used as a function parameter.

SATW-3499

Fixed a false negative of CERT FIO45-C about value tracking on the string literal.

SATW-3500

Fixed a false positive of MISRA C-2012 Rule 21.18 about appropriate argument value of `size_t` type.

SATW-3516

Fixed a false positive in CERT INT30-C about arithmetic operations on `struct` fields.

SATW-3531

Fixed a false positive of AUTOSAR C++14 A12-1-1 about delegating constructors.

SATW-3539

Fixed a false positive of AUTOSAR C++14 A4-5-1 where an explicit enumerator equality expression was used as an operand of logical and operator.

SATW-3553

Fixed False Positive in MISRA C++-2008 Rule 3-1-1 related to member initializations in class definitions.

24.3.1.4. 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-7224

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

24.3.2. Coverity Commands 2020.03

24.3.2.1. Deprecated products and features

COVP-2205

Support for FreeBSD 11.2 has been deprecated in this release.

24.3.2.2. New or changed features

- A new command, `cov-upload-findings`, has been added to Coverity Analysis. The `cov-upload-findings` command uploads Coverity Analysis defect reports to a Compliance Solution Findings Manager. Refer to the *2020.03 Command Reference* for more information about this command. (COVDOCS-35)

Coverity Compliance Solution is currently part of the Compliance Solution Beta program and is available only to program participants. For more information about the Compliance Solution Beta program, see the section "Compliance Solution". (COVDOCS-35)

- The `cov-commit-defects` command has been updated to support streams with attached priority filters. Refer to the *Command Reference* for more information. (COVDOCS-55)

Coverity Compliance Solution (and priority filtering) is currently part of the Compliance Solution Beta program and is available only to program participants. For more information about the Compliance Solution Beta program, see the section "Compliance Solution". (COVDOCS-55)

- Added support for FreeBSD 12.1. (COVP-2203)

24.3.2.3. Bug fixes

CMPCPP-9405

`cov-run-desktop/cov-manage-emit` recompile assertion Selected record-only TUs again after recompiling them has been eliminated.

CMPCPP-9617

Fixed an issue where `cov-configure` failed to generate configuration for the MetaWare `ccac` compiler.

CMPG-3205

`cov-run-desktop` and `cov-manage-emit` recompile will now update text files.

COVDOCS-62

Documentation for `cov-analyze` is amended to improve translation as requested.

SAT-31282

Added a new option, `--cxx-container-type-regex` to the `cov-analysis` command; this option allows specifying C++ container types for all checkers that look for them.

SAT-32361

Fix a bug so that `cov-format-error` can correctly set the `language` field in the generated json output.

SAT-32900

To reduce analysis time where there are multiple verbatim copies of the same function, `cov-analyze` processes just one of these, chosen heuristically. There was a flaw in the heuristic algorithm, which allowed it to enter an infinite loop in extremely rare cases. This flaw has been corrected.

24.3.2.4. 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.

24.3.3. Coverity Compilers and Capture 2020.03

24.3.3.1. End-of-life products

CMPG-3184

Support for the target StarCore DSP version 3.0 and StarCore SDMA version 3.0 of Freescale Codewarrior compilers is dropped as of 2020.03.

CMPG-3216

Support for Apple JDK is dropped as of 2020.03.

CMPG-3217

Dropped support for Visual Studio 2010 and 2012.

COVP-2208

Build capture of Apple's JDK 1.6 has been EOL'd.

24.3.3.2. Deprecated products and features

COVP-2211

Support for .NET Core 3.0. has been deprecated in this release.

COVP-2213

Support for Swift 5.0.x has been deprecated as of 2020.03 and will be removed in a future release.

24.3.3.3. New or changed features

- Support for Pre-Compiled Headers (PCH) has been improved for gcc and Clang compilers; in fast desktop scenarios, PCH dependencies will now be handled automatically. (CMPCPP-8302)
- Added support for the Green Hills Optimizing C and C++/EC++ V850 2018.5.5 compiler. (CMPCPP-8715)
- Added support for the IAR Renesas RX v4.12 compiler. (CMPCPP-9070)
- `cov-emit` now tolerates C99-designated initializers for non-POD subobjects in C++. (CMPCPP-9300)
- Added support for MPLAB xc32-gcc 2.20 compiler on Linux. (CMPCPP-9409)
- Added support for the Green Hills Optimizing C and C++/EC++ RH850 2019.5.5 compiler. (CMPCPP-9654)
- Added support for the Apple Clang 11 (Xcode 11) compiler. (CMPCPP-9692)
- Added support for Kotlin 1.3-1.3.61. (CMPFG-118)
- Support for Pre-Compiled Headers (PCH) has been improved for gcc and Clang compilers; in fast desktop scenarios, PCH dependencies will now be handled automatically. (CMPG-2482)
- `yield` statements are now supported in Java 13 preview `switch` expressions, and support for `break` statements in Java 12 preview `switch` expressions is removed. (CMPJ-1234)
- HTML source files that do not contain or include JavaScript code are now stored in the emit, rather than being ignored. (CMPJS-742)

- Added support for Oracle JDK 13. (COVP-2195)
- Added support for OpenJDK 13. (COVP-2196)
- Added support for .NET Core 3.1. (COVP-2207)

24.3.3.4. Bug fixes

CMPCPP-8483

Corrected translation of initialization of a GNU vector with a brace-enclosed list containing a single GNU vector.

CMPCPP-9054

`cov-emit` assertion failure at "edg/src/templates.c", line 1620, has been eliminated.

CMPCPP-9363

Fixed an issue where Coverity didn't recognize builtin prototype `_mm512_set_epi16` for the intel compiler on Linux.

CMPCPP-9415

Diagnostics in system headers are normally suppressed. In preprocessed and PCH files this was not happening. This has been corrected.

CMPCPP-9422

Fixed a problem with slow builds resulting from the processing of C++ Range Library.

CMPCPP-9466

The C++ dialect is now properly configured when compiling source as Objective-C++.

CMPCPP-9522

Fixed a bug where `cov-build` against icc 18.01 failed with error message `__gnuc_va_list` is undefined.

CMPCPP-9542

Fixed an issue where the option `--cygpath` to the `cov-emit` command was not being honored.

CMPCPP-9586

A `cov-emit` issue, `EXCEPTION_ACCESS_VIOLATION` was fixed.

CMPCPP-9597

A catastrophic compile error was fixed for the `cov-emit` command.

CMPCPP-9623

A number of bugs preventing Boost 1.68 from compiling on Solaris have been fixed.

CMPCPP-9695

Diagnostics in system headers are normally suppressed. In preprocessed and PCH files this was not happening. This has been corrected.

CMPCPP-9838

`Cov-emit` assertion default rescan info (`exprutil.c`, line 4749 in `get_expr_rescan_info`) is eliminated.

CMPJ-1219

An incremental build will now update the emit database based on the addition or modification of jar files in the classpath of a Java compilation, even if the source files being compiled have not changed.

CMPJ-1255

Fixed a `cov-run-desktop` error when emitting JSP files and some of them failed to compile.

CMPJS-419

HTML files imported from JavaScript code via the NgTemplate loader are now emitted properly.

CMPJS-767

Files with `.scss` extension linked from JavaScript code are emitted as text and no longer produce spurious errors in an attempt to parse the SCSS code as JavaScript.

CMPJS-770

JSON files with `.json` extension linked from JavaScript code are emitted as text and no longer produce spurious errors in an attempt to parse the JSON code as JavaScript.

SAT-31207

CSV output produced by Coverity analysis (`callgraph-metrics.csv`, `checked-return.csv`) can now be parsed by other tools

24.3.3.5. 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. * Ignore the `xdcmake.exe` process by adding `--capture-ignore xdcmake.exe` to your `cov-build` invocation.

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.

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.

CMPSCA-187

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

24.3.4. Coverity Dynamic Analysis 2020.03

24.3.4.1. End-of-life products

CMPG-3216

Support for Apple JDK is dropped as of 2020.03.

24.3.4.2. 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.

24.3.5. Coverity Test Advisor 2020.03

24.3.5.1. End-of-life products

COVP-2210

Support for Mercurial 3.1-3.2 has been EOL'd.

COVP-2215

Support for git 1.8-2.1 has been EOL'd.

24.3.5.2. Deprecated products and features

COVP-2219

Support for Team Foundation Server 2012 has been deprecated.

24.3.5.3. Bug fixes

SAT-31473

Documentation was amended to replace illegible images in Korean with legible images in English.

24.3.5.4. Known issues and solutions

TADE-2033

The use of "--cs-coverage opencover" with Test Advisor may 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. This can be worked around by manually registering the OpenCover profiler DLLs and passing "--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 on this, see the documentation of cov-build's "--cs-no-register-profiler" switch in the 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.

24.3.6. Coverity Wizard 2020.03

24.3.6.1. 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.

24.4. Coverity Desktop 2020.03

This section provides release notes for Coverity Desktop components.

24.4.1. Coverity Desktop for Android Studio 2020.03

24.4.1.1. Known issues and solutions

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.

24.4.2. Coverity Desktop for Eclipse 2020.03

24.4.2.1. New or changed features

- Added support for Eclipse 2019.12 (4.14) (PRD-12040)
- Added support for WindRiver 4.0. (PRD-12073)

24.4.2.2. Bug fixes

PRD-11859

Eclipse now expands `$(version)` in `coverity.conf` correctly.

24.4.2.3. 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/`.

24.4.3. Coverity Desktop for IntelliJ IDEA 2020.03

24.4.3.1. New or changed features

- Added support for IntelliJ 2019.3 (PRD-11985)
- Added support for PyCharm 2019.3 (PRD-11987)
- Added support for PhpStorm 2019.3 (PRD-11988)
- Added support for WebStorm 2019.3 (PRD-11989)
- Added support for RubyMine 2019.3 (PRD-11990)

24.4.3.2. 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.

24.5. Coverity Documentation 2020.03

This section provides release notes for Coverity Documentation components.

24.5.1. Coverity Documentation 2020.03

24.5.1.1. Bug fixes

COVDOCS-46

Reinserted three images that were missing from the Policy Manager section of the Coverity Platform User and Administration Guide.

IM-24577

Broken links have been fixed in the *Upgrade Guide*.

IM-24595

Table 3.3.4. (Exported component map JSON elements) of the Coverity Platform User and Administrator Guide previously indicated, incorrectly, that the `<component>` field could contain a `<description>` field. This is not true, and the row for that `<description>` field has been removed from the table.

Chapter 25. Coverity 2019.12-8 Release Notes

Table of Contents

25.1. Coverity Analysis Coverity Commands	97
---	----

25.1. Coverity Analysis Coverity Commands

SAT-35253

An internal limitation on the size of a command line was removed. Missing function fields were restored, aiding in tracking defects across source code changes. A ticker was added to show progress while converting Fortran defects.

Chapter 26. Coverity 2019.12-7 Release Notes

Table of Contents

26.1. Coverity Platform bug fixes	98
---	----

26.1. Coverity Platform bug fixes

CMPCPP-10325

Cannot have an in-class initializer and incomplete type errors eliminated.

CMPCPP-9042

Cannot have an in-class initializer errors eliminated.

Chapter 27. Coverity 2019.12-6 Release Notes

Table of Contents

27.1. Coverity Platform bug fixes 99

27.1. Coverity Platform bug fixes

IM-24913

Improved an SQL query which is used as part of the commit action. In some cases the execution time of the query was in the minutes order of magnitude, execution of the the improved query takes milliseconds.

Chapter 28. Coverity 2019.12-5 Release Notes

Table of Contents

28.1. Coverity Platform bug fixes	100
---	-----

28.1. Coverity Platform bug fixes

IM-24788

Fixed problems that prevented the `issue` type field to populate for JIRA Cloud.

Chapter 29. Coverity 2019.12-4 Release Notes

Table of Contents

29.1. Coverity Analysis bug fixes	101
---	-----

29.1. Coverity Analysis bug fixes

CMPCPP-9739

Assertion "Tried to emit a multi-part diagnostic with no source location" has been eliminated.

SAT-33617

Fixed a bug so that `cov-polaris-export-defects` can correctly export the missing localized defect checker fields.

Chapter 30. Coverity 2019.12-3 Release Notes

Table of Contents

30.1. Coverity Analysis bug fixes	102
30.2. Coverity Platform bug fixes	102

30.1. Coverity Analysis bug fixes

CMPCPP-9739

Assertion "Tried to emit a multi-part diagnostic with no source location" has been eliminated.

30.2. Coverity Platform bug fixes

IM-24644

A bug was fixed where the Coordinator was running out of memory after a 2019.12 upgrade.

Chapter 31. Coverity 2019.12-2 Release Notes

Table of Contents

31.1. Coverity Analysis bug fixes	103
---	-----

31.1. Coverity Analysis bug fixes

CMPSWIFT-352

Fixed a Swift compiler issue with `bool` literals causing the compiler to abort.

CMPSWIFT-353

Fixed an issue where Swift compiler was omitting symbols without storage.

CMPSWIFT-354

Fixed a Swift compiler issue where an internal function being called was causing an assertion failure due to wrong return type.

Chapter 32. Coverity 2019.12-1 Release Notes

Table of Contents

32.1. Coverity Analysis bug fixes	104
---	-----

32.1. Coverity Analysis bug fixes

CMPCPP-9713

Diagnostics in system headers are normally suppressed. In preprocessed and PCH files this was not happening. This has been corrected.

Chapter 33. Coverity 2019.12 Release Notes

Table of Contents

33.1. Important information for 2019.12	105
33.2. Coverity Platform 2019.12	106
33.3. Coverity Analysis 2019.12	110
33.4. Coverity Desktop 2019.12	132
33.5. Coverity Report Generators 2019.12	136
33.6. Coverity Documentation 2019.12	137

33.1. Important information for 2019.12

Support for this version of Coverity will be discontinued 18 months after the 2019.12 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.

Due to a change in our bug tracking system, items might now be identified by two bug numbers:

- One specifying the identity of the bug in our **old** bug tracking system, formatted like this: XXXXXX. (For example, 374568.)
- One specifying the identity of the bug in our **new** bug tracking system, formatted like this: CODE-XXXXX. (For example, IM-22788.)

33.1.1. Deprecated and End-of-Life (EOL) Products in Coverity 2019.12

Support for the following products, features, platforms, and third-party tools is classified as deprecated or end-of-life as of the Coverity 2019.12 release.

33.1.1.1. Deprecated Products

Support for the following products and features is deprecated as of the Coverity 2019.12 release.

Table 33.1. Deprecated products

Product	See also...
target StarCore DSP version 3.0 and StarCore SDMA version 3.0 of Freescale Codewarrior	Supported Compilers: Coverity Analysis for C/C++
IntelliJ 2017.1 (Includes IntelliJ-based IDEs CLion, PhpStorm, PyCharm, RubyMine, WebStorm)	Supported Platforms for Coverity Analysis
Perforce 2016.1	Coverity Test Advisor Supported SCM Systems

Product	See also...
SVN 1.9	Coverity Test Advisor Supported SCM Systems
Version 6, 7, and 8 of the Web Services API	Supported frameworks for Coverity Analysis

33.1.1.2. End-of-Life Products

Support for the following products and features is dropped in the Coverity 2019.12 release.

Table 33.2. End-of-Life Products

Product	See also...
The <code>--dot-coverity-location</code> argument to the <code>cov-capture</code> command is no longer supported.	The <code>.coverity</code> directory resides directly under the intermediate directory and is renamed <code>cov-capture</code> .
Eclipse 4.6	Coverity Coverity Desktop for Eclipse on supported platforms
MacOS 10.12	This affects support for Coverity Analysis, all compilers, Coverity Desktop plugins, Extend SDK, FLEXnet licensing, Coverity Wizard, and Architecture Analysis.
Microsoft Embedded C++ 4.0	Supported Compilers: Coverity Analysis for C/C++
OpenJDK 12	Supported Platforms for Coverity Analysis
Oracle JDK 12	Supported Platforms for Coverity Analysis
Perforce 2015.2	Coverity Test Advisor Supported SCM Systems
Swift 4.2	Supported Compilers: Coverity Analysis for Swift
Windows 7 for Cov_Wizard, and the Eclipse, IntelliJ, Android plugins.	Supported Platforms for Coverity Analysis

33.2. Coverity Platform 2019.12

This section provides release notes for Coverity Platform components.

33.2.1. Coverity Connect 2019.12

Coverity Connect is a component of the Coverity Platform installation package.

33.2.1.1. Important Coverity Connect Information

The following have been deprecated for Coverity Connect this release:

- Version 6 of the Web Services API. (IM-24428)

- Version 7 of the Web Services API. (COVDOCS-11)
- Version 8 of the Web Services API. (COVDOCS-13)

33.2.1.1.1. New and changed features

The following new and changed features have been added for Coverity Connect this release:

- A system administrator can use the new `cov-archive` command to export a stream, import a stream, or get information about an archive. (IM-23500)

NOTE: In some cases, importing a stream might take hours or days. This results from complex data and usually happens for archives larger than 200 MB. Performance might improve in a future release.

33.2.1.1.2. Bug fixes

The following bugs are fixed for Coverity Connect:

IM-15755

Added the ability to provide event-driven triage notification.

IM-20381

Coverity Connect URL construction: added filtering by snapshots.

IM-22108

Fixed an issue that prevented `cov-manage-im` from showing an error description when used with `--ssl` option

IM-23925

Fixed an issue with Kerberos authentication: access would fail when a database was restored to a new host system.

IM-23934

Some anomalies in the way Coverity Connect displayed certain Coverity IDs (CIDs) have been fixed.

IM-23939

We fixed an issue that prevented Coverity Connect from loading and displaying triage information for certain defects .

IM-24052

In "Creating, copying, and deleting projects and streams", the *User and Administrator Guide* now correctly describes how Coverity Connect treats streams that have been designated `Outdated`.

IM-24122

Executing `cov-admin-db upgrade-schema` with an external database no longer fails.

IM-24135

A bug was fixed whereby deleted users still received email notifications.

IM-24240

Added information about supported Linux versions in table 7.1.1 of the *Coverity Installation Guide*.

IM-24241

Updated doc to include information about a `url` variable for static fields in Table 3.1.1 in the *Coverity Platform User and Administrator Guide*.

IM-24255

Fixed an issue in which the user wanted to see defects that are both older than a given date and newer than a given date.

IM-24287

Fixed an issue causing Coverity Connect to connect to the analysis update server even when the updates were disabled.

INS-2716

Fixed an issue with `cov-analysis-win64-2019.03.exe` in a Windows environment. When attempting to uninstall it, the process returned an error with the message that GC overhead limit was exceeded.

33.2.1.1.3. Known issues and solutions

Coverity Connect has the following known issues:

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.

CPU-17, 80045

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, 82579

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, 67748

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-17660, 75263

An error occurs when a custom role is created using a multi-word rolename that is the same as a built-in rolename, even if there are case differences between the two rolenames.

IM-17701, 75559

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

IM-18707, 82643

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, 82648

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, 84453

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, 89897

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, 89946

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-23994

Internet Explorer 11 fails on operations using file upload.

INS-1274, 63454

Although the upgrade doc 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, fall back to backup-and-restore upgrade.

INS-1477, 73401

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 by specifying a max heap setting for `cov-im-daemon`.

INS-2133, 112939

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, 118662

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.
```

33.2.2. Coverity Policy Manager 2019.12

Coverity Policy Manager is a component of the Coverity Platform installation package.

33.2.2.1. Important Coverity Policy Manager information for 2019.12

There are no deprecated or EOL items for 2019.12:

33.2.2.2. Bug fixes in 2019.12

Coverity Policy Manager™ has no fixed bugs in 2019.12:

33.3. Coverity Analysis 2019.12

This section provides updates about Coverity Analysis components.

33.3.1. Important Coverity Analysis information

There have been several deprecations and EOLs this release:

- Undeprecated (re-added) Coverity Analysis platform support for glibc 2.12–2.13.x. (CMPG-3157)
- All support for macOS 10.12 is dropped as of 2019.12. (COVP-2103)

33.3.1.1. New and changed features

Coverity Analysis has the following new and changed features:

COVP-2150

Support for NetBSD 8.1 has been added.

COVP-2148

Support for FreeBSD 11.3 has been added.

33.3.1.2. Bug fixes

There are no bug fixes for Coverity Analysis in 2019.12.

33.3.1.3. Known Issues

There are no known issues for Coverity Analysis in 2019.12.

33.3.2. Coverity Analysis checkers and user directives in 2019.12

The following sections describe new and updated features, bug fixes, and known issues for Coverity checkers and associated elements.

33.3.2.1. New and updated checkers and directives

- Added support for the M rules (Mx-x-x, where x is a literal number, for example M0-1-1) in AUTOSAR C ++14 compliance standard for Clang-based compilers. (SAT-32261)

The following table lists new checkers and the languages they support.

Checker	Languages
CONFIG.COOKIE_SIGNING_DISABLED	Java, JavaScript, TypeScript
CONFIG.UNSAFE_SESSION_TIMEOUT	Java, JavaScript, TypeScript
CORS_MISCONFIGURATION	JavaScript, TypeScript
CORS_MISCONFIGURATION_AUDIT	JavaScript, TypeScript
DISTRUSTED_DATA_DESERIALIZATION	Go
EXPRESS_SESSION_UNSAFE_MEMORYSTORE	JavaScript, TypeScript
EXPRESS_WINSTON_SENSITIVE_LOGGING	JavaScript, TypeScript
REACT_DANGEROUS_INNERHTML	JavaScript, TypeScript
REVERSE_TABNABBING	JavaScript, Ruby, TypeScript
UNLESS_CASE_SENSITIVE_ROUTE_MATCHING	JavaScript, TypeScript

The following table documents added language support for existing checkers.

Checkers	Languages
ANGULAR_EXPRESSION_INJECTION	TypeScript
DEADCODE	Go
DIVIDE_BY_ZERO	VB.NET
HARDCODED_CREDENTIALS	Go
HEADER_INJECTION	Go
INFINITE_LOOP	Go, VB.NET
INSECURE_COOKIE	JavaScript, TypeScript
INSUFFICIENT_LOGGING	Go
LOCK_EVASION	VB.NET
NOSQL_QUERY_INJECTION	Go
NULL_RETURNS	VB.NET
OPEN_REDIRECT	Go
OS_CMD_INJECTION	Go

Checkers	Languages
PATH_MANIPULATION	Go
RISKY_CRYPTO	Go
SENSITIVE_DATA_LEAK	Go
SQLI	Go
TAINTED_ENVIRONMENT_WITH_EXECUTION	Go
TEMPLATE_INJECTION	Go
URL_MANIPULATION	Go
XML_EXTERNAL_ENTITY	Go
XSS	Go

New and changed checkers

SAT-3972

The new option `tainting_downcasts` has been added to the `TAINTED_SCALAR` checker. If this option is set to `true`, the checker will treat casts from raw data (like `char *` or `void*` type), to certain struct types (for example, those that look like network packets), as a source of tainted data.

SAT-4411

The `LOOP_BOUND` sink type for `TAINTED_SCALAR` has been replaced by `LOOP_BOUND_LOWER` and `LOOP_BOUND_UPPER`.

SAT-5604

The `TAINTED_SCALAR` checker now reports on division by an untrusted scalar, as in such a case an attacker could potentially create a division by zero.

SAT-6405, SAT-31520

The `USER_POINTER` checker is now enabled when using the `--security` option to the `cov-analyze` command.

SAT-27601

Setting `aggressiveness-level` to `high` now implies the `--distrust-all` option for the following C, C++ checkers: `FORMAT_STRING_INJECTION`, `OS_CMD_INJECTION`, `PATH_MANIPULATION`, `SQLI`, `URL_MANIPULATION`, `XPATH_INJECTION`. (C, C++)

SAT-28649

The `NULL_RETURNS` checker will no longer report defects when the return value of an unimplemented function is never checked for `null`, even if statistical thresholds and biases, along with the `allow_unimpl` option, indicate it should be considered to return `null`. A new option, `allow_unimpl_and_unchecked`, will revert to the previous behavior.

SAT-28830

Ruby security analysis now supports HAML version 5.

SAT-29081

The `DEADCODE` checker now supports Go.

SAT-29085

The `INFINITE_LOOP` checker now supports the Go language

SAT-29130

The `SQLI` checker can report untrusted data being passed into `MyBatis` queries with unescaped string substitution. This preview feature requires passing the option `--enable-mybatis-sqli` to the `cov-analyze` command. There might be a noticeable increase in analysis time if you enable this checker.

SAT-29921

The new `REVERSE_TABNABBING` checker finds cases where a link is dynamically generated and is set to open a new window by virtue of its `target` attribute being set to `_blank`.

SAT-30341

The `SENSITIVE_DATA_LEAK` checker now supports Go.

SAT-30342

The `RISKY_CRYPTO` checker now supports Go.

SAT-30346

The `XML_EXTERNAL_ENTITY` checker now supports Go.

SAT-30347

The `INSUFFICIENT_LOGGING` checker reports a defect in code that handles a security event or error condition but does not properly log the event. It now supports Go.

SAT-30350

The `SQLI` checker now supports Go.

SAT-30351

The `HEADER_INJECTION` checker now supports Go.

SAT-30359

The new `DISTRUSTED_DATA_DESERIALIZATION` Go checker reports an issue any time distrusted data is passed into a deserialization API.

SAT-30485

Added analysis support for new Java 12 support, in particular the new preview `switch` statement syntax and `switch` expression.

SAT-30550

The `typedefType` pattern in the CodeXM C/C++ library has had some of its fields renamed to match other patterns (such as `classType`): `alias` is now `mangledName`; `id` is now `identifier`. Also, `scopeList` was added.

SAT-30667

The `classDefinition` types in CodeXM C/C++, Java and C# libraries now expose accessors to find base classes: `findBaseClass` and `findMatchingBaseClass`.

SAT-30773

The `DIVIDE_BY_ZERO` checker now supports VB.NET.

SAT-30774

The `INFINITE_LOOP` checker now supports VB.NET.

SAT-30775

The `NULL_RETURNS` checker now supports VB.NET.

SAT-30776

The `LOCK_EVASION` checker now supports VB.NET.

SAT-30995

The new `EXPRESS_SESSION_UNSAFE_MEMORYSTORE` checker flags `express-session` instances where the `store` property is set to `(express-session).MemoryStore` in configuration or omitted (defaults to `(express-session).MemoryStore`).

SAT-31148

The new `CONFIG.COOKIE_SIGNING_DISABLED` checker flags `cookie-session` instances where the `signed` property is set to `false`, disabling cookie signing.

SAT-31322

The `INSECURE_COOKIE` checker now supports JavaScript and TypeScript.

SAT-31394

The new `UNLESS_CASE_SENSITIVE_ROUTE_MATCHING` checker finds cases where the `unless` function is called in an `Express` application with the `path` parameter that includes a case-sensitive negative regular expression.

SAT-31432

The new `CONFIG.UNSAFE_SESSION_TIMEOUT` checker finds issues with Java, JavaScript, and TypeScript code in which sessions are unlimited or are timing out after an excessive amount of time.

SAT-31459

The new `REACT_DANGEROUS_INNERHTML` checker finds cases where the `dangerouslySetInnerHTML` attribute of a `React` element is set.

SAT-31498

Added models for `boost` circular buffer.

SAT-31535

Added C/C++ models for the Xerces API to detect `PATH_MANIPULATION` defects.

SAT-31549

Added models for `boost` filesystem library.

SAT-31561

The `ANGULAR_EXPRESSION_INJECTION` checker now supports TypeScript.

SAT-31608

Added support for the OpenSSL low-level cryptographic APIs to `RISKY_CRYPTO`.

SAT-31619

Added models for `Boost` .Heap library.

SAT-31620

Added C/C++ models for the `Boost .beast` API to detect `PATH_MANIPULATION` defects.

SAT-31826

Added models for C/C++ API of Miniz library.

SAT-31834

The checker `REACT_DYNAMIC_URL_INSECURE_TARGET` has been renamed `REVERSE_TABNABBING`.

SAT-31897

Added models for the `Boost Iostreams` library.

SAT-31898

Added models for the `Boost Asio` library.

SAT-31944

Added models for `libfetch` library.

SAT-32091

The `NOSQL_QUERY_INJECTION` checker now supports Go.

SAT-32092

The `PATH_MANIPULATION` checker now supports Go.

SAT32093

The `TAINTED_ENVIRONMENT_WITH_EXECUTION` checker now supports Go.

SAT-32094

The `TEMPLATE_INJECTION` checker now supports Go.

SAT-32095

The `URL_MANIPULATION` checker now supports Go.

SAT-32096

The `HARDCODED_CREDENTIALS` checker now supports Go.

SAT-32097

The `OPEN_REDIRECT` checker now supports Go.

SAT-32098

The `OS_CMD_INJECTION` checker now supports Go.

SAT-32099

The `XSS` checker now supports Go.

SAT-32115, SAT-32116

The new `CORS_MISCONFIGURATION` checker finds insecure configurations of the Cross Origin Resource Sharing (CORS) policy, which uses additional HTTP headers to allow an application running at one origin to access selected resources from a different origin.

The new `CORS_MISCONFIGURATION_AUDIT` checker finds insecure configurations of the Cross Origin Resource Sharing (CORS) policy, which uses additional HTTP headers to allow an application running at one origin to access selected resources from a different origin. The checker reports different types of problematic issues in the CORS configuration that need to be audited (compared to `CORS_MISCONFUIGRATION`) based on the language of the application and the frameworks or libraries used.

SAT-32128

The new `EXPRESS_WINSTON_SENSITIVE_LOGGING` checker finds several cases where sensitive data is automatically logged by the middleware component of `express-winston`.

SAT-32261

Added support for the M rules (Mx-x-x, where x is literal number, for example M0-1-1) in AUTOSAR C++14 compliance standard for Clang-based compilers.

SAT-32167

Added support for the MISRA C++-2008 compliance standard for Clang-based compilers.

SAT-32422

Language coverage for options for `SQLI` has been updated.

SAT-32423

Language coverage for options for `SCRIPT_CODE_INJECTION` has been updated.

SAT-32424, SAT32436, SAT-32375,

Language coverage for options for `PATH_MANIPULATION` has been updated.

33.3.2.2. Bug fixes related to checkers in 2019.12

The following checker-related bugs have been fixed for this release.

CMPCPP-9483

Fixed a false negative of MISRA C-2012 Rule 3.2 about CRLF in the end of comment line.

SAT-6507

Fixed an `INTEGER_OVERFLOW` false positive, related to the `getenv` family of functions.

SAT-6840, SAT-6843, SAT-6844

Improved detection of tainted scalars used in loop bounds by the `TAINTED_SCALAR` checker.

SAT-12270, SAT-18853, SAT-25786

Fixed a class of `TAINTED_SCALAR` false positives, where the bounds of the scalar are checked by a called function.

SAT-14636

Fixed a source of `USER_POINTER` false positives where calls to `copy_from_user` were incorrectly reported when correctly passing a user pointer as `from` argument

SAT-18399, SAT-30034

Fixed complexity metric calculation for declaration statements containing conditional expressions.

SAT-25282, SAT-30154

The `NULL_RETURNS` checker will no longer report defects when the return value of an unimplemented function is never checked for null, even if statistical thresholds and biases, along with the `allow_unimpl` option, indicate it should be considered to return null. A new option, `allow_unimpl_and_unchecked`, will revert to the previous behavior.

SAT-25786, SAT-18853, SAT-12270

Fixed a class of `TAINTED_SCALAR` false positives, where the bounds of the scalar are checked by a called function.

SAT-26386

Many Coverity checkers now support the Go language. See Chapter 3 of the *Coverity Checkers Reference Guide* for a complete list.

SAT-26964

Fixed a `TAINTED_SCALAR` false positive where a scalar was sanitized by an equality check.

SAT-29065

Fixed a source of false positives when using `org.junit.Assert.assertThat()`.

SAT-29167

Improved the `UNINIT` checker to report defects on arguments to `printf`-style functions.

SAT-29503

Fixed a source of `OVERRUN` false positives with functions accessing buffers in blocks.

SAT-29793

Improved support for the Go `select` statement.

SAT-29827

Improved handling of function literals in the Go language.

SAT-29852

Changed wording in the `HIS` metrics report to more clearly indicate the number of `HIS` violations.

SAT-30248

The `BUFFER_SIZE` checker no longer reports on calls to `memset`. The reports were confusing and `OVERRUN` reports the same defects in a clearer way.

SAT-30652

Fixed a source of `FORWARD_NULL` false positives when calling the `getcwd` function on platforms where it allows `NULL` for the first parameter.

SAT-30845

Fixed a source of `OVERRUN` false positives when reading from the address of a function

SAT-30846

Fixed a `TAINTED_SCALAR` false positive that involved an unsigned character used as an offset for a pointer to a 256 byte block of memory.

SAT-31025

Fixed `RISKY_CRYPTO` false negatives when using SHA and SHA2 related APIs in OpenSSL library.

SAT-31069

Fixed a source of `OVERRUN` false positives when calling some functions that iterate over a NUL-terminated string and a fixed size buffer, such as custom implementations of `strncpy`.

SAT-31074

Broken links in the *Coverity Checker Reference* guide have now been fixed.

SAT-31276

Broken links to CodeXM documentation have now been fixed.

SAT-31309

Fixed a problem where `PROPERTY_MIXUP` checker was delaying completion of `cov-analyze`.

SAT-31369

Fixed a bug that affected the default trust settings for taint sources in the `FORMAT_STRING_INJECTION` checker.

SAT-31374

Updated documentation to specify the correct way of enabling `INTEGER_OVERFLOW` checker.

SAT-31409, SAT-31235

An appendix describing new checkers and added language support for checkers in each release has been restored to the *Checker Reference Guide*.

SAT-31499

Fixed broken links related to CodeXM documentation.

SAT-31547

The sample code in the "Quick start tutorial" section of *Learning to Write Code XM Checkers* has been corrected.

SAT-31601

Bouncy castle is now detected by the `RISKY_CRYPTO` checker if custom functions from Bouncy Castle are used.

SAT-31700

Documentation for `STACK_USE` checker has been updated with an example that recognizes the default values of `max_total_use_bytes` and `max_single_base_use_bytes`.

SAT-31701

Fixed an analysis crash with message "Cannot call toString on this lock name" in particular with Android `GuardedBy` annotations.

SAT-31757

Added support for `std::shared_mutex`; this was affecting `LOCK` checker and also other concurrency checkers: `MISSING_LOCK`, `LOCK_INVERSION`.

SAT-31857

Fixed a `BUFFER_SIZE` false positive where the defect would claim that an array had 0 elements and was overrun in a call to `strncat`.

SAT-31891

Fixed a false positive for the `UNINIT` checker involving nested loops.

ST-31951

Fixed a false positive of AUTOSAR C++-14 M3-4-1 with range-based `for` loop.

SAT-32044

Fixed a source of false positives when using the `NULL_RETURNS null_fields_config` option.

SAT-32049

Fixed a `cov-analyze` crash due to handling the `class_like_print_writer_for_servlet_output` directive.

SAT-32195

Fixed a false positive of MISRA C-2012 Rule 8.7 where `static const` was incorrectly claimed as external linkage.

SAT-32270

Fixed a source of `DELETE_ARRAY` false positives when using `new(void *)`.

SAT-32303

The checker option `no_dead_default` for the `DEADCODE` checker is now documented.

SAT-32322

Fixed a problem that could cause analysis or commit crashes when source files used some unsupported character encodings, such as `x-IBM33722`. Now the files are interpreted as ASCII instead.

SAT-32591

Fixed a bug where `cov-analyze` Javascript security checkers were reporting defects from Javascript minified code in the non-minified code that called it, when the `--report-in-minified-js` option was not specified.

SAT-32655

Fixed a recoverable error when the `SENSITIVE_DATA_LEAK` checker is run with `--enable-audit-mode`.

SATW-2970

Improved the implementation of CERT ENV30-C to not report defects where the referenced objects were not modified.

SATW-3046

Fixed a false positive of MISRA C++-2008 Rule 5-2-8 when using `NULL` as null-pointer-constant on 64-bits platform.

SATW-3058

Fixed a false positive of CERT EXP34-C about unimplemented functions that never return null.

SATW-3099

Fixed a false positive of CERT INT30-C where a variable of unsigned type was subtracted by `UINT64_MAX`.

SATW-3110

Fixed a false positive of CERT ERR33-C about unrecognized `errno` macro.

SATW-3112

Fixed a false positive of CERT INT31-C where `smaller` unsigned type was cast to `wider` unsigned type after integer promotion.

SATW-3113

Fixed false positives of MISRA C-2012 Rule 9.3 where an array's initializer only consisted of designated initializers.

SATW-3116

Fixed a false positive of CERT STR34-C when using `char` type for non-character data.

SATW-3118

Fixed a false positive of CERT ARR37-C when accessing array member of a struct.

SATW-3123

Fixed a false positive of MISRA C-2012 Rule 8.7 where `static const` was incorrectly claimed as external linkage.

SATW-3129

Fixed a false positive of MISRA C++-2008 Rule 6-4-6 where all the enumerators were listed in case labels.

SATW-3133

Fixed false positives of AUTOSAR C++14 A5-1-1 where boolean literals or `nullptr` were used.

SATW-3138

Fixed false positives of CERT INT34-C where integer literals were used as operands of `shift` operators.

SATW-3148

Fixed a false positive of CERT FIO32-C when opening a regular file.

SATW-3207

Fixed a false positive of CERT EXP37-C about the compatible type conversion.

SATW-3232

Fixed a false positive of CERT INT32-C related to subtraction overflow of signed 64-bits type.

SATW-3250

Fixed a false positive in MISRA C-2004 Rule 6.1 about an `enum` type when the option `--short_enums` was used.

SATW-3251

Fixed a false positive in MISRA C-2004 Rule 19.4 about the header inclusion guard macro.

SATW-3252

Fixed a false positive of MISRA C++-2008 Rule 6-6-1 where `goto` was placed in a statement expression.

SATW-3293

Fixed a false negative in MISRA C++-2008 Rule 17-0-5 on Windows platform.

33.3.2.3. Known issues and solutions

The following known issues and solutions have been identified for this release.

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-7224, 43971: XSS

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

SAT-17490, 84256:

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

33.3.3. Compiler configuration, Build capture, and Compiler Integration Toolkit (CIT) 2019.12

This section lists new features, bug fixes, and known issues related to Coverity-supported compilers (including configuration), and the Compiler Integration Toolkit (CIT).

33.3.3.1. Important information

There were several deprecations and EOLs for this release:

- The `--dot-coverity-location` argument to the `cov-capture` command is no longer supported. For better debugging experience, the `.coverity` directory resides directly under the intermediate directory and is renamed `cov-capture`. (BLC-899)
- Support for Microsoft Embedded C++ 4.0 compiler has been dropped. (CMPG-3084)
- Support for the target StarCore DSP version 3.0 and StarCore SDMA version 3.0 of Freescale Codewarrior compilers is deprecated and will be removed in a future release.(CMPG-3143)
- Coverity Analysis support for Swift 4.2 is dropped. (CMPG-3168)
- All support for macOS 10.12 is dropped. (COVP-2103)
- Support for OpenJDK 12 is dropped. (COVP-2179)

- Support for Oracle JDK 12 is dropped. (COVP-2181)

33.3.3.2. New and changed features

The following features have been added or changed for this release.

BLC-563

The `--delete-stale-tus` option has been added to the `cov-capture` command. This option automatically deletes translation units that are created from source files that were renamed or removed. This capability is off by default.

BLC-899

The `--dot-coverity-location` argument to the `cov-capture` command is no longer supported. For better debugging experience, the `.coverity` directory resides directly under the intermediate directory and is renamed `cov-capture`.

BLC-897

For the `cov-capture` command, the `--dir <idir>` argument is no longer optional; it is required.

CAP-756

If any JVM language has been configured with `cov-configure`, `cov-build` will now automatically disable the Gradle daemon. It is no longer necessary for the user to pass `--no-daemon`.

CAP-1213

If any JVM language has been configured with `cov-configure`, `cov-build` will now automatically disable the Gradle build cache. It is no longer necessary for the user to pass `--no-build-cache`.

CMPCPP-8824

Added support for Freescale Codewarrior StarCore C++ Compiler v10.9 on Windows.

CMPCPP-8836

Added support for the CrossWorks for MSP430 version 3.1.1 compiler on Windows.

CMPCPP-8924

Added support for the TI TMS320C6x version 8.3.1 compiler on Linux.

CMPG-3132

Added support for Go 1.12.

CMPG-3167

Added support for the Clang 9 compiler.

CMPJS-764

Added support for TypeScript 3.3.

CMPSWIFT-263

Added support for XCODE 10.2.1.

CMPSWIFT-268

Added support for Swift 5.0.

33.3.3.3. Bug fixes

The following bugs were fixed for compilers and the Compiler Integration Toolkit (CIT) for Coverity Analysis analysis in 2019.12:

BLC-886

Buildless capture fallback will now capture sources for all modules in multi-module Maven projects.

BLC-928

Fixed a bug where `cov-capture` could fail due not being able to retrieve dependencies that would have been built by the captured Maven project.

BLC-949

Running `cov-capture` within a terminal emulator or other job-managing context on Windows 7 no longer leads to memory exhaustion and crashing.

CAP-1506

Fixed an assertion failure in `libcapture` that could cause programs run under `cov-build` to fail on Unix. This failure would result in the following message being written to the `build` log.

```
capture-unix.c: assertion failed: s
```

CMPCSH-1167

Fixed a bug for catastrophic signal: C0000005 (EXCEPTION_ACCESS_VIOLATION) : tried to read from addr 0x0000000000000000.

CMPCPP-5390

A new error recovery mode has been implemented in `cov-emit` which addresses crashes caused by the back-end encountering an error node. This new mode can be enabled by adding `--no_error_recovery_walk` to the `cov-emit` command line.

CMPCPP-6844

Fixed a crash in `cov-emit` involving nested generic lambdas.

CMPCPP-7830

Fixed a spurious `PW.BAD_FRIEND_DECL` warning produced by `cov-emit` for some cases involving member classes of class templates.

CMPCPP-8379

Fixed an issue in `cov-emit` where the frontend sometimes failed to constant-evaluate a `constexpr` constructor call, producing spurious errors about accessing expired storage.

CMPCPP-8910

Fixed an issue that caused Clang's `-Wno-c++11-narrowing` option to be ignored resulting in unexpected translation failures.

CMPCPP-8951

Fixed an issue where `cov-configure` sanity test failed with `ecomarm`.

CMPCPP-9108

Fixed an issue where some encoding options were not handled for MSVC compiler.

CMPCPP-9150

Invalid `nontype` template argument error with `using` declaration has been eliminated.

CMPCPP-9155

Allow `__builtin_expect` in `constexpr` initializer.

CMPCPP-9160

Fixed an issue affecting gcc and Synopsys MetaWare hacc and mcc compilers that resulted in spurious "fixed-point constant is out of range" errors for valid ISO/IEC TR 18037 fixed point literals.

CMPCPP-9165

Failure to find files specified on the build command line no longer causes `cov-run-desktop` to exit with a successful exit code.

CMPCPP-9244

Fixed an issue causing incorrect `PW.BAD_MACRO_REDEF` reports on macros such as `__TIME__` and `__DATE__` when using Microsoft Visual Studio.

CMPCPP-9365

Fixed a crash in `cov-emit` that could occur when processing some large complex macro expansions.

CMPCPP-9370

Fixed the Clang CIT configuration to recognize variants of the `-flto` option that accept an = attached operand, such as `-flto=thin`. Previously, such options were discarded resulting in errors when options that require LTO, such as `-fwhole-program-vtables`, were passed.

CMPCPP-9371

Fixed an issue that caused Clang's `-Wno-register` option to be ignored resulting in unexpected translation failures.

CMPCPP-9382

Fixed the crash in C++17 MSVC compiler configuration with templates with pure `virtuals` functions.

CMPCPP-9410

Fixed an issue in which `cov-translate` died by signal: 6.

CMPCPP-9476

Fixed a crash in `cov-emit` on use of `enum class` in generic `lambda` parameter in Microsoft emulation mode.

CMPCSH-1137

When parsing C# code, using a Windows network path would cause `cov-emit-cs` to crash. This has been fixed.

CMPG-3166

Updated information in Table 7.2.19 of the *Coverity Installation Guide* about supported compilers for Scala.

CMPSCA-199

Build capture now correctly handles the `sourcepath` compiler switch for Scala.

CMPVB-66

When using the `optionstrict` option with the Visual Basic Compiler, `cov-build` would neglect to emit some files. This resulted in too few defects being detected. The `cov-build` command now handles this option correctly.

33.3.3.4. Known issues and solutions

The following known issues and solutions have been identified for this release.

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.

CMPCPP-9122

Clang-based compilers do not support `pragma` methods of annotating deviations and suppressing false positives.

CAP-1176, 97630

`cov-build --instrument` has a known issue when running the `xdcmake.exe` tool of Visual Studio 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.
- Ignore the `xdcmake.exe` process by adding `--capture-ignore xdcmake.exe` to your `cov-build` invocation.

CMPSCA-187

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

CMPJ-368, 65669, 65721

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 `LANG` or `LC_CTYPE` environment variables.

CMPJS-286, 95651

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 used with some frameworks.

33.3.4. Coverity Analysis 2019.12 Commands

The following sections provide information about new and updated commands relating to the build and capture process, analysis, and commands related to Test Advisor.

33.3.4.1. Commands related to the build and capture process

This section provides updates about `cov-build` and related commands, including `capture`, `emit`, and `translate` commands.

33.3.4.1.1. New and changed features

No new and changed features were added for commands related to the build and capture process in 2019.12.

33.3.4.1.2. Bug fixes

The following bugs were fixed for build and capture-related commands (including `emit` and `translate` commands) in 2019.12:

CMPCPP-9128

Fixed an error for `cov-analyze` failing with WUR error for `GNERIC_STATS.7866`.

IM-24307

Fixed an issue in which the `cov-commit-defects` command hangs for a C++ file with very long function names.

SAT-30034

Fixed complexity metric calculation for declaration statements containing conditional expressions.

SAT-31847

Corrected doc to specify the correct option name: `--android-security`, not `--android`.

SAT-32049

Fixed a `cov-analyze` crash due to handling the `class_like_print_writer_for_servlet_output` directive.

33.3.4.1.3. Known issues and solutions

Build-related commands have the following known issues and solutions:

CAP-812, 64428

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 hang occurs, 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 either:

- Uninstall KB2919355

OR

- Add the `--instrument` flag to your `cov-build` invocation:

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

CMPCPP-4764, 72964

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.

SAT-12174, 62745

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.

CAP-332, 26881, 38175

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

Error message:

```
[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
```

33.3.4.2. Commands related to analysis

This section lists new features, bug fixes, and known issues for `cov-analyze` and related commands.

33.3.4.2.1. New and changed features

There were no new features added or changed for commands related to the analysis process in 2019.12.

33.3.4.2.2. Bug Fixes

The following bugs were fixed for analysis-related commands in 2019.12.

SAT-13947

The arguments to `--webapp-security-config` are now properly propagated from central analysis to desktop analysis

SAT-30107

The index page produced by `cov-format-errors` in HTML mode now includes the line number for each defect.

SAT-31372

Fixed an issue that could cause commands to fail with message "ASN CA path length larger than signer error" with self-signed root certificates.

SAT-31584

Fixed an analysis crash with message "Can't print taint location" .

SAT-31831

Fixed an issue that could cause the analysis to run out of memory when a lot of text files were captured.

33.3.4.2.3. Known issues and solutions in 2019.12

There were no known issues for analysis-related commands in 2019.12.

33.3.4.3. Commands related to Test Advisor

There were no new features for Test Advisor in 2019.12.

33.3.4.3.1. Bug fixes in 2019.12

There were no bugs fixed for Test Advisor-related commands in 2019.12.

33.3.5. Coverity Wizard 2019.12

This section lists new features, bug fixes, and known issues related to Coverity Wizard.

33.3.5.1. Important Coverity Wizard Information

- All support for macOS 10.12 is dropped as of 2019.12. (COVP-2103)
- Support for Windows 7 has been dropped in this release. (PRD-11914)
- We are changing how we document the minimum supported platform for CovWizard: The minimum version is 64-bit Linux that can run OpenJRE 1.8 and Eclipse 4.4 (PRD-11977)

33.3.5.2. New and changed features

Coverity Wizard has the following new and changed features in 2019.12.

- Documentation for Coverity Wizard has been updated to reflect GUI changes related to buildless capture. (PRD-11995)

33.3.5.3. Bug fixes

There were no fixed bugs for Coverity Wizard in 2019.12.

33.3.5.4. Known issues and solutions

Coverity Wizard has the following known issues in version 2019.12:

PRD-11727

`cov-wizard` might not successfully emit Java with the default version that is installed in Ubuntu 18.04. (For more information, 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-9245, 90621

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

PRD-9208, 90489

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-8453, 83450

When using a self-signed certificate, if the user chooses not to trust a certificate, they might be prompted multiple times in a row (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-8227, 82196

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-7595, 77742

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.

PRD-6832, 70361

The guided policy creation wizard "Documentation" link fails to open properly on Linux. Open the *Coverity Wizard 2020.12 User Guide* separately to view this documentation.

PRD-6760, 69815

The *Guided Test Advisor Policy Creation Wizard* uses Java regex validation instead of the Perl regex validation that Coverity Analysis Test Advisor users. 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-5770, 59676

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-5387, 54143

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

PRD-5290, 53367

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.

33.3.6. Test Advisor 2019.12

Coverity Test Advisor is a component of the Coverity Analysis installation package.

33.3.6.1. Important Test Advisor information

The following support has been dropped or deprecated in this release:

- SCM support for Perforce 2016.1 has been deprecated. (COVP-2165)
- SCM support for Perforce 2015.2 has been dropped. (COVP-2166)
- Support for SVN 1.9 is deprecated as of 2019.09 and will be removed in a future release. (TADE-2016)

33.3.6.2. New and changed features

Test Advisor has the following new or changed features in 2019.12.

COVP-2162

SCM support for Perforce 2019.1 has been added.

COVP-2163

SCM support for SVN 1.10 - 1.12 has been added.

COVP-2164

SCM support for Mercurial 4.6–5.1 has been added.

33.3.6.3. Bug fixes

No bugs were fixed for Test Advisor in 2019.12.

33.3.6.4. Known issues and solutions

Test Advisor has the following known issues and solutions in 2019.12:

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. You can work around this issue by manually registering the OpenCover profiler DLLs and passing `--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 on this, see the documentation of `cov-build`'s `--cs-no-register-profiler` switch in the *Command Reference*.

33.3.7. Dynamic Analysis 2019.12

Dynamic Analysis is a component of the Coverity Analysis installation package.

33.3.7.1. New and changed features in 2019.12

Dynamic Analysis has no new and changed features in 2019.12.

33.3.7.2. Known issues and solutions

Dynamic Analysis has the following known issues and solutions in 2019.12:

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.

JDA-681, 20788

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

JDA-694, 21417

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, then Dynamic Analysis reports races on neither collections nor arrays.

JDA-720, 22148

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

33.3.8. Architecture Analysis 2019.12

Coverity Architecture Analysis is a component of the Coverity Analysis installation package.

33.3.8.1. Important Architecture Analysis information

Architecture Analysis has the following EOL and deprecated items in 2019.12:

- All support for macOS 10.12 is dropped as of 2019.12. (COVP-2103)

33.3.8.2. Bug Fixes

AA-453

Fixed an error in publishing project after creating new repository in Architecture Analyzer.

33.3.9. Extend SDK 2019.12

Coverity Extend Software Development Kit is a component of the Coverity Analysis installation package.

33.3.9.1. Important information

All support for macOS 10.12 is dropped as of 2019.12. (COVP-2103)

33.3.9.2. Known issues and solutions

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.

33.4. Coverity Desktop 2019.12

The Coverity Desktop plug-in is available for various platforms from the Coverity Connect *Downloads* menu.

33.4.1. Important information

The following products have been deprecated or dropped for this release.

- Support for MacOSX 10.12 has been dropped in this release. (PRD-11850)
- Support for JDK 12 has been dropped in this release. (PRD-11852)
- Support for Windows 7 has been dropped in this release. (PRD-11914)
- The IBM RTC plugin has been replaced with the Eclipse plugin.

Starting from 2019.09, when upgrading the IBM RTC plugin, the user might encounter a screen that prompts for evaluating the changes to be applied before continuing. This includes renaming the plugin feature from `com.coverity.desktop.java.ibm.feature.feature.group` to `com.coverity.desktop.java.feature.feature.group`. This is expected and the user should accept applying these changes and continue with the installation. (PRD-11952, PRD-12001)

- All support for macOS 10.12 is dropped as of 2019.12. (COVP-2103)

33.4.2. Coverity Desktop for Android Studio in 2019.12

The following new and changed features, bug fixes, and known issues were included in this release.

33.4.2.1. New and changed features

Coverity Desktop for Android Studio has the following new and changed feature in 2019.12:

PRD-11943

Support for Android Studio 3.5 has been added in this release.

33.4.2.2. Bug fixes

No bugs were fixed for Coverity Desktop for Android Studio in 2019.12.

33.4.2.3. Known issues and solutions

Coverity Desktop for Android Studio has no known issues in version 2019.12.

33.4.3. Coverity Desktop for Eclipse in 2019.12

The following new and changed features, bug fixes, and known issues were included in this release.

33.4.3.1. Important information

- Support for MacOSX 10.12 has been dropped in this release. (PRD-11850)
- Support for JDK 12 has been dropped in this release. (PRD11852)
- Support for Windows 7 has been dropped in this release. (PRD-11914)
- The IBM RTC plugin has been replaced with the Eclipse plugin.

Starting from 2019.09, when upgrading the IBM RTC plugin, the user might encounter a screen that prompts for evaluating the changes to be applied before continuing. This includes renaming the plugin feature from `com.coverity.desktop.java.ibm.feature.feature.group` to `com.coverity.desktop.java.feature.feature.group`. This is expected and the user should accept applying these changes and continue with the installation. (PRD-11952, PRD-12001)

- Support for Eclipse 4.6 has been dropped in this release. (PRD-11984)

33.4.3.2. New and changed features

Coverity Desktop for Eclipse has the following new and changed features in 2019.12:

PRD-11944

Support for Eclipse 4.13 has been added in this release.

33.4.3.3. Bug fixes

The following bugs were fixed for Coverity Desktop for Eclipse in 2019.12.

PRD-11928

Fixed a `java.lang.StackOverflowError` issue.

33.4.3.4. Known issues and solutions

Coverity Desktop for Eclipse has the following known issues in version 2019.12:

PRD-10694

For OXS 10.14 users with JDK-8136913 installed, using the `hostname_regex` in the `coverity.conf` file causes 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.exe` 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/`.

33.4.4. Coverity Desktop for Microsoft Visual Studio in 2019.12

The following new and changed features, bug fixes, and known issues were included in this release.

33.4.4.1. Important information

No support was deprecated or dropped for this release.

33.4.4.2. New and changed features

Coverity Desktop for Microsoft Visual Studio has no new and changed feature in 2019.12:

33.4.4.3. Bug fixes

Coverity Desktop for Microsoft Visual Studio has the following bug fixes in 2019.12.

PRD-11929

Coverity VS Extension properly validates the configured server url.

33.4.4.4. Known issues and solutions

Coverity Desktop for Visual Studio has no known issues in version 2019.12.

33.4.5. Coverity Desktop for IntelliJ IDEA in 2019.12

The following new and changed features, bug fixes, and known issues were included in this release.

33.4.5.1. Important information

The following support has been deprecated or dropped for this release:

- Support for MacOSX 10.12 has been dropped in this release. (PRD-11850)

- Support for JDK 12 has been dropped in this release. (PRD-11852)
- Support for Windows 7 has been dropped in this release. (PRD-11914)
- Support for IntelliJ 2017.1 has been deprecated in this release. (PRD-11998)

33.4.5.2. New and changed features

Coverity Desktop for IntelliJ IDEA has the following new and changed feature(s) in 2019.12:

PRD-11945

Support has been added for CLion 2019.2, PyCharm 2019.2, IntelliJ 2019.2, PhpStorm 2019.2, RubyMine 2019.2.

33.4.5.3. Bug fixes

Coverity Desktop for IntelliJ has no new bug fixes in 2019.12.

33.4.5.4. Known issues and solutions

Coverity Desktop for IntelliJ IDEA has the following known issues in version 2019.12:

PRD-7453, 76907

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, 80573

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

PRD-7991, 80599

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, 80693

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

PRD-8397, 83106

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

PRD-8042, 80698

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.

PRD-10076, 105052

When using whole program checkers in IntelliJ, a warning about missing class files might be seen 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, 119444

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.

33.5. Coverity Report Generators 2019.12

The Coverity Report Generators' installer can be downloaded from the downloads page in Coverity Connect.

33.5.1. Important Coverity Report Generators information

Coverity Report Generators has the following deprecation in 2019.12.

- The plugin jar is no longer supported Use `plugin1.yaml` instead. (RG-1318)

33.5.1.1. New and changed features

The following features were added or changed for the Coverity Report Generators in 2019.12.

RG-747

The `components` key is now available to filter any kind of report, not just Coverity Integrity Reports. In addition, you can use a new `issue-kind` field to filter a report so that it displays only quality or only security issues.

RG-1308

The Coverity Software Integrity Report can now assess specified defects.

33.5.1.2. Bug fixes

The following bugs were fixed for the Coverity Report Generators in 2019.12.

RG-747

MISRA reports can now be filtered by components.

RG-1313

Fixed a problem with the CVSS report not updating scores on the linked streams in a project.

33.5.1.3. Known issues and solutions

Coverity Report Generators have the following known issues and solutions:

RG-1128

For ATP-based systems, you may 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 do encounter this error message, please install these missing libraries: `libgl1, libgl1-mesa-dri, and libgl1-mesa-glx.`

RG-1260, RG-1232

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 Poretcode SC.

33.6. Coverity Documentation 2019.12

No new documents were added nor organizational changes were made in 2019.12:

Chapter 34. Coverity 2019.09-6 Release Notes

Table of Contents

34.1. Coverity Platform 138

34.1. Coverity Platform

IM-23942

A bug was fixed for a situation in which repeatedly soft-deleting users caused user names so long that DB errors were raised.

Chapter 35. Coverity 2019.09-5 Release Notes

Table of Contents

35.1. Coverity Analysis	139
-------------------------------	-----

35.1. Coverity Analysis

SATW-3410

Fixed a false positive of CERT STR34-C when using char type for non-character data.

SATW-3411

Fixed a false positive of CERT ARR37-C when accessing array member of a struct.

Chapter 36. Coverity 2019.09-4 Release Notes

Table of Contents

36.1. Coverity Analysis	140
36.2. Compiler configuration, Build capture, and Compiler Integration Toolkit (CIT) bug fixes	140
36.3. Coverity Platform	140

36.1. Coverity Analysis

CMPCPP-9577

Fixed a false negative of MISRA C-2012 Rule 3.2 about CR/LF in the end of comment line.

SAT-32632

Fixed a false negative in MISRA C++-2008 Rule 17-0-5 on Windows platform.

36.2. Compiler configuration, Build capture, and Compiler Integration Toolkit (CIT) bug fixes

CMPSCA-201

Build capture now correctly handles the `sourcepath` compiler switch for Scala.

36.3. Coverity Platform

IM-24539

Fixed an issue that hindered the ability of Coverity Connect to load and display triage information for certain defects.

Chapter 37. Coverity 2019.09-3 Release Notes

Table of Contents

37.1. Coverity Platform bug fixes	141
---	-----

37.1. Coverity Platform bug fixes

PRD-12001

The IBM RTC plugin has been replaced with the Eclipse plugin.

Starting from version 2019.09, when upgrading the IBM RTC plugin, the user might encounter a screen that prompts for evaluating the changes to be applied before continuing. This includes renaming the plugin feature from `com.coverity.desktop.java.ibm.feature.feature.group` to `com.coverity.desktop.java.feature.feature.group`. This is expected and the user should accept applying these changes and continue with the installation.

Chapter 38. Coverity 2019.09-2 Release Notes

Table of Contents

38.1. Coverity Analysis bug fixes	142
38.2. Coverity Documentation Release Notes	142

38.1. Coverity Analysis bug fixes

CMPSWIFT-302

Fixed a memory corruption issue that affected translation of subscript expressions.

SAT-32194

Fixed an issue that could cause commands to fail with message "ASN CA path length larger than signer error" with self-signed root certificates.

38.2. Coverity Documentation Release Notes

The following documentation fixes were made in Release Notes:

COVDOCS-14

Fixed errant text formatting in the Coverity Analysis User and Administration Guide.

SAT-31893

Reinserted images that were missing in the Coverity Platform User and Administration Guide.

Chapter 39. Coverity 2019.09-1 Release Notes

Table of Contents

39.1. Coverity Analysis bug fixes	143
---	-----

39.1. Coverity Analysis bug fixes

CMPCPP-8934

Fixed an issue where Coverity failed with declarations of the vector data types for gcc. (*This issue was fixed in the base 2019.09 version.*)

CMPCPP-9063

Fixed an issue where Coverity failed with undefined `_Float64x` and `_Float128` for gcc. (*This issue was fixed in the base 2019.09 version.*)

CMPCPP-9364

Fixed an issue affecting gcc and Synopsys MetaWare hcac and mcc compilers that resulted in spurious "fixed-point constant is out of range" errors for valid ISO/IEC TR 18037 fixed point literals.

CMPG-3114

Fixed an issue by which the `cov-emit` command incorrectly rejected some ISO/IEC TR 18037 fixed point literal values. This issue had affected the gcc and Synopsys MetaWare hcac and mcc compilers.

Chapter 40. Coverity 2019.09 Release Notes

Table of Contents

40.1. Important information for 2019.09	144
40.2. Coverity Platform 2019.09	146
40.3. Coverity Documentation 2019.09	150

40.1. Important information for 2019.09

Support for this version of Coverity will be discontinued 18 months after the 2019.12 release.

Due to a change in our bug tracking system, items are now identified by two bug numbers:

- One specifying the identity of the bug in our **old** bug tracking system, formatted like this: XXXXXX. (For example, 374568.)
- One specifying the identity of the bug in our **new** bug tracking system, formatted like this: CODE-XXXXX. (For example, IM-22788.)

40.1.1. Deprecated and End-of-Life (EOL) Products in Coverity 2019.09

Support for the following products, features, platforms, and third-party tools is classified as deprecated or end-of-life as of the Coverity 2019.09 release.

40.1.1.1. Deprecated Products

Support for the following products and features is deprecated as of the Coverity 2019.09 release.

Table 40.1. Deprecated products

Product	See also...
Architecture Analysis build-check tool	Supported Platforms for Coverity Analysis
Architecture Analysis Eclipse plug-in	Supported Platforms for Coverity Analysis
Architecture Analysis IntelliJ plug-in	Supported Platforms for Coverity Analysis
__coverity_format_string_sink__ built-in primitive	Users can now use __coverity_taint_sink__(arg, FORMAT_STRING) instead.
Coverity Desktop support for Java 12	Supported Platforms for Coverity Analysis
File system capture	Buildless capture can be used in place of filesystem capture. Please see the section "Moving from filesystem capture to buildless capture" in the <i>Coverity Analysis User and Administrator Guide</i> for more information. (BLC-827)

Product	See also...
FreeBSD 11.1	Supported Compilers: Coverity Analysis for C/C++
FreeBSD 8.4	Supported Compilers: Coverity Analysis for C/C++
GNU GCC and G++ 4.2.1 compiler on FreeBSD 8.4	Supported Compilers: Coverity Analysis for C/C++
Microsoft Embedded C++ 4.0 compiler	Supported Compilers: Coverity Analysis for C/C++
Microsoft Visual Studio 2010 and 2012	Supported Platforms for Coverity Analysis
OpenJDK 12	Supported Platforms for Coverity Analysis
Plugin jar	Start using plugin.yaml (refer: plugins/plugin.html) instead.
Sun/Oracle JDK 12	Supported Platforms for Coverity Analysis
SCM support for SVN 1.9	Coverity Test Advisor Supported SCM Systems
Swift 4.2.x	Supported Compilers: Coverity Analysis for Swift
Windows 7, Coverity Analysis support for Windows 7	Supported Platforms for Coverity Analysis

40.1.1.2. End-of-Life Products

Support for the following products and features is dropped in the Coverity 2019.09 release.

Table 40.2. End-of-Life Products

Product	See also...
Android Jack Compiler	Supported Compilers: Coverity Analysis for C/C++
Architecture Analysis on 32-bit Windows platforms	Supported Platforms for Coverity Analysis
ARM ADS 1.1–1.2 C/C++	Supported Compilers: Coverity Analysis for C/C++
ARM RVDS 2.0–4.1 compilers	Supported Compilers: Coverity Analysis for C/C++
GNU GCC and G++ 2.*	Supported Compilers: Coverity Analysis for C/C++
HI-TECH PICC compiler	Supported Compilers: Coverity Analysis for C/C++
IAR Embedded Workbench C/C++ 7.30B–8.10 compiler for the 8051 processor	Supported Compilers: Coverity Analysis for C/C++
SNC C/C++	Supported Compilers: Coverity Analysis for C/C++
SNC GNU C/C++ compiler	Supported Compilers: Coverity Analysis for C/C++
STMicroelectronics GNU C/C++ compilers	Supported Compilers: Coverity Analysis for C/C++
STMicroelectronics ST Micro C/C++ compilers	Supported Compilers: Coverity Analysis for C/C++
Sun/Oracle JDK 10	Supported Platforms for Coverity Analysis
TCS compiler	Supported Compilers: Coverity Analysis for C/C++
Visual C++ 6, 2003 compilers	Supported Compilers: Coverity Analysis for C/C++
Visual Studio 2008	Supported Compilers: Coverity Analysis for C/C++

40.2. Coverity Platform 2019.09

This section provides release notes for Coverity Platform components.

40.2.1. Coverity Connect 2019.09

Coverity Connect is a component of the Coverity Platform installation package.

40.2.1.1. Important Coverity Connect Information

There have been no deprecations or EOL'd items for Coverity Connect this release:

40.2.1.1.1. New and changed features

The following new and changed features have been added for Coverity Connect this release:

- The language (info from the committed defect) is now included on both the commit preview and triage store import/export. (IM-21182)
- The Oshi library is used instead of Sigar for getting OS and hardware info. For pre-requisites and dependencies, please consult the *Coverity Install Guide*. (IM-23990, IM-23745)
- Added support for AWS RDS PostgreSQL as an external managed database for Coverity. (IM-23805)
- The embedded PostgreSQL version has been upgraded from 10.4 to 10.9. Coverity Connect now supports external PostgreSQL databases up to 10.9. (IM-24187)
- Added support for capturing and analyzing C# projects that use the Unity Roslyn compiler on MacOS. (INS-2711)

40.2.1.1.2. Bug fixes

The following bugs are fixed for Coverity Connect:

IM-20712

Documentation was updated to fix a typo.

IM-23364

A workaround has been provided for this commit problem.

IM-23457

Updated documentation to specify that Coverity Connect upgraded the embedded PostgreSQL version from 10.4 to 10.9. Coverity Connect now supports external PostgreSQL databases up to 10.9.

IM-23583

Documentation has been updated to explain how to use a secure SSL setting for customers deploying to the cloud.

IM-23669

An error page was updated to show correct information about Support email address.

IM-23687

Two charts were updated in documentation to show which roles are impacted by classify issues permission.

IM-23796

We have reduced the database size by optimizing the internal relationships on the defect storage model.

IM-23812

Documentation was updated to explain how to retrieve information for more than one CID using the Web Services API.

IM-23836

A bug was fixed in which selecting a defect and pressing the Enter key multiple times caused defects to be exported to JIRA.

IM-23870

Documentation has been updated to specify the default set of status for information returned by `getMergedDefectsForStreams`.

IM-23926

A bug was fixed that caused the program to hang when the user edited component file mapping rules.

IM-23959,

The *Coverity Platform User and Administrator Guide* has been updated to provide correct information about adding and configuring a reverse proxy.

IM-23988

A bug has been fixed that allowed a user with the Observer role at project level to triage defects using `updateTriageForCIDsInTriageStore()` or using the `cov-manage-im` command.

IM-23989

The `cov-manage-im` command in stream mode now displays all fields.

IM-23997

Preview report json file is now pretty-printed again.

IM-23998

As part of fix for IM-23384, `getSignInConfigurationRequest` and `updateSignInConfigurationRequest` web service calls will not return `enableSessionTimeout` any more.

IM-24011

The `notify.using.configuration.from.address.only` is now correctly documented.

IM-24026

Soft-deleted users are now excluded from counts and `groups.json`.

IM-24028

Release notes in 2019.03, 2019.06, and 2019.09 were updated: References to Java8 were removed.

IM-24111

A new column named "Last Triaged User" that shows the last user who triaged has been added to views.

40.2.1.1.3. Known issues and solutions

Coverity Connect has the following known issues:

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.

CPU-38, 82579

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.

CPU-17, 80045

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.

IM-16076, 67748

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, 75559

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

IM-17660, 75263

An error occurs when a custom role is created using a multi-word rolename that is the same as a built-in rolename, even if there are case differences between the two rolenames.

IM-18707, 82643

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, 82648

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, 84453

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, 89897

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, 89946

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-23994

Internet Explorer 11 fails on operations using file upload.

INS-1274, 63454

Although the upgrade doc 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, fall back to backup-and-restore upgrade.

INS-1477, 73401

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 by specifying a max heap setting for `cov-im-daemon`.

INS-2133, 112939

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, 118662

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.
```

40.2.2. Coverity Policy Manager 2019.09

Coverity Policy Manager is a component of the Coverity Platform installation package.

40.2.2.1. Important Coverity Policy Manager information for 2019.09

There are no deprecated or EOL items for 2019.09:

40.2.2.2. Bug fixes in 2019.09

Coverity Policy Manager™ has the following fixed bugs in 2019.09:

IM-22605

Documentation has been updated to clarify Policy Manager data lifecycle.

IM-22804

A bug was fixed for a Dashboard display that was cut off and missing text.

40.3. Coverity Documentation 2019.09

The following new documents and changes were made in 2019.09:

SAT-30024

Enhanced the document to clarify how to enable CERT-C/CPP checkers. See "Running coding standard analyses" in the *Coverity Analysis User and Administrator Guide*.

Chapter 41. Coverity 2019.06-11 Release Notes

Table of Contents

41.1. Coverity Platform	151
-------------------------------	-----

41.1. Coverity Platform

IM-24613

In rare cases, the `getStreamDefects` API could encounter an error and fail to return the requested data. This is now fixed.

Chapter 42. Coverity 2019.06-10 Release Notes

Table of Contents

42.1. Coverity Analysis	152
-------------------------------	-----

42.1. Coverity Analysis

SAT-33349

Fixed a source of `OVERRUN` false positives when a function accesses a buffer using a minimum value multiplied by a constant.

SAT-33387

Fixed some serious performance issues affecting analysis of C++ codebases with compliance standards.

Chapter 43. Coverity 2019.06-9 Release Notes

Table of Contents

43.1. Coverity Analysis	153
-------------------------------	-----

43.1. Coverity Analysis

CMPCPP-9639

Assertion for `mangled_encoding_for_sizeof_pack` has been eliminated.

Chapter 44. Coverity 2019.06-7 Release Notes

Table of Contents

44.1. Coverity Platform bug fixes	154
---	-----

44.1. Coverity Platform bug fixes

IM-24295

Fixed an issue when displaying source code for .cshhtml and other file types that are not initially parsed correctly.

Chapter 45. Coverity 2019.06-6 Release Notes

Table of Contents

45.1. Coverity Analysis bug fixes	155
---	-----

45.1. Coverity Analysis bug fixes

CMPCPP-8880

Fixed a crash in `cov-emit` with complementary information enabled when emitting a redeclaration of some functions.

Chapter 46. Coverity 2019.06-5 Release Notes

Table of Contents

46.1. Coverity Analysis bug fixes	156
---	-----

46.1. Coverity Analysis bug fixes

CMPCPP-9028

Fixed a stack overflow in `cov-emit` when using the `--emit_complementary_info` option after being prompted by a long macro definition.

CMPCPP-9040

Fixed a case of emit DB corruption triggered by use of GNU multiversioning on a member function.

INS-2760

Installing Coverity Analysis will not fail if Windows UAC dialog is not accepted.

SAT-30196, SAT-30310

Fixed a source of `OVERRUN` false positives involving certain implementations of the `va_start` function.

SATW-3066, SATW-3067

Fixed a false positive in `CERT_EXP62-CPP` that occurred when using `memset` on an array of pointers to class objects.

SATW-3069

Fixed a false positive in `CERT_FLP32-C` where user-defined functions were mistaken for standard math functions.

SATW-3070, SATW-3071

Fixed a false positive in `CERT_OOP51-CPP` where an expected object slicing did not occur.

SATW-3072, SATW-3073

Fixed a false positive in `CERT_OOP57-CPP` that occurred when using `memset` on an array of pointers to class objects.

Chapter 47. Coverity 2019.06-1 Release Notes

Table of Contents

47.1. Coverity Analysis bug fixes	157
---	-----

47.1. Coverity Analysis bug fixes

SAT-30196, SAT-30310

Fixed a source of `OVERRUN` false positives involving certain implementations of the `va_start` function.

Chapter 48. Coverity 2019.06 Release Notes

Table of Contents

48.1. Important information for 2019.06	158
48.2. Coverity Platform 2019.06	159
48.3. Coverity Analysis 2019.06	163
48.4. Coverity Desktop 2019.06	181
48.5. Coverity Report Generators 2019.06	184
48.6. Coverity Documentation 2019.06	185

48.1. Important information for 2019.06

Due to a change in our bug tracking system, items are now identified by two bug numbers:

- One reflecting the identity of the bug in our **old** bug tracking system, formatted like this: XXXXXX. (For example, 374568.)
- One reflecting the identity of the bug in our **new** bug tracking system, formatted like this: CODE-XXXXX. (For example, IM-22788.)

 **Note**

Bugs with only a CODE-XXXXX number do not have an old number.

48.1.1. Deprecated and End-of-Life (EOL) Products in Coverity 2019.06

Support for the following products, features, platforms, and third-party tools is classified as deprecated or end-of-life as of the Coverity 2019.06 release.

48.1.1.1. Deprecated Products

Support for the following products and features is deprecated as of the Coverity 2019.06 release.

Table 48.1. Deprecated products

Product	Comments
Eclipse 4.6 support	Coverity Coverity Desktop for Eclipse on supported platforms
Git 1.8-2.1 support	Coverity Test Advisor Supported SCM Systems
JDK 1.6 for macOS support	Supported Compilers: Coverity Analysis for C/C++
Mercurial 3.1 and 3.2 support	Coverity Test Advisor Supported SCM Systems
Oracle JDK 10 support	Supported Compilers: Coverity Analysis for C/C++
Perforce 2015.2 support	Coverity Test Advisor Supported SCM Systems

48.1.1.2. End-of-Life Products

Support for the following products and features is dropped in the Coverity 2019.06 release.

Table 48.2. End-of-Life Products

Product	Comments
Accurev 6.2 support	Coverity Test Advisor Supported SCM Systems
AIX 6.1 support	Supported Compilers: Coverity Analysis for C/C++
Architecture Analysis on Windows 32-bit systems	Supported Platforms for Coverity Analysis
Git 1.4-1.7 support	Coverity Test Advisor Supported SCM Systems
Glibc 2.12-2.13 support	Supported Platforms for Coverity Analysis
IntelliJ 2016.x support	Coverity Desktop for IntelliJ on supported platforms
Mercurial 1.0-3.0 support	Coverity Test Advisor Supported SCM Systems
NetBSD 6.1 and earlier support	Supported Platforms for Coverity Analysis
Perforce 2014.2 and 2015.1 support	Coverity Test Advisor Supported SCM Systems
RubyMine, WebStorm, and PyCharm support	Coverity Desktop for IntelliJ on supported platforms
Swift 4.0-4.1.2 compiler support	Supported Compilers: Coverity Analysis for Swift
TFS 2010 support	Coverity Test Advisor Supported SCM Systems
Windows Server 2008 R2 support	Supported Platforms for Coverity Analysis

48.2. Coverity Platform 2019.06

This section provides release notes for Coverity Platform components.

48.2.1. Coverity Connect 2019.06

Coverity Connect is a component of the Coverity Platform installation package.

48.2.1.1. Important Coverity Connect Information

There have been several deprecations for Coverity Connect this release:

- Dropped support for glibc 2.12-2.13. Currently, we are supporting glibc 2.14 or higher. (IM-23864)
- Dropped Coverity Platform support for Windows Server 2008. (INS-2668)

48.2.1.1.1. New and changed features

The following new and changed features have been added for Coverity Connect this release:

- Added the `-deststoretype JKS` option to the procedure for configuring Coverity Connect with TLS/SSL.

For more information, see Section 3.1.1.8. "Configuring Coverity Connect for TLS/SSL" in the Coverity Platform 2019.06 User and Administrator Guide. (IM-23655)

- Added support for PostgreSQL 10.4. (IM-23686)

48.2.1.1.2. Bug fixes

The following bugs are fixed for Coverity Connect:

IM-22426

Fixed a broken link inside an email notification.

IM-22776

Fixed a commit issue caused by a database constraint violation exception.

IM-22822

Fixed the commit failures that resulted from improper use of scrollable results.

IM-23112

Fixed an issue involving password reset failures.

IM-23264

If the database backup fails, an email notification now gets sent to an administrator.

IM-23288

Fixed occasional failures that occurred when generating a Coverity Integrity Report that included large amounts of defects.

IM-23295

Fixed an upgrade failure that resulted from a JVM GC misconfiguration.

IM-23384

Updated the Coverity Connect UI to reflect a (required) default timeout after an inactive session. The default timeout no longer displays as an optional setting under Sign In Settings. Instead, users can now only specify the number of inactive minutes before they are automatically logged out.

IM-23482

Fixed PostgreSQL upgrade failures that resulted from incorrect parsing of external database version numbers.

IM-23545

Fixed an issue which now takes `rowCount` into account for any offset values when accessing `/api/viewContents/*` via curl or wget.

IM-23623

Fixed an error that occurred when multiple triage stores were removed during the coordinator-subscriber synchronization process when triage stores were being used by subscriber streams.

INS-2500

Improved the generic error messages which a user might encounter when backing up the Coverity Connect database during an upgrade.

INS-2535

The `config/system.properties` file now also lists the `os_user` property. The `os_user` property ensures that the installation is being run by the correct user.

48.2.1.1.3. Known issues and solutions

Coverity Connect has the following known issues:

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.

CPU-38, 82579

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.

CPU-17, 80045

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.

IM-16076, 67748

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, 75559

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

IM-17660, 75263

An error occurs when a custom role is created using a multi-word rolename that is the same as a built-in rolename, even if there are case differences between the two rolenames.

IM-18707, 82643

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, 82648

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, 84453

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, 89897

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, 89946

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.

INS-1274, 63454

Although the upgrade doc 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, fall back to backup-and-restore upgrade.

INS-1477, 73401

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 by specifying a max heap setting for `cov-im-daemon`.

INS-2133, 112939

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, 118662

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`.

48.2.2. Coverity Policy Manager 2019.06

Coverity Policy Manager is a component of the Coverity Platform installation package.

There are no new or changed features, and no bug fixes or known issues for this release.

48.3. Coverity Analysis 2019.06

This section provides updates about Coverity Analysis components.

48.3.1. Important Coverity Analysis information

There have been several deprecations and EOLs this release:

- Deprecated support for macOS 10.12. (COVP-2108)
- Deprecated support for JDK 1.6 on macOS. (COVP-2122)
- Deprecated support for Oracle JDK 10. (COVP-2122)
- Dropped support for Windows Server 2008 R2. (COVP-2126)
- Dropped support for Swift 4.0-4.1.2. (CMPSWIFT-265)
- The `_coverity_tainted_data_sink_` and `_coverity_tainted_string_sink_content_` primitives have been deprecated. These primitives are replaced by `_coverity_taint_sink_`, which allows specification of `TaintSinkType`. (SAT-28684)
- Deprecated support for the following function annotations: `tainted_data_return`, `tainted_data_argument`, `tainted_string_return_content`, and `tainted_string_argument`

These function annotations have been replaced by `taint_source`, which can be used to uniformly specify a source of tainted data across the following C/C++ checkers: `TAINTED_SCALAR`, `TAINTED_STRING`, `OS_CMD_INJECTION`, `PATH_MANIPULATION`, `SQLI`, `URL_MANIPULATION`, and `XPATH_INJECTION`. (SAT-28914)

48.3.1.1. New and changed features

Coverity Analysis has the following new and changed features:

- Added support for Swift 4.2. (CMPSWIFT-213)
- Added support for OpenJDK 11 on Linux and Windows. (COVP-2122)
- Coverity Analysis has been further broken down into components providing the ability to select new pieces of the product to omit.

For more information, see Section 2 "Installing Coverity Analysis components" in the Coverity 2019.06 Installation and Deployment Guide. (INS-2564)

- Added the `--component.cov-wizard` option to the Coverity Analysis silent installer. The `--component.cov-wizard` option controls whether Coverity Wizard is included in the installation.

For more information, see Section 2.2.2. "Coverity Analysis silent installer" in the Coverity 2019.06 Installation and Deployment Guide. (INS-2628)

- Added Server Name Indication (SNI) support to Coverity Analysis tools. (SAT-22256)
- Added cross file support for Java and .NET security checkers run with `cov-run-desktop` and Coverity Analysis summaries. (SAT-25191)
- Added taint flow modeling for STL APIs. (SAT-28879)
- Removed the `cov-template-da` binary. Its functionality has been moved into the `cov-security-da` binary, which will now automatically run at the end of a filesystem capture for a JavaScript project. (SAT-28979)
- Added a way to mark specific arguments as tainted in the `simple_entry_point` directive. (SAT-29063)
- Added a new version (v7) to the JSON defect output format, which includes a new optional *MISRA Category* field for MISRA defects. The following MISRA categories are now available: Advisory, Required, and Mandatory. (SAT-29154)
- Added the `--MISRA-category-regex` option to filter `cov-run-desktop` defects based on MISRA category. (SAT-29200)
- Added models for the GUPnP C/C++ Library. (SAT-29441)
- Added models for the libmicrohttpd C/C++ Library. (SAT-29459)
- Added models for the GLib C/C++ Library. (SAT-29460)
- Added models for the libcurl C/C++ Library. (SAT-29463)
- Added C/C++ models for GNET APIs to detect `URL_MANIPULATION` defects. (SAT-29524)
- Added C/C++ models for the libxml2 library to detect `PATH_MANIPULATION` defects. (SAT-29614)
- Added support for doT.js, Hogan, Lodash, Twig, and Underscore JavaScript template engines. (SAT-29713)
- `cov-run-desktop` now supports sorting by MISRA Category level. (SAT-29736)

48.3.1.2. Bug fixes

INS-2590

Installation can now successfully be aborted when a user is asked to continue installation on an existing directory.

INS-2597; INS-2091

Fixed an issue where an unattended installer would not work on the Windows command line.

SAT-7226; SAT-28978

`TAINTED_STRING` no longer reports on dataflow from the environment back to the environment.

SAT-21704

Fixed an issue that could cause false positives in rules such as AUTOSAR C++ 14 A0-1-3, where the checker would falsely claim that a symbol was never referenced.

SAT-22637

`cov-run-desktop` text output now includes "MISRA Category" values when reporting MISRA defects.

SAT-23842

Fixed a bug where Java, C#, and Visual Basic .NET analysis might generate bogus new defects every analysis run due to anonymous inner classes.

SAT-25623

Fixed a `PATH_MANIPULATION` false negative involving a call to the `xmlReadFile` function in `libxml`.

SAT-26483

Improved Coverity Analysis to avoid false positives if an object was assigned a temporary value and that object also contained a field that affected a condition.

SAT-26651

In earlier releases, Coverity Fortran Syntax Analysis failed with a memory access violation when run under Clear Linux 4.14-64. This issue has been resolved.

SAT-27176

Fixed a class of `NULL_RETURNS` false negatives where a field of a pointer was checked for nullity and then returned even when it was on a null path.

SAT-28072

Fixed an issue that caused SSL verification failures with the following error message: "Server's SSL certificate is not trusted. Its CA certificate was found but a chain of trust could not be constructed." This error message would appear when multiple CA certificates with the same name were installed.

SAT-28112

`cov-analyze` now only analyzes one file when multiple files end up being the same after the effects of the `--strip-path` option.

SAT-28384

Fixed a class of `USE_AFTER_FREE` false positives where previously derived models were being used for current recursive functions.

SAT-28588

Fixed a class of `SINGLETON_RACE` false positives that occurred when Coverity Analysis did not properly recognize a `RequestScope` annotation.

SAT-28661

Updated `PRINTF_ARGS`'s default behavior so that it no longer reports on mismatches of integral types with the same bit size. `strict_integral_type_match`, a new checker option, was added to revert `PRINTF_ARGS` back to its previous behavior.

SAT-28697

Fixed an issue that would cause `PRINTF_ARGS` false positives when using the `%s` specifier in a call to the `wprintf` function on Windows.

SAT-28828

Fixed a class of `OVERRUN` false positives which involved retrieving a buffer pointer offset.

SAT-28839

Moved the location of defects (when reported from the function declaration) for Flask views in Python applications to the return statement.

SAT-28936

Fixed an error that would cause an unrecoverable analysis crash when AUTOSAR C++14 Rule A2-7-1 was enabled and a source file contained a `//`, immediately followed by a new line.

SAT-28938

Fixed an issue that could cause unrecoverable (and sometimes silent) analysis crashes when a file contained a very large number of coding standard violations (for standards such as MISRA C-2012 Rule 20.5).

SAT-28981

Updated `PRINTF_ARGS` to understand the Microsoft-specific `I32`, `I64`, and `I` size attributes.

SAT-28983

Fixed an error where invoking the version of `swprintf` that does not take a size argument resulted in `OVERRUN` false positives.

SAT-28984

Fixed a class of `INVALIDATE_ITERATOR` false positives where an end iterator was passed as an argument to `std::prev`.

SAT-29074

Custom text checkers were documented as being enabled by default, although they required explicit enablement. They are now correctly enabled by default. (See the `TEXT.CUSTOM_CHECKER` section in the Checker Reference.)

SAT-29091

Fixed a bug where SpotBugs would generate bogus new defects during every analysis run due to the usage of lambdas.

SAT-29098

Fixed an issue where non-detection of feature settings in XML parsers resulted in `XML_EXTERNAL_ENTITY` false positive defects.

SAT-29182; SAT-29715

Fixed a source of `STRAY_SEMICOLON` false positives when using the C++ `if constexpr` construct.

SAT-29475; SAT-29888

Fixed a source of `UNINIT` false positives with the `enable_write_context` option.

SAT-29484

Fixed a `BUFFER_SIZE` false positive by adding support for compound operators.

SAT-29487

Modeled APIs in `libsoup` (C/C++ Library) as taint sources and `URL_MANIPULATION` taint sinks.

SAT-29494

Fixed an analysis crash that resulted in a "File name was not obtained from `getCasePreservedFileName`" error message when the HFA checker was enabled on Windows.

SAT-29523

Modeled APIs in `libfetch` (C/C++ Library) as `URL_MANIPULATION` taint sinks.

SAT-29527

Modeled taint sources for LZ4 APIs.

SAT-29528; SAT-29530

Fixed an analysis crash that occurred when using the `--enable-constraint-fpp` option while running compliance checkers.

SAT-29709

Modeled taint sources in File, Networking, and Data Conversion APIs in the C/C++ GIO library. (SAT-29709)

SAT-29721

Fixed event messages erroneously printing `<arg-1>` instead of referencing `this`.

SAT-29806

Fixed a bug that prevented code annotations from suppressing MISRA defect reports.

SAT-29816

Fixed a memory leak that could cause performance issues (especially when running compliance checkers).

SAT-29844

Fixed a crash in JavaScript analysis involving large codebases during the generation of a call graph.

SAT-29945

Fixed a `cov-analyze` Analysis Master (AM) assertion failure that could occur when attempting to detect JavaScript duplicate source files that involved source maps.

SAT-29998

Improved Coverity Analysis with Clang compilers so that false positives are avoided when an object is assigned a temporary value and that object also contains a field that affects a condition.

SAT-30137

Fixed a crash in `cov-analyze` involving `catastrophic signal: 22 (SIGABRT)`, which sometimes resulted from very large output files on Windows.

SATW-1766

Fixed a false positive in MISRA C-2012 Rule 18.4 where a pointer to an array decayed into a pointer.

SATW-2965

Fixed a false positive in CERT OOP57-CPP that occurred when using `memset` on an array of pointers to class objects.

SATW-2966

Fixed a false positive in CERT OOP51-CPP where an expected object slicing did not occur.

SATW-2967

Fixed a false positive in CERT FLP32-C where user-defined functions were mistaken for standard math functions.

SATW-2968

Fixed a false positive in CERT EXP62-CPP that occurred when using `memset` on an array of pointers to class objects.

SATW-2984

Fixed a false positive in MISRA C-2012 Rule 7.2 involving macro arguments.

SATW-2985

Fix a false positive in MISRA C-2012 Rule 10.3 where a boolean literal `false` was used in a struct field initializer.

SATW-3002

Fixed the wrong event messages for MISRA C-2012 Rule 10.3 in the Japanese locale.

SATW-3003

Fixed a false positive in MISRA C++-2008 Rule 0-1-10 where a virtual function was called.

SATW-3019

Fixed a MISRA C-2012 Rule 14.4 false positive where `false` was used as a conditional expression for a while statement in a function-like macro.

SATW-3023

Fixed a MISRA C++-2008 Rule 8-5-2 false positive involving zero initialization of float types.

48.3.1.3. Known Issues

IM-23994

Internet Explorer 11 fails on operations using file upload.

48.3.2. Coverity Analysis checkers and user directives in 2019.06

The following sections describe new and updated features, bug fixes, and known issues for Coverity checkers and associated elements.

48.3.2.1. New and updated checkers and directives

The following table lists new checkers and the languages they support.

Checker	Languages
CONFIG.CSURF_IGNORE_METHODS	JavaScript
CONFIG.MYBATIS_MAPPER_SQLI	Java
JSONWEBTOKEN_UNTRUSTED_DECODE	JavaScript
REACT_DYNAMIC_URL_INSECURE_TARGET	JavaScript

The following table documents added language support for existing checkers.

Languages	Checkers	Checkers
C/C++	URL_MANIPULATION	
Go	CONSTANT_EXPRESSION_RESULT	IDENTICAL_BRANCHES
	COPY_PASTE_ERROR	REVERSE_INNULL
	DIVIDE_BY_ZERO	UNINTENDED_INTEGER_DIVISION
	FORWARD_NULL	UNUSED_VALUE
TypeScript	EXPLICIT_THIS_EXPECTED	NULL_RETURNS
	FORWARD_NULL	REVERSE_INNULL
	NO_EFFECT	

New and changed checkers

SAT-23228

CONFIG.MYBATIS_MAPPER_SQLI, a new checker, reports unescaped variable substitution in iBatis and MyBatis Mapper XML files.

SAT-23687

Improved the defect merging strategy for user-defined dataflow checkers created with DF.CUSTOM_CHECKER, so that these checkers better distinguish separate defects from the same checker. These checkers now use the same defect merging strategy as built-in dataflow checkers. As a result, some defects from such checkers that were previously merged into occurrences of the same CID are now reported as separate CIDs. These separate CIDs can now be triaged separately.

SAT-26837

C/C++ security checkers now use the `__coverity_taint_sink__` primitive, which allows for specification of sink type.

SAT-27303

Added `yarrow_start` and `rc4_start` LibTomCrypt API functions to the DC.WEAK_CRYPTO checker.

SAT-28492

Added TypeScript support for the EXPLICIT_THIS_EXPECTED quality checker.

SAT-28537

Added TypeScript support for the FORWARD_NULL quality checker.

SAT-28618

Added TypeScript support for the NULL_RETURNS quality checker.

SAT-28619

Added TypeScript support for the REVERSE_INULL quality checker.

SAT-28684; SAT-28913

Added the `__coverity_taint_sink__` primitive. The advantage of this new primitive is that it allows specification of a TaintSinkType.

This primitive is shared by the following C/C++ security checkers: TAINTED_STRING, TAINTED_SCALAR, OS_CMD_INJECTION, PATH_MANIPULATION, SQLI, XPATH_INJECTION, URL_MANIPULATION, and INTEGER_OVERFLOW.

The `__coverity_taint_sink__` primitive also replaces the `__coverity_tainted_string_sink_content__` and `__coverity_tainted_data_sink__` primitives.

SAT-29049

CONFIG.CSURF_IGNORE_METHODS, a new checker, finds cases (such as POST, PUT, and DELETE) where the csrf middleware is configured to ignore requests with HTTP methods that change server state.

SAT-29050

JSONWEBTOKEN_UNTRUSTED_DECODE finds cases where JWT tokens are decoded but their signature is not verified. If the token is not verified, attackers might submit forged tokens and gain access to sensitive data and functionality.

SAT-29052

REACT_DYNAMIC_URL_INSECURE_TARGET finds cases where a link is dynamically generated and is set to open a new window by virtue of its target attribute being set to `_blank`. Third-party sites opened from such links are able to redirect the original window or tab to an arbitrary URL without user interaction. When returning to the original window or tab, a user might be tricked into disclosing sensitive information through a phishing attack.

SAT-29055

The IDENTICAL_BRANCHES checker now supports Go.

SAT-29056

Added a new C/C++ checker: URL_MANIPULATION.

SAT-29064

The CONSTANT_EXPRESSION_RESULT checker now supports Go.

SAT-29080

The COPY_PASTE_ERROR checker now supports Go.

SAT-29083

The `DIVIDE_BY_ZERO` checker now supports Go.

SAT-29084

The `FORWARD_NULL` checker now supports Go.

SAT-29087

The `REVERSE_INULL` checker now supports Go.

SAT-29088

The `UNINTENDED_INTEGER_DIVISION` checker now supports Go.

SAT-29089

The `UNUSED_VALUE` checker now supports Go.

SAT-28767

Added support for Ruby on Rails major version series 6.x to the Ruby security checkers.

SAT-29573

Further increased Coverity's coverage of the AUTOSAR C++14 standard, 18-10 edition.

SAT-29698

Added TypeScript support for the `NO_EFFECT` quality checker.

48.3.2.2. Known issues and solutions

SAT-7224, 43971: `XSS`

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

SAT-17490, 84256: `INTEGER_OVERFLOW` churn

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

48.3.3. Compiler configuration, Build capture, and Compiler Integration Toolkit (CIT) 2019.06

This section lists new features, bug fixes, and known issues related to Coverity-supported compilers (including configuration), and the Compiler Integration Toolkit (CIT).

48.3.3.1. Important Compiler Integration Toolkit (CIT) information

There were several deprecations and EOLs for Compiler Integration Toolkit (CIT) this release:

- Dropped support for the `--coverity_source_<language>` switches to `cov-translate`. (CMPCCPP-8306)
- Dropped support for AIX 6.1. (CMPG-3016)

48.3.3.2. New and changed features

- Added support for the Kyoto Microcomputer gcc version 6.4.0 compiler. (CMPCPP-6765)
- Added support for Sony PS4 SDK versions 5.0, 5.5, and 6.0. (CMPCPP-7072)
- Added support for the Keil ARM uVision 5.06 compiler. (CMPCPP-7474)
- Added support for the Kyoto Microcomputer Clang version 6.0.0 compiler. (CMPCPP-8127)
- `cov-emit` now supports `_Imaginary` types. (CMPCPP-8306)
- Added support for the Analog Devices SHARC 8.12.0.0 compiler. (CMPCPP-8331)
- Added support for the Analog Devices Blackfin 8.12.0.0 compiler. (CMPCPP-8332)
- Added support for the ARM Clang 6.3 compiler. (CMPCPP-8333)
- Added support for GCC 8.2 and 8.3. (CMPCPP-8478)
- Added support for LLVM Clang 8. (CMPCPP-8567)
- Added support for the MetaWare ccac P-2019.03 compiler. (CMPCPP-8615)
- Added support for capturing and analyzing C# projects on Mac OS X that use the Unity Roslyn compiler. Analysis support is limited to quality checkers only. (CMPCSH-992)
- Added support for .NET Core 2.2. (CMPCSH-1043)
- Undeprecated support for GNU GCC and G++ compilers on macOS. Coverity will continue to support GNU compilers on macOS. (CMPG-2987)
- Added support for Fortran 18 standard. (CMPG-2988)

48.3.3.3. Bug fixes

The following bugs were fixed for compilers and the Compiler Integration Toolkit (CIT) for Coverity Analysis analysis in 2019.06:

CAP-1418

The `cov-build` command can now successfully capture OJDeploy-based builds.

CAP-1433

Fixed an issue involving `EXECUTING` lines that were elided in the `build-log.txt` file for commands executed with the `execvp()` system call.

CMPCPP-6342

Corrected an issue with xref generation that resulted in an Expression: `!isNull() && "Cannot retrieve a NULL type pointer"` assertion failure in `cov-internal-emit-clang` when the `__underlying_type` builtin type trait was used with a dependent type.

CMPCPP-7132

Fixed an assertion in `cov-emit` that could occur when an xvalue expression appeared within parentheses.

CMPCPP-7622

`cov-emit` now accepts inline namespaces declared with the `__inline` keyword.

CMPCPP-8019

Fixed an issue in `cov-emit` when emulating GCC where an enumeration type with no explicit underlying type was incorrectly given a signed underlying type.

CMPCPP-8357

Fixed an issue where using compiler intrinsics would cause false positive defects in MISRA-C 2012 Rule 8.2.

CMPCPP-8427

Fixed a suppressible assertion failure that occurred when initializing classes and structures with multiple inheritances combined with regular class inheritances.

CMPCPP-8477

`cov-emit` now allows aggregate initialization of objects with SIMD vector types.

CMPCPP-8493

Fixed a stack overflow in `cov-emit` when using the `--emit_complementary_info` option after being prompted by a long macro definition.

CMPCPP-8522

Fixed an issue where the C/C++ source type was incorrectly translated for the Green Hills ARM compiler.

CMPCPP-8536

Corrected an issue with Clang compilers that resulted in a "decl is part of a template" assertion failure error message and TU loss. This issue occurred when a build was captured with support enabled for compliance checkers.

CMPCPP-8561

Fixed an issue where `cov-build` would not recognize the `-tcf_core_config` option for the MetaWare ccac compiler.

CMPCPP-8562, CMPCPP-8605

A number of performance improvements were made for `--emit_complementary_info` and `--emit_referenced_types`.

CMPCPP-8569

Diagnostics performance has been improved.

CMPCPP-8616

Compilations with many parse warnings can be slow. This has been fixed.

CMPCPP-8626

Fixed an assertion in `cov-emit` error recovery that could occur when emitting referenced types.

CMPCPP-8654

Macro-related performance issues have been fixed.

CMPCPP-8669

Fixed an internal error issued by `cov-emit` when a template with a template-dependent base class was encountered while emulating the GCC `-fms-extensions` switch.

CMPCPP-8709

Fixed an issue that could cause a crash with the following message: "TU <N1> and <N2> both have primary source file <file>".

CMPCPP-8710

Fixed an issue where some header files for the MetaWare ccac compiler could not be found.

CMPCSH-642

Fixed an issue where `cov-emit-cs` would crash if a function definition was used in an external alias.

CMPCSH-682

Addressed an issue in the C# frontend where error recovery would result in the following assertion error message: "assertion failed: Cannot find class".

CMPCSH-1097

Fixed several (minor) issues with the Visual Basic compiler switch table.

CMPJ-1128

Fixed a stack overflow in the `cov-emit-java` frontend for large binary operation expressions with thousands of sub expressions.

CMPSWIFT-212, CMPSWIFT-236

`cov-emit-swift` is now more fault tolerant to unsupported Swift versions and language constructs.

CMPSWIFT-251

Fixed an issue where the `enable-batch-mode` and `pch-output-dir` Swift native compiler options were not recognized.

CMPVB-60

Addressed an issue in the Visual Basic FE where an `Optional DateTime` parameter would cause the FE to crash.

48.3.3.4. Known issues and solutions

CAP-1176, 97630

`cov-build --instrument` has a known issue when running the `xdcmake.exe` tool of Visual Studio 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 either:

- Modify the build such that `xdcmake.exe` is run from a 64-bit process.
- Ignore the `xdcmake.exe` process by adding `--capture-ignore xdcmake.exe` to your `cov-build` invocation.

CMPSCA-187

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

CMPJS-286, 95651

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 used with some frameworks.

CMPJ-368, 65669, 65721

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 `LANG` or `LC_CTYPE` environment variables.

48.3.4. Coverity Analysis 2019.06 Commands

48.3.4.1. Important information about commands related to the build and capture process

This section provides updates about `cov-build` and related commands, including capture, emit, and translate commands.

48.3.4.1.1. New and changed features

There were no new and changed features added for commands related to the build and capture process (including emit and translate commands) in 2019.06.

48.3.4.1.2. Bug fixes

There was one bug fixed for build and capture-related commands (including emit and translate commands) in 2019.06:

BLC-366

Fixed an issue where `cov-capture` could intermittently hang on macOS.

48.3.4.1.3. Known issues and solutions

Build-related commands have the following known issues and solutions:

CAP-812, 64428

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 hang occurs, 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 either:

- Uninstall KB2919355

OR

- Add the `--instrument` flag to your `cov-build` invocation:

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

CMPCPP-4764, 72964

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.

SAT-12174, 62745

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.

CAP-332, 26881, 38175

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

Error message:

```
[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
```

48.3.4.2. Commands related to analysis

This section lists new features, bug fixes, and known issues for `cov-analyze` and related commands.

48.3.4.2.1. New and changed features

There were no new features added or changed for commands related to the analysis process in 2019.06.

48.3.4.2.2. Bug Fixes

There were no bug fixes for analysis-related commands in 2019.06.

48.3.4.2.3. Known issues and solutions in 2019.06

Analysis-related commands have the following known issues and solutions:

CMPG-1741, 70845, 71216

The `cov-run-desktop` command sometimes fails on large Java compilations, potentially causing emit database corruption on Windows platforms. This can manifest as a `cov-analyze` crash. More commonly, `cov-emit-java` itself will fail with access violation crashes or errors concerning a failure to acquire a lock. These will appear in `cov-run-desktop-log.txt`. If this issue occurs, you can work around it by specifying `-j 1` with `cov-run-desktop`.

48.3.4.3. Commands related to Test Advisor

This section lists new features for Test Advisor.

48.3.4.3.1. Bug fixes in 2019.06

There were no bugs fixed for Test Advisor-related commands in this release.

48.3.5. Coverity Wizard 2019.06

This section lists new features, bug fixes, and known issues related to Coverity Wizard.

48.3.5.1. Important Coverity Wizard Information

There was one EOL this release:

- Dropped support for Microsoft Team Foundation Server (TFS) 2010. (PRD-11763)

48.3.5.2. New and changed features

Coverity Wizard has the following new and changed feature in 2019.06:

- Added support for Microsoft Azure DevOps Server 2019. (PRD-11751)

48.3.5.3. Bug fixes

The following bug was fixed for Coverity Wizard in 2019.06:

PRD-11657

Fixed Java filesystem capture in `cov-wizard` and Coverity plugins to capture all Java source files.

48.3.5.4. Known issues and solutions

Coverity Wizard has the following known issues in version 2019.06:

PRD-11727

`cov-wizard` might not successfully emit Java with the default version that is installed in Ubuntu 18.04. (For more information, 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-9245, 90621

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

PRD-9208, 90489

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-8453, 83450

When using a self-signed certificate, if the user chooses not to trust a certificate, they might be prompted multiple times in a row (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-8227, 82196

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-7595, 77742

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.

PRD-6832, 70361

The guided policy creation wizard "Documentation" link fails to open properly on Linux. Open the *Coverity Wizard 2020.12 User Guide* separately to view this documentation.

PRD-6760, 69815

The *Guided Test Advisor Policy Creation Wizard* uses Java regex validation instead of the Perl regex validation that Coverity Analysis Test Advisor users. 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-5770, 59676

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-5387, 54143

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

PRD-5290, 53367

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.

48.3.6. Test Advisor 2019.06

Coverity Test Advisor is a component of the Coverity Analysis installation package.

48.3.6.1. Important Test Advisor information

There have been several deprecations and EOLs this release:

- Dropped support for Git 1.4-1.7. (TADE-1984)
- Dropped support for Mercurial 1.0-3.0. (TADE-1985)
- Deprecated support for Git 1.8-2.1. (TADE-1987)
- Dropped support for Perforce 2014.2 and 2015.1. (TADE-1992)
- Deprecated support for Perforce 2015.2. (TADE-1992)
- Dropped support for Accurev 6.2. (TADE-1996)
- Deprecated support for Mercurial 3.1 and 3.2. (TADE-1996)
- Dropped support for Team Foundation Server (TFS) 2010. (TADE-1999)

48.3.6.2. New and changed features

Test Advisor has several new or changed features in 2019.06:

- Added support for Azure DevOps Server 2019 to our SCM tools. It can be accessed by passing the `--scm ads` argument. (TADE-1986)
- Added support for versions of Git up to 2.21. (TADE-1987)
- Added support for Perforce 2017.2, 2018.1, and 2018.2. (TADE-1992)

48.3.6.3. Bug fixes

The following bug was fixed for Test Advisor in 2019.06:

TADE-1111

`cov-import-scm` and `cov-extract-scm` now accept the `--scm-tool`, `--scm-tool-arg`, `--scm-command-arg`, and `--scm-project-root` options as synonyms for the `--tool`, `--tool-arg`, `--command-arg`, and `--project-root` options. This makes their interfaces consistent with our other tools using SCM arguments.

48.3.6.4. Known issues and solutions

Test Advisor has the following known issues and solutions in 2019.06:

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.

48.3.7. Dynamic Analysis 2019.06

Dynamic Analysis is a component of the Coverity Analysis installation package.

48.3.7.1. Important Dynamic Analysis information

There has been one deprecation this release:

- Deprecated support for JDK for macOS 1.6. (JDA-1090)

48.3.7.2. Known issues and solutions

Dynamic Analysis has the following known issues and solutions in 2019.06:

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.

JDA-681, 20788

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

JDA-694, 21417

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, then Dynamic Analysis reports races on neither collections nor arrays.

JDA-720, 22148

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

48.3.8. Architecture Analysis 2019.06

Coverity Architecture Analysis is a component of the Coverity Analysis installation package.

48.3.8.1. Important Architecture Analysis information

Architecture Analysis has the following EOL in 2019.06:

- Dropped support for Architecture Analysis on Windows 32-bit systems. (COVP-2127)

48.3.9. Extend SDK 2019.06

Coverity Extend Software Development Kit is a component of the Coverity Analysis installation package.

48.3.9.1. Known issues and solutions

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.

48.4. Coverity Desktop 2019.06

The Coverity Desktop plug-in is available for various platforms from the Coverity Connect *Downloads* menu.

48.4.1. Important information

There were several deprecations and EOLs for Coverity Desktop this release:

- Dropped support for Microsoft Team Foundation Server (TFS) 2010. (PRD-10649; PRD-10651)
- Dropped support for the `--preview` option. (PRD-11714)
- Dropped support for Android Studio 2.2. (PRD-11756)
- Dropped support for IntelliJ 2016.x. (PRD-11780)
- Dropped support for RubyMine, WebStorm, and PyCharm. (PRD-11780)
- Deprecated support for Eclipse 4.6. (PRD-11781)

48.4.2. Coverity Desktop for Android Studio in 2019.06

48.4.2.1. New and changed features

Coverity Desktop for Android Studio has the following new and changed feature in 2019.06:

- Added Android plugin support for Gradle 3.1.4. (PRD-11803)

48.4.2.2. Bug fixes

The following bug was fixed for Coverity Desktop for Android Studio in 2019.06:

PRD-6585

Fixed an issue where entering long strings to run build tests would cause UI issues in `cov-wizard`.

48.4.2.3. Known issues and solutions

Coverity Desktop for Android Studio has no known issues in version 2019.06.

48.4.3. Coverity Desktop for Eclipse in 2019.06

48.4.3.1. New and changed features

Coverity Desktop for Eclipse has the following new and changed features in 2019.06:

- Added support for Eclipse 2019.03. (PRD-11716)
- The Fast Desktop IDEs now support sorting and filtering by MISRA Category level. (PRD-10707; PRD-11740)
- Added support for Microsoft Azure DevOps Server (ADS) 2019. (PRD-11749)
- Added support for PhpStorm 2019.1, WebStorm 2019.1, and RubyMine 2019.1. (PRD-11766)

48.4.3.2. Bug fixes

The following bug was fixed for Coverity Desktop for Eclipse in 2019.06:

PRD-10661

Fixed an issue where using dynamic variables for the working directory in the Eclipse plugin would cause the build to fail. Prior to the fix, the dynamic variables were not being properly evaluated when a custom build command was used.

48.4.3.3. Known issues and solutions

Coverity Desktop for Eclipse has the following known issues in version 2019.06:

PRD-10694

For OXS 10.14 users with JDK-8136913 installed, using the `hostname_regex` in the `coverity.conf` file causes 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 may 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/`.

48.4.4. Coverity Desktop for Microsoft Visual Studio in 2019.06

48.4.4.1. New and changed features

Coverity Desktop for Microsoft Visual Studio has the following new and changed feature in 2019.06:

- Added support for Microsoft Azure DevOps Server (ADS) 2019. (PRD-11751)

48.4.4.2. Bug fixes

Coverity Desktop for Microsoft Visual Studio has no bug fixes in 2019.06.

48.4.4.3. Known issues and solutions

Coverity Desktop for Visual Studio has no known issues in version 2019.06.

48.4.5. Coverity Desktop for IntelliJ IDEA in 2019.06

48.4.5.1. New and changed features

Coverity Desktop for IntelliJ IDEA has the following new and changed feature(s) in 2019.06:

- Added support for IntelliJ 2019.1. (PRD-11743)

 **Note**

Support for IntelliJ and PyCharm 2019.1 JDK11 has been pushed out to 2019.09. As of 2019.06, we only support IDEA JDK8 products.

- Added support for Microsoft Azure DevOps Server (ADS) 2019. (PRD-11749; PRD-11750)
- Added support for CLion 2019.1. (PRD-11764)

48.4.5.2. Bug fixes

Coverity Desktop for IntelliJ has no new bug fixes in 2019.06.

48.4.5.3. Known issues and solutions

Coverity Desktop for IntelliJ IDEA has the following known issues in version 2019.06:

PRD-7453, 76907

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, 80573

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

PRD-7991, 80599

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, 80693

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

PRD-8397, 83106

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

PRD-8042, 80698

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.

PRD-10076, 105052

When using whole program checkers in IntelliJ, a warning about missing class files might be seen 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, 119444

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.

48.5. Coverity Report Generators 2019.06

The Coverity Report Generators' installer can be downloaded from the downloads page in Coverity Connect.

48.5.1. Important Coverity Report Generators information

Coverity Report Generators have no deprecations or EOLs in 2019.06.

48.5.1.1. New and changed features

There were no new or changed features added for the Coverity Report Generators in 2019.06.

48.5.1.2. Bug fixes

The following bug was fixed for the Coverity Report Generators in 2019.06:

INS-2636

During Coverity Reports installation, users are no longer prompted to create file associations for the `.covsr`, `.covmr`, `.cover`, or `.sir` file extensions because these extensions are no longer used by Coverity Reports.

48.5.1.3. Known issues and solutions

Coverity Report Generators have the following known issues and solutions:

RG-1128

For ATP-based systems, you may 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 do encounter this error message, please install these missing libraries: `libgl1`, `libgl1-mesa-dri`, and `libgl1-mesa-glx`.

48.6. Coverity Documentation 2019.06

The following new documents and changes were made in 2019.06:

PRD-11725

Coverity Plugin, Test Advisor and Wizard support for TFS 2010 has been discontinued as of 2019.06.

RG-1113

A new chapter has been added that describes how you must now configure reports using a `.yaml` configuration file.

SAT-26233

The CodeXM documents are all now in a consistent HTML format.

They have been edited to improve their consistency and clarity.

Chapter 49. Coverity 2019.03-12 Release Notes

Table of Contents

49.1. Coverity Platform bug fixes	186
---	-----

49.1. Coverity Platform bug fixes

IM-24880

Fixed WS API which ignores server domain when querying for a group with same name in multiple LDAP servers

Chapter 50. Coverity 2019.03-11 Release Notes

Table of Contents

50.1. Coverity Analysis bug fixes	187
---	-----

50.1. Coverity Analysis bug fixes

CMPCPP-9822

A number of bugs preventing Boost 1.68 from compiling on Solaris have been fixed.

Chapter 51. Coverity 2019.03-10 Release Notes

Table of Contents

51.1. Coverity Analysis bug fixes	188
---	-----

51.1. Coverity Analysis bug fixes

CMPCPP-9696

Diagnostics in system headers are normally suppressed. In preprocessed and PCH files this was not happening. This has been corrected.

Chapter 52. Coverity 2019.03-9 Release Notes

Table of Contents

52.1. Coverity Platform bug fixes	189
---	-----

52.1. Coverity Platform bug fixes

IM-24163

Restore the "pretty print" functionality to the Commit Preview Report JSON output.

Chapter 53. Coverity 2019.03-7 Release Notes

Table of Contents

53.1. Coverity Analysis bug fixes	190
---	-----

53.1. Coverity Analysis bug fixes

CMPCPP-9028

Fixed a stack overflow in `cov-emit` when using the `--emit_complementary_info` option after being prompted by a long macro definition.

Chapter 54. Coverity 2019.03-6 Release Notes

Table of Contents

54.1. Coverity Analysis bug fixes	191
---	-----

54.1. Coverity Analysis bug fixes

SAT-30309

Fixed a source of `OVERRUN` false positives involving the use of the `strncpy_s` function.

SAT-30310

Fixed a source of `OVERRUN` false positives with certain implementations of the `va_start` function.

SAT-30232

Fixed a class of `SINGLETON_RACE` false positives that occurred when Coverity Analysis did not properly recognize a `RequestScope` annotation.

SAT-30237

Fixed an issue that would cause `PRINTF_ARGS` false positives when using the `%s` specifier in a call to the `wprintf` function on Windows.

Chapter 55. Coverity 2019.03-5 Release Notes

Table of Contents

55.1. Coverity Analysis bug fixes	192
---	-----

55.1. Coverity Analysis bug fixes

CMPCPP-8802

Improved Coverity Analysis with Clang compilers, so that false positives are avoided when an object is assigned a temporary value and that object also contains a field that affects a condition.

SAT-29761

Updated `PRINTF_ARGS` to understand the Microsoft-specific `I32`, `I64`, and `I` size attributes.

SAT-29762

Fixed an error where invoking the version of `swprintf` that does not take a size argument resulted in `OVERRUN` false positives.

Chapter 56. Coverity 2019.03-4 Release Notes

Table of Contents

56.1. Coverity Platform bug fixes	193
---	-----

56.1. Coverity Platform bug fixes

IM-23819

Fixed occasional commit failures that occurred during high concurrency commit scenarios.

Chapter 57. Coverity 2019.03-3 Release Notes

Table of Contents

57.1. Coverity Analysis bug fixes	194
---	-----

57.1. Coverity Analysis bug fixes

CMPCPP-8754

Fixed an issue where usage of compiler intrinsics caused false positive defects with MISRA-C 2012 Rule 8.2.

Chapter 58. Coverity 2019.03-2 Release Notes

Table of Contents

58.1. Coverity Analysis bug fixes	195
58.2. Report Generators bug fixes	195

58.1. Coverity Analysis bug fixes

SATW-2993

Fixed a MISRA C-2012 Rule 14.3 false positive, involving an explicit casting that was handled incorrectly.

SAT-28810

This introduces a pragma-based mechanism, allowing inline source code annotations to suppress reporting of defects and false positives that are found in C and C++ code. The fields that are available in the pragma support compliance deviation use the cases by generating a (CSV) file. The CVS file that is generated lists all the suppressed defects and false positives.

58.2. Report Generators bug fixes

RG-1199

The Report Generators now support a non-empty root context path. The context path for the Coverity Connect URL can also be used to generate a report.

Chapter 59. Coverity 2019.03-1 Release Notes

Table of Contents

59.1. Coverity Analysis bug fixes	196
---	-----

59.1. Coverity Analysis bug fixes

BLC-534

Fixed an issue where `cov-capture` failed to capture some JSP files in Maven and Gradle projects.

SATW-2983

Fixed a false positive of MISRA C-2012 Rule 14.3, involving an explicit cast that was handled incorrectly.

Chapter 60. Coverity 2019.03 Release Notes

Table of Contents

60.1. Important information for 2019.03	197
60.2. Coverity Platform 2019.03	198
60.3. Coverity Analysis 2019.03	202
60.4. Coverity Desktop 2019.03	221
60.5. Coverity Report Generators 2019.03	224
60.6. Coverity Documentation 2019.03	225

60.1. Important information for 2019.03

Due to a change in our bug tracking system, items are now identified by two bug numbers:

- One reflecting the identity of the bug in our **old** bug tracking system, formatted like this: XXXXXX. (For example, 374568.)
- One reflecting the identity of the bug in our **new** bug tracking system, formatted like this: CODE-XXXXX. (For example, IM-22788.)

 **Note**

Bugs with only a CODE-XXXXX number do not have an old number.

60.1.1. Deprecated and End-of-Life (EOL) Products in Coverity 2019.03

Support for the following products, features, platforms, and third-party tools is classified as deprecated or end-of-life as of the Coverity 2019.03 release.

60.1.1.1. Deprecated Products

Support for the following products and features is deprecated as of the Coverity 2019.03 release.

Table 60.1. Deprecated products

Product	Comments
ARM ADS 1.11-1.2 C/C++ compiler support	Supported Compilers: Coverity Analysis for C/C++
ARM RVDS 2.0–4.1 C/C++ compiler support	Supported Compilers: Coverity Analysis for C/C++
Git 1.4-1.7 support	Coverity Test Advisor Support SCM Systems
GNU GCC and G++ 2.7.2–8.1.0 compilers on Mac OS support	Supported Compilers: Coverity Analysis for C/C++
IAR Embedded Workbench C/C++ 7.30B–8.10 compiler for the 8051 processor support	Supported Compilers: Coverity Analysis for C/C++
macOS 10.12 support	Supported Compilers: Coverity Analysis for C/C++

Product	Comments
Mercurial 1.0-3.0 support	Coverity Test Advisor Support SCM Systems
Microsoft Visual C++ 6 and 2003 compiler support	Supported Compilers: Coverity Analysis for C/C++
Solaris 10 support	Supported Compilers: Coverity Analysis for C/C++
STMicroelectronics GNU C/C++ 2.3.1 and 4.1.1 compiler support	Supported Compilers: Coverity Analysis for C/C++
Swift compiler running in Xcode 9.0-9.4 support	Supported Compilers: Coverity Analysis for Swift

60.1.1.2. End-of-Life Products

Support for the following products and features is dropped in the Coverity 2019.03 release.

Table 60.2. End-of-Life Products

Product	Comments
Android Jack compiler support	Supported Compilers: Coverity Analysis for C/C++
HP-UX platform support	Supported Compilers: Coverity Analysis for C/C++
Java support on Solaris SPARC	Supported Compilers: Coverity Analysis for C/C++
macOS 10.11 support	Supported Compilers: Coverity Analysis for C/C++
NetBSD 6.1 and earlier support	Supported Platforms for Coverity Analysis
QNX Momentics 5.0 support	Supported Platforms for Coverity Analysis
Xbox 360 compiler support	Supported Compilers: Coverity Analysis for C/C++

60.2. Coverity Platform 2019.03

This section provides release notes for Coverity Platform components.

60.2.1. Coverity Connect 2019.03

Coverity Connect is a component of the Coverity Platform installation package.

60.2.1.1. Important Coverity Connect Information

There has been one EOL for Coverity Connect this release:

- Triage store merging and copying are no longer supported. (IM-23489)

60.2.1.1.1. New and changed features

The following new and changed features have been added for Coverity Connect this release:

- Coverity Connect now supports the Code Sight plugin for IntelliJ, Visual Studio, and Eclipse. The following features are supported:
 - Coverity Connect and Code Sight can share triage lists

- Code Sight can leverage available scan summaries from Coverity Connect to augment the accuracy of results found on the desktop

60.2.1.1.2. Bug fixes

The following bugs are fixed for Coverity Connect:

IM-22132, 115871

Triage store export now provides JSON (gzip) files and import now optionally accepts JSON (gzip) files.

IM-22216, 116780

Enhanced the export functionality by filtering out disabled users when the "Do not show disabled users" option was selected in the Users panel.

IM-22662

Fixed a null pointer exception in the `updateComponentMap` API when a null value was passed in components.

IM-22842

Fixed a concurrency bug in LDAP integration code, which might have caused login issues for LDAP users.

IM-22916

The Coverity Platform administrator can now unlock all temporarily locked user accounts. An *Unlock User Accounts* button has been added to the *Configuration* → *System* → *Authentication and Sign In* → *Sign In Log* screen. This button unlocks accounts of users who are temporarily locked out as a result of repeated unsuccessful login attempts. It does not unlock user accounts that the administrator has explicitly locked.

IM-22994

Triage store export functionality now produces compressed (gzip) JSON data. Triage store import functionality now also accepts compressed JSON data.

IM-23245

Improved the stability of Coverity Connect for Windows operating systems by changing the library that Coverity Connect uses to gather system information.

IM-23296

`cov-manage-im` now accepts values for the `https_proxy` environment variable when it contains an http scheme.

IM-23310

The bundled JRE was updated, and now comes from OpenJDK 11. As a result, Coverity Connect is now using enhanced Java garbage collector (GC) logging and low overhead Java flight recording (JFR) is now always enabled.

The GC logs are now named `gc_%t_%p.log`, where `%t` is the process start time and `%p` is the process identifier (PID). Consequently, restarting Coverity Connect does not overwrite the previous GC logs (like it previously did). While the size of GC logs is limited per process, the total size of all

GC logs (including the logs from previous runs) is not limited, and the logs from older executions need to be managed by the Coverity Connect administrator because Coverity Connect does not automatically remove them.

Similarly to the GC logs, the size of JFR output is limited per process, but recordings from previous runs (the file format is `hotspot-pid-%p-id-1-%t.jfr`) need to be managed by the Coverity Connect administrator because Coverity Connect does not automatically remove them.

IM-23338

Improved the efficiency of the Policy Manager's Trend Report job, especially for data sets involving large amounts of function data.

IM-23388

Fixed a security issue so that general users are no longer able to improperly access authentication key files.

IM-23491

Recomputed the recent Policy Manager trend data after a configuration change had invalidated the existing data. Prior to this fix, Policy Manager trend charts did not display data before the date of the configuration change.

IM-23516

Fixed a bug in the *Purge Snapshot Details* functionality that was causing Issue Occurrences to be lost for some Issues for snapshots that were not directly purged. New snapshots would still display the correct Occurrences, but existing snapshots would still be missing data.

60.2.1.1.3. Known issues and solutions

Coverity Connect has the following known issues:

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.

CPU-38, 82579

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.

CPU-17, 80045

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.

IM-16076, 67748

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, 75559

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

IM-17660, 75263

An error occurs when a custom role is created using a multi-word rolename that is the same as a built-in rolename, even if there are case differences between the two rolenames.

IM-18707, 82643

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, 82648

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, 84453

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, 89897

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, 89946

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.

INS-1274, 63454

Although the upgrade doc 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, fall back to backup-and-restore upgrade.

INS-1477, 73401

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 by specifying a max heap setting for `cov-im-daemon`.

INS-2133, 112939

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, 118662

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.

60.2.2. Coverity Policy Manager 2019.03

Coverity Policy Manager is a component of the Coverity Platform installation package.

There are no new or changed features, and no bug fixes or known issues for this release.

60.3. Coverity Analysis 2019.03

This section provides updates about Coverity Analysis components.

60.3.1. Important Coverity Analysis information

There have been several deprecations and EOLs this release:

- Coverity Analysis support for macOS 10.11 has been dropped. (COVP-2102)
- `__coverity_tainted_data_argument__`, `__coverity_tainted_data_return__`, `__coverity_tainted_string_argument__`, and `__coverity_tainted_string_return_content__` primitives have been deprecated. These primitives are replaced by `__coverity_mark_pointee_as_tainted__`, which allows specification of `taintType`. (SAT-28378)

60.3.1.1. New and changed features

Coverity Analysis has the following new and changed features:

- Added support for Windows Server 2019. (COVP-2087)
- Added support for NetBSD v7.1, 7.2, and 8.0. (COVP-2088)
- Added support for FreeBSD v11.2 and 12.0. (COVP-2091)
- Added support for `IsNothing` as a null check in Visual Basic. (SAT-26396, 121688)
- Added TypeScript support for the `COPY_PASTE_ERRORS` and `DEADCODE` quality checkers. (SAT-26374)
- Added TypeScript support for the following quality checkers: `CONSTANT_EXPRESSION_RESULT`, `IDENTICAL_BRANCHES`, `IDENTIFIER_TYPO`, `MISSING_BREAK`, `NESTING_INDENT_MISMATCH`, `STRAY_SEMICOLON`, `UNEXPECTED_CONTROL_FLOW`, `UNINTENDED_GLOBAL`, and `UNREACHABLE`. (SAT-27242)
- Added support for audit taint analysis in Visual Basic sources. When analysis is run with the `--enable-audit-mode` option, additional audit-mode defects are reported. (SAT-27998)
- Added a built-in model for the linux `_raw_spin_trylock()` function. (SAT-28281)
- A new command, `cov-security-da`, runs the dynamic analysis check for security issues that was previously performed by `cov-analyze` and `cov-capture`. These commands now invoke `cov-security-da`, instead.

For `cov-analyze` and `cov-capture`, the option `--no-security-da` disables the invocation of `cov-security-da`. (SAT-28393)

- Added support for the Mustache template engine to JavaScript Template Dynamic Analysis. (SAT-28568)
- Added support for the Nunjucks template engine to JavaScript Template Dynamic Analysis. (SAT-28567)
- Added a new option to `cov-run-desktop`: The `--connect-timeout` option allows users to change (in seconds) the connection timeout to the given duration. (SAT-28563)

60.3.1.2. Bug fixes

INS-2444

The Coverity installers now bundle and run OpenJDK (instead of Oracle JRE).

INS-2519

On 32-bit Windows platforms, the `cov-install-updates` rollback command failed if the path exceeded 260 characters in length. This limitation has been removed.

INS-2520

Installers for Solaris x86_64 and Solaris SPARC are now only available as archives, not as executable installers. For more information, see *Chapter 2.4 Using an archive file to install Coverity Analysis* in the Coverity Installation and Deployment Guide.

INS-2561

Due to the removal of Java analysis support, Coverity Analysis installations for Solaris will no longer contain the utilities `cov-manage-im`, `cov-copy-overflow-triage`, `cov-start-da-broker`, and `cov-stop-da-broker`.

SAT-26153

Fixed an issue where analyzing large codebases on Windows, especially on some versions such as Windows server 2016, could result in significant slowdowns.

SAT-27094

Fixed a bug for `TAINTED_SCALAR` where a sink was not generated for a struct field when it was used as a loop bound.

SAT-27112

Fixed the configuration for the `--dc-config` option, so that users can specify a function's name using the unmangled name printed by `cov-manage-emit`.

SAT-27386

The `--enable-audit-mode` option is now accepted by `cov-run-desktop` and passed to `cov-analyze`.

SAT-27581

Checker-specific trust options now override the `--distrust-all` option.

SAT-27721

Fixed web application security false negatives involving values that were returned by Django `request.GET.get` and the `%` string operator.

SAT-27808

Fixed an analysis crash involving an "issueType = BUFFER_SIZE_WARNING" error message.

SAT-27811

Fixed a source of false positives involving `C99_Bool` and `?:` expressions.

SAT-27962; SAT-3244, 18204

Fixed an error that caused the `RESOURCE_LEAK no_vararg_leak` option to be ineffective.

SAT-28096

Fixed an issue where `cov-analyze --append` would fail to report the set of enabled checkers. With the fix, the set of enabled checkers for `cov-analyze` is now properly reported on when the `--append` option is used with any tool.

SAT-28115

The `cov-dotnet-aot` command has been removed, as it is no longer necessary. The `.NET` bytecode that was previously affected by `cov-dotnet-aot` is now compiled ahead of time (as part of the product).

SAT-28180

Fixed a `cov-analyze` recoverable error that occurred with the `UNLOGGED_SECURITY_EXCEPTION` checker on some C# throw statements.

SAT-28211

Fixed a bug where analysis erroneously considered `System.String.Split` methods as dereferencing their separator parameter.

SAT-28263

A new `--checker-option` to the `cov-make-library` command enables users to set options when invoking this command. The options set here are passed along to `cov-analyze`.

SAT-28288

Fixed an error that would prevent `FORWARD_NULL` reports when a function initializes all the members of a struct to 0 or `NULL`.

SAT-28338

Fixed the issue where `for...of` loops with destructuring iterators caused false positive `NO_EFFECT` defects.

SAT-28339

Improved Coverity Analysis to report on `TAINTED_SCALAR` defects for indices passed to `std::vector`.

SAT-28386

Fixed a crash in `cov-analyze` dataflow analysis which was caused by long string concatenation.

SAT-28393

As of the 2019.03 release, a small amount of work that used to run during `cov-analyze` now runs during `cov-build` and `cov-capture`. Users may see slightly increased runtime for `cov-build` and `cov-capture`, and slightly decreased runtime for `cov-analyze`.

To revert back to the old behavior, you can run `cov-build` and `cov-capture` with the `--no-security-da` option and then run the `cov-security-da` command before running `cov-analyze`.

SAT-28446; SAT-28347

Fixed an analysis crash involving `LENGTH_FUNCTION` when using the `--enable-fnptr` option.

SAT-28447

The Fortran analysis program was incorrectly substituting backslash characters when composing include file paths with included filenames. The program has been corrected to use only forward slashes in the absolute filenames that it produces.

SAT-28476

A logic error in `cov-run-fortran` caused arguments that do not look like source files to be removed from the argument list, causing include file paths to be elided. This problem has been corrected. Include file paths that are passed with the `-I` option now work as intended.

Note that the include file paths are part of the input for the Fortran analysis tool, so they must appear after the double dash that separates control options from analysis options.

On Windows platforms, the colon (`:`) is no longer recognized as a path delimiter. Multiple paths following the `-I` option should be delimited with semicolons or commas. On Linux platforms, the comma (`,`) has been added as a path delimiter to support uniform handling of include paths across platforms.

SAT-28484

Fixed a source of false positives that occurred when the same condition was checked multiple times and it involved temporary objects.

SAT-28528

Fixed an issue that could cause `OVERRUN` false positives when using the `memcpy_s` function without a prototype.

SAT-28521

Fixed an issue that could cause false positives when a variable was modified using `*(type *)&var`.

SAT-28522; SAT-28507; SAT-28485

Fixed an issue that caused `OVERRUN` false positives when using `_TRUNCATE` with secure Windows API functions.

SAT-28524; SAT-28482

Fixed the overflow detection logic in the calculation of loop variable bounds to avoid `roundup_lower >= *new_lower` assertion failures in `intervalfpp.cpp`.

SAT-28719

Improved models for the Apache Commons Codec library. The `RISKY_CRYPTO` checker and other checkers may report more defects in code that utilizes this library.

SATW-1005

Fixed a false negative in MISRA C-2012 Rule 22.2 where a successful `realloc` function call was not processed as freeing its pointer argument.

SATW-2181, 102438

Fixed a false negative in CERT MEM34-C, where a successful `realloc` function call was not processed as freeing its pointer argument.

SATW-2184

Fixed a false negative in FIO34-C caused by integer sizes of different platforms.

SATW-2379

Fixed an issue that caused slow performance of the CERT DCL40-C checker.

SATW-2656

Fixed a false positive in CERT STR31-C.

SATW-2826

Fixed false positives in MISRA C-2012 Rule 6.1 involving explicitly signed and unsigned integer types. We've also improved the defect presentation.

SATW-2839

Improved the defect presentation of MISRA C-2012 Rule 21.2 to be clearer.

SATW-2855

Fixed false positives in MISRA C-2012 Rule 10.1, 10.3, 10.4 and 10.5 where an integer literal indicating a value of 0 or 1 was incorrectly recognized as a boolean essential type in function-like macros.

60.3.1.3. Known issues and solutions

COVP-1717, 81025

Older versions of FlexLM might produce the wrong hostid version. Please make sure you have installed a version of FlexLM which supports Windows 10.

INS-1694, 84234: Installing into an existing folder

Coverity Analysis cannot be installed into an existing empty folder. Please select a non-existing folder.

INS-1792, 89937

The Coverity Analysis installer fails when the installer path contains Japanese characters.

SAT-26530

On 64-bit Windows platforms, the length of the command string that can be passed to the Fortran syntax analyzer is limited (internally) to 32768 characters. If this limit is exceeded, `cov-run-fortran` fails and reports an "Argument list too long" error.

SAT-26651

Coverity Fortran Syntax Analysis fails with a memory access violation when run under Clear Linux 4.14-64. There is a possible incompatibility with the Fortran runtime library on the Clear Linux platform.

SAT-27581

When `--webapp-security-aggressiveness-level` is set to `high`, it has the effect of setting the `distrust_all` checker option for many checkers. In this case, trusting individual taints using `--trust-<taint-type>` options does not override the `distrust_all` checker option. Note that `--enable-audit-mode` sets `--webapp-security-aggressiveness-level=high` by default. This describes the current behavior. It might change in future releases and should not be relied upon.

60.3.2. Coverity Analysis checkers and user directives in 2019.03

The following sections describe new and updated features, bug fixes, and known issues for Coverity checkers and associated elements.

60.3.2.1. New and updated checkers and directives

The following table lists new checkers and the languages they support.

Checker	Languages
CONFIG.ANDROID_BACKUPS_ALLOWED	Android CodeXM
CONFIG.ANDROID_UNSAFE_MINSDKVERSION	Android CodeXM
CONFIG.ANDROID_OUTDATED_TARGETSDKVERSION	Android CodeXM
CONFIG.JSONWEBTOKEN_NON_EXPIRING_TOKEN	JavaScript
CONFIG.MYSQL_SSL_VERIFY_DISABLED	JavaScript
CONFIG.REQUEST_STRICT_SSL_DISABLED	JavaScript
CONFIG.SOCKETIO_MAXHTTPBUFFERSIZE_SET_TOO_LARGE	JavaScript
CONFIG.SOCKETIO_ORIGINS_ACCEPT_ALL	JavaScript
CONFIG.SEQUELIZE_ENABLED_LOGGING	JavaScript

The following table documents added language support for existing checkers.

Languages	Checkers	Checkers
TypeScript	COOKIE_INJECTION	REGEX_INJECTION
	CSS_INJECTION	SCRIPT_CODE_INJECTION
	DOM_XSS	SESSIONSTORAGE_MANIPULATION
	HEADER_INJECTION	SQLI

Languages	Checkers	Checkers
	LOCALSTORAGE_MANIPULATION	TEMPLATE_INJECTION
	PATH_MANIPULATION	URL_MANIPULATION

New and changed checkers

SAT-19970; SAT-20431, 95964

The `XML_EXTERNAL_ENTITY` checker reports more defects when data from the filesystem is distrusted.

SAT-26670

For the `PATH_MANIPULATION` checker, built-in sanitizer heuristics are now disabled when the `--enabled-audit-mode` option is present.

SAT-23064

The `NO_EFFECT` checker no longer reports on `0, <IIFE>` (the Javascript idiom), which is used to disambiguate IIFEs (Immediately Invoked Function Expressions) from function definitions.

SAT-27589; SAT-28201; SAT-28212

The following checkers have been updated as indicated. Note that the updates vary subtly between checkers. For example, although the four options mentioned are always the same four, in some cases the options are newly added to the checker, while in other cases they already existed but have added language support:

- `ANGULAR_EXPRESSION_INJECTION`

Added these new options: `trust_mobile_other_app:<boolean>`, `trust_mobile_other_privileged_app:<boolean>`, `trust_mobile_same_app:<boolean>`, and `trust_mobile_user_input:<boolean>`.

- `COOKIE_INJECTION`

Added TypeScript language support to the `COOKIE_INJECTION` checker.

Added these new options: `trust_mobile_other_app:<boolean>`, `trust_mobile_other_privileged_app:<boolean>`, `trust_mobile_same_app:<boolean>`, and `trust_mobile_user_input:<boolean>`.

- `CSS_INJECTION`

Added TypeScript language support to the `CSS_INJECTION` checker.

Added these new options: `trust_mobile_other_app:<boolean>`, `trust_mobile_other_privileged_app:<boolean>`, `trust_mobile_same_app:<boolean>`, and `trust_mobile_user_input:<boolean>`.

- `DOM_XSS`

Added TypeScript language support to the `DOM_XSS` checker.

Added these new options: `trust_mobile_other_app:<boolean>`,
`trust_mobile_other_privileged_app:<boolean>`,
`trust_mobile_same_app:<boolean>`, and `trust_mobile_user_input:<boolean>`.

- **HEADER_INJECTION**

Added TypeScript language support to the `HEADER_INJECTION` checker.

Added these new options for JavaScript and TypeScript only:

`trust_mobile_other_app:<boolean>`,
`trust_mobile_other_privileged_app:<boolean>`,
`trust_mobile_same_app:<boolean>`, and `trust_mobile_user_input:<boolean>`.

- **LOCALSTORAGE_MANIPULATION**

Added TypeScript language support to the `LOCALSTORAGE_MANIPULATION` checker.

Added these new options: `trust_mobile_other_app:<boolean>`,
`trust_mobile_other_privileged_app:<boolean>`,
`trust_mobile_same_app:<boolean>`, and `trust_mobile_user_input:<boolean>`.

- **PATH_MANIPULATION**

Added TypeScript language support to the `PATH_MANIPULATION` checker.

Added JavaScript, PHP, Python, and TypeScript language support

to these options: `trust_mobile_other_app:<boolean>`,
`trust_mobile_other_privileged_app:<boolean>`,
`trust_mobile_same_app:<boolean>`, and `trust_mobile_user_input:<boolean>`.

- **REGEX_INJECTION**

Added TypeScript language support to the `REGEX_INJECTION` checker.

Added JavaScript and TypeScript language support to

these options: `trust_mobile_other_app:<boolean>`,
`trust_mobile_other_privileged_app:<boolean>`,
`trust_mobile_same_app:<boolean>`, and `trust_mobile_user_input:<boolean>`.

- **SCRIPT_CODE_INJECTION**

Added TypeScript language support to the `SCRIPT_CODE_INJECTION` checker.

Added these new options for JavaScript, PHP, Python,

TypeScript only: `trust_mobile_other_app:<boolean>`,
`trust_mobile_other_privileged_app:<boolean>`,
`trust_mobile_same_app:<boolean>`, and `trust_mobile_user_input:<boolean>`.

- **SESSIONSTORAGE_MANIPULATION**

Added TypeScript language support to the `SESSIONSTORAGE_MANIPULATION` checker.

Added these new options: `trust_mobile_other_app:<boolean>`,
`trust_mobile_other_privileged_app:<boolean>`,
`trust_mobile_same_app:<boolean>`, and `trust_mobile_user_input:<boolean>`.

- `SQLI`

Added TypeScript language support to the `SQLI` checker.

Added JavaScript, PHP, Python, and TypeScript language support to these options: `trust_mobile_other_app:<boolean>`,
`trust_mobile_other_privileged_app:<boolean>`,
`trust_mobile_same_app:<boolean>`, and `trust_mobile_user_input:<boolean>`.

- `TEMPLATE_INJECTION`

Added TypeScript language support to the `TEMPLATE_INJECTION` checker.

Added these new options: `trust_mobile_other_app:<boolean>`,
`trust_mobile_other_privileged_app:<boolean>`,
`trust_mobile_same_app:<boolean>`, and `trust_mobile_user_input:<boolean>`.

- `URL_MANIPULATION`

Added TypeScript language support to the `URL_MANIPULATION` checker.

Added these new options: `trust_mobile_other_app:<boolean>`,
`trust_mobile_other_privileged_app:<boolean>`,
`trust_mobile_same_app:<boolean>`, and `trust_mobile_user_input:<boolean>`.

SAT-27648

The `CONFIG.SEQUELIZE_ENABLED_LOGGING` checker finds cases where a Sequelize connection is created with logging enabled. In these cases, the SQL queries would be logged to the console and may leak sensitive data.

SAT-27916

The new `CONFIG.SOCKETIO_ORIGINS_ACCEPT_ALL` checker finds cases where a `socket.io` instance is configured to allow connections from any origin.

SAT-27977

The new `CONFIG.SOCKETIO_MAXHTTPBUFFERSIZE_SET_TOO_LARGE` checker finds cases where a `Socket.IO` server is created with a buffer size that is too large.

SAT-28042

The new `CONFIG.MYSQL_SSL_VERIFY_DISABLED` checker finds cases where a MySQL connection is configured to not verify the validity of the SSL certificate and accepts invalid certificates.

SAT-28053

The new `CONFIG.JSONWEBTOKEN_NON_EXPIRING_TOKEN` checker finds cases where JWTs are created without an expiration time, making them tokens that are valid forever.

SAT-28149; SAT-28139

`CONFIG.ANDROID_BACKUPS_ALLOWED`, a new Android CodeXM checker, reports a defect in an `AndroidManifest.xml` file when an application is configured to allow its data to be backed up. Backup files can leak sensitive information or can be tampered with and then restored to the same or to a different device, potentially evading security controls and assumptions.

SAT-28150

The `CONFIG.REQUEST_STRICT_SSL_DISABLED` checker finds cases where the request module makes calls over an SSL channel and disables the verification of the SSL certificate.

SAT-28160; SAT-28162

`CONFIG.ANDROID_OUTDATED_TARGETSDKVERSION`, a new Android CodeXM checker, reports a defect in an `AndroidManifest.xml` file when an application is configured to target a version of the Android operating system that is not the latest available. By targeting older OS versions, the application cannot take advantage of security enhancements added in later OS versions.

SAT-28161; SAT-28159

`CONFIG.ANDROID_UNSAFE_MINSDKVERSION`, a new Android CodeXM checker, reports a defect in an `AndroidManifest.xml` file when an application is configured to run on a legacy Android operating system version that no longer receives security updates and that contains high-risk, publicly known vulnerabilities. Allowing an application to execute on such Android versions is unsafe, as malicious applications might exploit operating system weaknesses to perform a variety of attacks.

SAT-28225

Further increased Coverity's coverage of the AUTOSAR C++14 standard, 18-10 edition.

SAT-28262

Trust options have been added for the following checkers: `INTEGER_OVERFLOW`, `TAINTED_STRING`, `TAINTED_SCALAR`, `OS_CMD_INJECTION`, `PATH_MANIPULATION`, `XPATH_INJECTION`, and `SQLI`.

SAT-28466

Updated the version of SpotBugs used internally. The following checkers are no longer available due to removal from newer versions of SpotBugs:

- `VA_FORMAT_STRING_BAD_CONVERSION`
- `VA_FORMAT_STRING_BAD_CONVERSION_TO_BOOLEAN`
- `VA_FORMAT_STRING_BAD_CONVERSION_FROM_ARRAY`
- `VA_FORMAT_STRING_NO_PREVIOUS_ARGUMENT`
- `VA_FORMAT_STRING_ARG_MISMATCH`
- `VA_FORMAT_STRING_BAD_ARGUMENT`
- `VA_FORMAT_STRING_MISSING_ARGUMENT`
- `VA_FORMAT_STRING_ILLEGAL`
- `VA_FORMAT_STRING_EXTRA_ARGUMENTS_PASSED`

- `VA_FORMAT_STRING_EXPECTED_MESSAGE_FORMAT_SUPPLIED`

60.3.2.2. Known issues and solutions

SAT-7224, 43971: `XSS`

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

SAT-17490, 84256: `INTEGER_OVERFLOW` churn

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

60.3.3. Compiler configuration, Build capture, and Compiler Integration Toolkit (CIT) 2019.03

This section lists new features, bug fixes, and known issues related to Coverity-supported compilers (including configuration), and the Compiler Integration Toolkit (CIT).

60.3.3.1. Important Compiler Integration Toolkit (CIT) information

There were several deprecations and EOLs for Compiler Integration Toolkit (CIT) this release:

- Support for emitting Java on Solaris SPARC has been dropped. (CMPJ-1105)
- Support for ARM ADS 1.1 C/C++ compilers is now deprecated.
Support for ARM ADS 1.2 C/C++ compilers is now deprecated.
Support for ARM RVDS 2.0–4.1 C/C++ compilers is now deprecated. (CMPG-2928)
- Support for the STMicroelectronics GNU C/C++ 4.1.1 compiler is now deprecated.
Support for the STMicroelectronics ST Micro C/C++ 2.3.1 compiler is now deprecated. (CMPG-2929)
- Support for the IAR Embedded Workbench C/C++ 7.30B–8.10 compiler for the 8051 processor is now deprecated. (CMPG-2930)
- Support for the Microsoft Visual C++ 6 and 2003 compilers is now deprecated. (CMPG-2931)
- Support for Solaris 10 is now deprecated. (CMPG-2932)
- Support for GNU GCC and G++ 2.7.2–8.1.0 compilers on Mac OS is now deprecated. (CMPG-2933)
- Support for the HP-UX platform has been removed. (CMPG-2945)
- Support for the Android Jack compiler has been dropped. (CMPG-2947)
- Support for the Xbox 360 compiler has been dropped. (CMPG-2948)
- Support for NetBSD 6.1 and earlier has been dropped. (CMPG-2951)
- Support for the Swift compiler running in Xcode 9.0-9.4 is now deprecated. (CMPSWIFT-241)

60.3.3.2. New and changed features

- Added Java 11 support for `cov-emit-java`. (CMPJ-1087)
- Added support for wildcards in the `cov-emit-java --module-source-path` option. (CMPJ-1121)
- Added support for the Green Hills Optimizing C and C++/EC++ ARM 2018.1.4 compiler. (CMPCPP-8322)

60.3.3.3. Bug fixes

The following bugs are fixed for compilers and the Compiler Integration Toolkit (CIT) for Coverity Analysis analysis in 2019.03:

CAP-1412

Fixed an issue with `cov-build` on Unix-like platforms that would cause the build to fail with "capture-unix.c: assertion failed: s_new".

CMPCSH-987

Addressed a Coverity Analysis false positive involving the `CallerMemberName`, `CallerLineNumber`, and `CallerFilePath` attributes in C# and Visual Basic analysis.

CMPCPP-4267, 64284

`cov-emit` will now ignore the following set of pragma directives: `diag_suppress`, `diag_remark`, `diag_warning`, `diag_error`, `diag_once`, and `diag_default`. Previously, use of these directives intended for the native compiler could inadvertently interfere with diagnostics produced by `cov-emit`.

CMPCPP-5622

Non-source files (such as `.doj` and `.idf`) that are input to a Blackfin compiler invocation are no longer incorrectly treated as source files.

CMPCPP-6055

Improved the performance of `cov-emit` when emitting classes with a large number of base classes and virtual functions.

CMPCPP-6495, 104543

`cov-emit` now accepts C++17 nested namespaces in pre-C++17 modes when emulating recent versions of GCC.

CMPCPP-7277, 117408

Fixed various parse errors, which were caused by friend template functions.

CMPCPP-7498

`cxxintppc.exe`, the Green Hills Integrity OS compiler, no longer causes errors when being used in preprocessor code that detects whether exceptions are enabled.

CMPCPP-7716, 117190

Fixed a spurious error in `cov-emit` where a template parameter default argument included a cast operation applied to a nontrivial expression with a constant value.

CMPCPP-7810

Fixed a spurious error that occurred in `cov-emit` when a type qualifier was applied to a function type in an alias template.

CMPCPP-8277

Fixed an issue involving MetaWare `ccac` switches, where some switches were insufficiently translated.

CMPCPP-8372

Fixed an internal error in `cov-emit` that could occur when certain complex macro expansions were performed.

CMPCPP-8382

Corrected the Clang compiler configuration to forward the `-flto` and `-fvisibility` options to compiler probes in order to satisfy the requirements when using the `-fsanitize=cfi` option.

CMPCPP-8409

Fixed a case of memory corruption in `cov-emit` when a call was made to an extern inline method, and the target of the call was wrapped in parentheses. For example: `(f)()`.

CMPCSH-987

Addressed a Coverity Analysis false positive involving the `CallerMemberName`, `CallerLineNumber`, and `CallerFilePath` attributes in C# and Visual Basic analysis.

CMPCSH-996

Fixed an issue in `cov-emit-cs` where properties assignments in a deconstruction expression could cause the following assertion failure in analysis: "Function object should have function type".

CMPJ-1065

Fixed a Java compilation bug involving type annotations on method parameters. If the type was defined in an implicitly resolved source file, it could cause a crash.

CMPJ-1068

Fixed an issue in `cov-emit-java` where the frontend could generate incorrect information for generic classes, depending upon the order in which classes were encountered.

CMPJ-1118

Fixed an issue where `cov-emit-java` was rejecting command lines with multiple instances of the `--add-reads`, `--add-exports`, and `--add-modules` options.

CMPJS-399

Absolute file paths in HTML files are now resolved correctly against the web root path. Users can pass the web root path with the `--fs-library-path` option.

CMPJS-454

Fixed the issue where `for...of` loops with destructuring iterators caused false positive `NO_EFFECT` defects.

CMPJS-673

Improved the performance of emitting JavaScript (`.js`) source files that use JSX and TypeScript features.

CMPJS-711

Improved the extraction of AngularJS code from HTML attributes to avoid a spurious syntax error.

CMPJS-723

Fixed an issue in which source map files captured along with JavaScript source files could, in rare circumstances, result in a hang or crash while processing the JavaScript source.

CMPJS-735

Fixed an issue that avoids database corruption. The issue would occur when deleting JavaScript translation units that were associated with module link records.

CMPPY-228

Removed the extraneous error messages that were produced (by the Python 2 compiler) when the Python compiler version was unspecified and a later version of Python successfully parsed the source code.

CMPOCCPP-176

Clang compilers that are invoked with the `-fsyntax-only` option will cause any translation units in that invocation to no longer be emitted.

CMPOCCPP-206; CMPOCCPP-200; CMPOCCPP-198; CMPCPP-8242; CMPCPP-7534

Corrected an issue that caused `cov-internal-emit-clang` to fail with a "Key not found" assertion failure. This issue occurred when capturing Clang compiler invocations where Clang module support was enabled.

CMPCPP-6674, 107688

The `-lang` switch for the Renesas compilers now sets the source language for all source files on the command line, including source files that appear before the switch.

CMPCPP-7716

Fixed a spurious error in `cov-emit` where a template parameter default argument included a cast operation applied to a nontrivial expression with a constant value.

60.3.3.4. Known issues and solutions

CAP-1176, 97630

`cov-build --instrument` has a known issue when running the `xdcmake.exe` tool of Visual Studio 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 either:

- Modify the build such that `xdcmake.exe` is run from a 64-bit process.
- Ignore the `xdcmake.exe` process by adding `--capture-ignore xdcmake.exe` to your `cov-build` invocation.

CMPJS-286, 95651

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 used with some frameworks.

CMPJ-368, 65669, 65721

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 `LANG` or `LC_CTYPE` environment variables.

60.3.4. Coverity Analysis 2019.03 Commands

60.3.4.1. Important information about commands related to the build and capture process

This section provides updates about `cov-build` and related commands, including capture, emit, and translate commands.

- Buildless capture is no longer supported on Solaris SPARC. (BLC-457)

60.3.4.1.1. New and changed features

The following new and changed features were added for commands related to the build and capture process (including emit and translate commands) in 2019.03:

- C# buildless capture now supports projects targeting any target framework as long as the relevant SDK is present on the system. (BLC-395)
- For C# buildless capture: As long as the user has the required SDK installed on their system, buildless capture will now be able to capture projects targeting any target framework. (BLC-433)

60.3.4.1.2. Bug fixes

There were several bugs fixed for build and capture-related commands (including emit and translate commands) in 2019.03:

BLC-38

Customers no longer need to install Bower for use with buildless capture.

BLC-337

Capturing JavaScript projects with `cov-capture` on Windows should no longer leave occasional `node.exe` processes running after completion.

BLC-368

Capturing JavaScript projects with `cov-capture` on Windows should no longer encounter occasional hangs in `npm` during project inspection.

60.3.4.1.3. Known issues and solutions

Build-related commands have the following known issues and solutions:

CAP-812, 64428

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 hang occurs, 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 either:

- Uninstall KB2919355

OR

- Add the `--instrument` flag to your `cov-build` invocation:

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

CMPCPP-4764, 72964

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.

SAT-12174, 62745

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.

CAP-332, 26881, 38175

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

Error message:

```
[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
```

60.3.4.2. Commands related to analysis

This section lists new features, bug fixes, and known issues for `cov-analyze` and related commands.

60.3.4.2.1. New and changed features

There were no new features added or changed for commands related to the analysis process in 2019.03.

60.3.4.2.2. Bug Fixes

There were no bug fixes for analysis-related commands in 2019.03.

60.3.4.2.3. Known issues and solutions in 2019.03

Analysis-related commands have the following known issues and solutions:

CMPG-1741, 70845, 71216

The `cov-run-desktop` command sometimes fails on large Java compilations, potentially causing emit database corruption on Windows platforms. This can manifest as a `cov-analyze` crash. More commonly, `cov-emit-java` itself will fail with access violation crashes or errors concerning a failure to acquire a lock. These will appear in `cov-run-desktop-log.txt`. If this issue occurs, you can work around it by specifying `-j 1` with `cov-run-desktop`.

60.3.4.3. Commands related to Test Advisor

This section lists new features for Test Advisor.

60.3.4.3.1. Bug fixes in 2019.03

There were no bugs fixed for Test Advisor-related commands in this release.

60.3.5. Coverity Wizard 2019.03

This section lists new features, bug fixes, and known issues related to Coverity Wizard.

60.3.5.1. Important Coverity Wizard Information

There were no deprecations or EOLs this release.

60.3.5.2. New and changed features

Coverity Wizard has the following new and changed feature in 2019.03:

- `cov-wizard` now allows users to set the context path to connect to the Coverity Connect server. The host, port, and SSL fields in the *Commit Defect* configuration page have been merged into a single URL field. (PRD-10597)

60.3.5.3. Bug fixes

No bugs were fixed for Coverity Wizard in 2019.03.

60.3.5.4. Known issues and solutions

Coverity Wizard has the following known issues in version 2019.03:

PRD-9245, 90621

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

PRD-9208, 90489

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-8453, 83450

When using a self-signed certificate, if the user chooses not to trust a certificate, they might be prompted multiple times in a row (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-8227, 82196

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-7595, 77742

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.

PRD-6832, 70361

The guided policy creation wizard "Documentation" link fails to open properly on Linux. Open the *Coverity Wizard 2020.12 User Guide* separately to view this documentation.

PRD-6760, 69815

The *Guided Test Advisor Policy Creation Wizard* uses Java regex validation instead of the Perl regex validation that Coverity Analysis Test Advisor users. 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-5770, 59676

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-5387, 54143

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

PRD-5290, 53367

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.

60.3.6. Test Advisor 2019.03

Coverity Test Advisor is a component of the Coverity Analysis installation package.

60.3.6.1. Important Test Advisor information

There has been several deprecations and EOLs this release:

- The `export-ta-gae-data` subcommand of the `cov-manage-emit` command is no longer supported. (TADE-1981)
- Coverity Test Advisor support for Git 1.4–1.7 is deprecated. (TADE-1982)
- Coverity Test Advisor support for Mercurial 1.0–3.0 is deprecated. (TADE-1983)

60.3.6.2. New and changed features

Test Advisor has no new or changed features in 2019.03.

60.3.6.3. Bug fixes

No bugs were fixed for Test Advisor in 2019.03.

60.3.6.4. Known issues and solutions

Test Advisor has the following known issues and solutions in 2019.03:

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.

60.3.7. Dynamic Analysis 2019.03

Dynamic Analysis is a component of the Coverity Analysis installation package.

60.3.7.1. Known issues and solutions

Dynamic Analysis has the following known issues and solutions in 2019.03:

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.

JDA-681, 20788

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

JDA-694, 21417

Specifying certain combinations of the `instrument-arrays`, `instrument-collections`, `detect-races`, and `detect-deadlocks` options to the Dynamic Analysis agent have 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, then Dynamic Analysis reports races on neither collections nor arrays.

JDA-720, 22148

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

60.3.8. Architecture Analysis 2019.03

Coverity Architecture Analysis is a component of the Coverity Analysis installation package.

60.3.8.1. Important Architecture Analysis information

Architecture Analysis has one changed feature in 2019.03:

- Support for Architecture Analysis on 32-bit Windows operating systems is now deprecated and will be removed in a future release. (COVP-2098)

60.3.9. Extend SDK 2019.03

Coverity Extend Software Development Kit is a component of the Coverity Analysis installation package.

60.3.9.1. Known issues and solutions

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.

60.4. Coverity Desktop 2019.03

The Coverity Desktop plug-in is available for various platforms from the Coverity Connect *Downloads* menu.

60.4.1. Important information

There has been one EOL this release:

- Dropped Coverity Desktop support for QNX Momentics 5.0. (PRD-11694)

60.4.2. Coverity Desktop for Android Studio in 2019.03

60.4.2.1. New and changed features

Coverity Desktop for Android Studio has the following new and changed features in 2019.03:

- Added support for Android Studio v2018.3. (PRD-10683; PRD-11683)

60.4.2.2. Bug fixes

The following bug was fixed for Coverity Desktop for Android Studio in 2019.03:

PRD-11693

In Android Studio 3.0, some dialogs in **Analysis Configuration** have an extra help button with no help message available. It has been fixed in the latest Android Studio version.

60.4.2.3. Known issues and solutions

Coverity Desktop for Android Studio has no known issues in version 2019.03.

60.4.3. Coverity Desktop for Eclipse in 2019.03

60.4.3.1. New and changed features

Coverity Desktop for Eclipse has the following new and changed features in 2019.03:

- The Eclipse extension now allows users to set the context path to connect to the Coverity Connect server. The host, port, and ssl fields in the Coverity Connect configuration page have been merged into a single URL field. (PRD-10596)
- Added support for Java 11. (PRD-10641)
- Added support for Eclipse 4.10 (SimRel 2018-12) (PRD-11661)
- Added support for Clion v2018.3. (PRD-11684)
- Added support for Rubymine, PHPStorm, Webstrom, and Pycharm v2018.3. (PRD-11685)
- Added support for QNX Momentics 7.0. (PRD-11694)
- The Coverity Desktop plugin for Eclipse now supports Eclipse 4.10 (SimRel 2018-12)

60.4.3.2. Bug fixes

There following bug was fixed for Coverity Desktop for Eclipse in 2019.03:

PRD-10670

The `--enable-java-annotation-framework-support` option has been enabled by default, so plugin users no longer need to specify it in their `coverity.conf` file as an extra `cov-build` option.

60.4.3.3. Known issues and solutions

Coverity Desktop for Eclipse has the following known issues in version 2019.03:

PRD-10694

For OXS 10.14 users with JDK-8136913 installed, using the `hostname_regex` in the `coverity.conf` file causes 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 may 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/`.

60.4.4. Coverity Desktop for Microsoft Visual Studio in 2019.03

60.4.4.1. New and changed features

Coverity Desktop for Microsoft Visual Studio has the following new and changed features in 2019.03:

- The Visual Studio extension now allows users to set the context path to connect to the Coverity Connect server. (PRD-10693)

60.4.4.2. Bug fixes

Coverity Desktop for Microsoft Visual Studio has no bug fixes in 2019.03.

60.4.4.3. Known issues and solutions

Coverity Desktop for Visual Studio has no known issues in version 2019.03.

60.4.5. Coverity Desktop for IntelliJ IDEA in 2019.03

60.4.5.1. New and changed features

Coverity Desktop for IntelliJ IDEA has the following new and changed feature(s) in 2019.03:

- The IntelliJ extension now allows users to set the context path to connect to the Coverity Connect server. The host, port, and ssl fields in the Coverity Connect configuration page have been merged into a single URL field. (PRD-10596)
- Added support for IntelliJ v2018.3. (PRD-11656)

60.4.5.2. Bug fixes

Coverity Desktop for IntelliJ has no bug fixes in 2019.03.

60.4.5.3. Known issues and solutions

Coverity Desktop for IntelliJ IDEA has the following known issues in version 2019.03:

PRD-7453, 76907

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, 80573

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

PRD-7991, 80599

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, 80693

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

PRD-8397, 83106

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

PRD-8042, 80698

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.

PRD-10076, 105052

When using whole program checkers in IntelliJ, a warning about missing class files might be seen 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, 119444

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.

60.5. Coverity Report Generators 2019.03

The Coverity Report Generators' installer can be downloaded from the downloads page in Coverity Connect.

60.5.1. Important Coverity Report Generators information

Coverity Report Generators have no deprecations or EOLs in 2019.03.

60.5.1.1. New and changed features

The following new and changed features were added for the Coverity Report Generators in 2019.03:

- The Coverity Report Generators in this installation use a new and unified configuration file mechanism. All report generators are now configured using the same configuration file format and schema. The new format is human-readable and YAML-based.

Please see the `config/config.yaml` configuration file for full documentation and an example of how to configure the report generators.

The old configuration formats are no longer supported. (RG-1086)

60.5.1.2. Bug fixes

There was one bug fixed for the Coverity Report Generators in 2019.03:

RG-902

Updated the Report Generators so that they refer to the 2017 version of the OWASP Top 10 Application Security Risks.

60.5.1.3. Known issues and solutions

Coverity Report Generators have the following known issues and solutions:

RG-1142

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

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

60.6. Coverity Documentation 2019.03

The following new documents and changes were made in 2019.03:

CMPG-2882

The `extract-files` sub-command was added to the `cov-manage-emit` command in release 2018.09, but the `<<filename>>` argument was accidentally omitted from the Command Reference. This argument has now been added to the Command Reference.

COVP-2101

Coverity Analysis, Coverity Wizard, and Coverity Desktop support for macOS 10.12 has been deprecated as of 2019.03.

IM-23447

Added `PRINTF_ARGS` checker to the Checker History Appendix in the Coverity Checker Reference.

PRD-11703

Coverity Desktop support for Visual Studio 2013 has been deprecated as of 2019.03.

PRD-11707

Coverity Desktop support for QNX Momentics 5.0 is end-of-life.

Coverity Desktop support for QNX Momentics 7.0 is added.

SAT-27607

The following commands and options have already been released, but were not documented. Documentation has been added for the following:

- The `cov-emit-vb` command
- These options to the `cov-emit-cs` command:
 - `--langversion`
 - `--link`

- `--target`

SAT-28378

`__coverity_tainted_data_argument__`, `__coverity_tainted_data_return__`,
`__coverity_tainted_string_argument__`, and
`__coverity_tainted_string_return_content__` primitives have been deprecated.
These primitives are replaced by `__coverity_mark_pointee_as_tainted__`, which allows
specification of `taintType`.

SAT-28597

Coverity Analysis now supports the following frameworks: ReactiveX (RxJava, Reactor), Ember,
Fastify, Restify, and Google Cloud APIs (Storage).

Chapter 61. Coverity 2018.12-12 Release Notes

Table of Contents

61.1. Coverity Analysis bug fixes	227
---	-----

61.1. Coverity Analysis bug fixes

SATW-3067

Fixed a false positive in CERT EXP62-CPP that occurred when using the `memset` function on an array of pointers to class objects.

SATW-3069

Fixed a false positive in CERT FLP32-C where user-defined functions were mistaken for standard math functions.

SATW-3071

Fixed a false positive in CERT OOP51-CPP where an expected object slicing did not occur.

SATW-3073

Fixed a false positive in CERT OOP57-CPP that occurred when using the `memset` function on an array of pointers to class objects.

Chapter 62. Coverity 2018.12-11 Release Notes

Table of Contents

62.1. Coverity Platform bug fixes 228

62.1. Coverity Platform bug fixes

IM-23799

Fixed a problem that was preventing the generation of Policy Manager chart images in email notifications.

Chapter 63. Coverity 2018.12-10 Release Notes

Table of Contents

63.1. Coverity Analysis bug fixes	229
---	-----

63.1. Coverity Analysis bug fixes

CMPCPP-8802; SAT-29998

Improved Coverity Analysis with Clang compilers so that false positives are avoided when an object is assigned a temporary value and that object also contains a field that affects a condition.

Chapter 64. Coverity 2018.12-9 Release Notes

Table of Contents

64.1. Coverity Analysis bug fixes	230
---	-----

64.1. Coverity Analysis bug fixes

SAT-29762

Fixed an error where invoking the version of `swprintf` that does not take a size argument would result in `OVERRUN` false positives.

SAT-29761

Updated `PRINTF_ARGS` to understand the Microsoft-specific `I32`, `I64`, and `I` size attributes.

SAT-28027

Fixed a class of `OVERRUN` false positives that occurred when accessing a buffer within a decreasing loop in a callee.

SAT-26361

Fixed a class of `NO_EFFECT` false positives, which incorrectly reported on a misused comma operator when `va_start` was invoked in Visual Studio 2017.

Chapter 65. Coverity 2018.12-8 Release Notes

Table of Contents

65.1. Coverity Analysis bug fixes	231
---	-----

65.1. Coverity Analysis bug fixes

CMPCPP-8672

A number of performance improvements have been made and a heuristic has been added to try to work around the performance issues.

In addition to a number of performance improvements, we've implemented a mitigation option that acts as a temporary workaround. It is like `--no_emit_referenced_types`, but rather than being absolute, it will be a dial. The dial is zero to infinity and it controls a heuristic, which only means that lower settings will be faster. This dial will impact fidelity by eliminating some referenced types, however, the impact is expected to be minimal. The ideal setting will be code base-specific and will have to be determined experimentally, although just using 0 for all code bases might be acceptable.

For example:

```
$ COVERITY_REFERENCED_TYPES_THRESHOLD=80 cov-build -dir emit make
```

Chapter 66. Coverity 2018.12-7 Release Notes

Table of Contents

66.1. Coverity Analysis bug fixes	232
---	-----

66.1. Coverity Analysis bug fixes

SAT-29425

Fixed an error that caused SSL verification failures, resulting in the following error message:
"Server's SSL certificate is not trusted. Its CA certificate was found but a chain of trust could not be constructed." This error occurred when multiple CA certificates with the same name were installed.

Chapter 67. Coverity 2018.12-6 Release Notes

Table of Contents

67.1. Coverity Compiler Integration Toolkit (CIT) bug fixes	233
---	-----

67.1. Coverity Compiler Integration Toolkit (CIT) bug fixes

CMPJS-729

Defects were reported at line 1 of a generated JavaScript file, where the source file (indicated by a source map) was not found on the file system. Now, the defects are reported at the relevant location in the generated file.

CMPJS-731

Provided a workaround that avoids an assertion violation during analysis. The crash occurs when there is an inconsistency in the emit of TypeScript syntax: `export = class { ... }.`

CMPCPP-8547

Corrected an issue with Clang compilers that resulted in a "decl is part of a template" assertion failure error message and a TU loss. This occurred when capturing a build where the support for compliance checkers was enabled.

Chapter 68. Coverity 2018.12-5 Release Notes

Table of Contents

68.1. Coverity Connect bug fixes	234
--	-----

68.1. Coverity Connect bug fixes

IM-22785

Fixed a `cov-admin-db` upgrade that would fail if the current user was not a superuser on an external database.

IM-23281

Fixed the `totalNumberOfRecords` count in `getUsers` WSAPI. Prior to the fix, the `totalNumberOfRecords` count was being incorrectly calculated.

Chapter 69. Coverity 2018.12-4 Release Notes

Table of Contents

69.1. Coverity Compiler Integration Toolkit (CIT) bug fixes	235
---	-----

69.1. Coverity Compiler Integration Toolkit (CIT) bug fixes

CMPJ-1112

Fixed an issue in `cov-emit-java` where the front-end could generate incorrect information for generic classes depending upon the order in which classes were encountered.

CMPJ-1127

Fixed an issue where `cov-emit-java` was rejecting command lines with multiple instances of the `--add-reads`, `--add-exports`, and `--add-modules` options.

Chapter 70. Coverity 2018.12-3 Release Notes

Table of Contents

70.1. Coverity Connect bug fixes	236
--	-----

70.1. Coverity Connect bug fixes

IM-23483

Improved `cov-commit-defects` to enable and quicken the processing of defect instances through concurrency.

IM-23495

You can now specify email addresses for custom top-level-domains (TLDs) during an LDAP user import.

Chapter 71. Coverity 2018.12-2 Release Notes

Table of Contents

71.1. Coverity Analysis bug fixes	237
---	-----

71.1. Coverity Analysis bug fixes

CMPOCCPP-210; CMPOCCPP-206; CMPOCCPP-200; CMPOCCPP-198; CMPOCCPP-7534;
CMPOCCPP-8242

Corrected an issue that caused `cov-internal-emit-clang` to fail with a "Key not found" assertion failure diagnostic when capturing Clang compiler invocations where Clang module support enabled.

CMPCPP-8396, 116638; CMPCPP-7236, 116638

Fixed a Microsoft compatibility issue in `cov-emit` where a parse error was emitted on a specialization of a template function with different exception specifiers.

Chapter 72. Coverity 2018.12-1 Release Notes

Table of Contents

72.1. Coverity Analysis bug fixes	238
---	-----

72.1. Coverity Analysis bug fixes

BLC-390

Previously, when `cov-capture` was used on Windows to capture a JavaScript project that uses the Bower project manager, the Bower utility failed to run. As a result, dependencies were not downloaded correctly. This has been fixed.

SAT-28127

Fixed a bug where the `enabled-checkers` setting in the `ANALYSIS.metrics.xml` file did not include regular quality or security checkers, even though they were enabled.

Chapter 73. Coverity 2018.12 Release Notes

Table of Contents

73.1. Important information for 2018.12	239
73.2. Coverity Platform 2018.12	240
73.3. Coverity Analysis 2018.12	244
73.4. Coverity Desktop 2018.12	269
73.5. Coverity Documentation 2018.12	273

73.1. Important information for 2018.12

Due to a change in our bug tracking system, items are now identified by two bug numbers:

- One reflecting the identity of the bug in our **old** bug tracking system, formatted like this: XXXXXX. (For example, 374568.)
- One reflecting the identity of the bug in our **new** bug tracking system, formatted like this: CODE-XXXXX. (For example, IM-22788.)

 **Note**

Bugs with only a CODE-XXXXX number do not have an old number.

73.1.1. Deprecated and End-of-Life (EOL) Products in Coverity 2018.12

Support for the following products, features, platforms, and third-party tools is classified as deprecated or end-of-life as of the Coverity 2018.12 release.

73.1.1.1. Deprecated Products

Support for the following products and features is deprecated as of the Coverity 2018.12 release.

Table 73.1. Deprecated products

Product	Comments
Accurev 6.2 support	Coverity Test Advisor Support SCM Systems
AIX 6.1 support	Supported Platforms for Coverity Analysis
Android Studio 2.2 support	Supported Platforms for Coverity Analysis
Desktop plugin support for Eclipse v4.5	Coverity Desktop Eclipse platform support
GNU-GCC version 2.x compiler support	Supported Compilers: Coverity Analysis for C/C++
HI-TECH PICC compiler support	Supported Compilers: Coverity Analysis for C/C++
HP-UX platform support	Supported Platforms for Coverity Analysis
IntelliJ IDEA 2016.1 and 2016.3 support	Supported Platforms for Coverity Analysis

Product	Comments
Linux Kernel 2.6.31 and older deprecated for Coverity Connect	Supported Platforms for Coverity Analysis
misra_config settings have been deprecated	Users can use the <code>coding_standard_config</code> option instead.
NetBSD 6.1 and earlier deprecated for Coverity	Supported Platforms for Coverity Analysis
Perforce 2014.2-2015.1 support	Coverity Test Advisor Support SCM Systems
2016.3 version support for RubyMine, WebStorm, PyCharm, and PhpStorm	Supported Platforms for Coverity Analysis
SNC C/C++ and SNC GNU C/C++ compilers support	Supported Compilers: Coverity Analysis for C/C++
TFS 2010 support	Coverity Test Advisor Support SCM Systems
TriMedia compiler support	Supported Compilers: Coverity Analysis for C/C++
Visual Studio 2008 support	Supported Platforms for Coverity Analysis
Windows Server 2008 SP2 support deprecated for Coverity Analysis	Coverity Desktop Supported platforms
Xbox 360 compiler	Supported Compilers: Coverity Analysis for C/C++
Xcode gcc 4.2 and llvm-gcc 4.2 support	Supported Compilers: Coverity Analysis for C/C++

73.1.1.2. End-of-Life Products

Support for the following products and features is dropped in the Coverity 2018.12 release.

Table 73.2. End-of-Life Products

Product	Comments
Accurev 6.0 and 6.1 support	Coverity Test Advisor Support SCM Systems
Apple Clang 2.1-5.1 support	Supported Compilers: Coverity Analysis for C/C++
Clearcase 7.0.x, 7.1.x and 8.0.x support	Coverity Test Advisor Support SCM Systems
Eclipse 4.5 support	Supported Compilers: Coverity Analysis for C/C++
Perforce 2007.2 – 2014.1 support	Coverity Test Advisor Support SCM Systems
Linux Kernel 2.6.31 and older support dropped for Coverity Analysis	Supported Platforms for Coverity Analysis
Scratchbox support dropped for Coverity Analysis	Supported Compilers: Coverity Analysis for C/C++
TFS 2008 support	Coverity Test Advisor Support SCM Systems

73.2. Coverity Platform 2018.12

This section provides release notes for Coverity Platform components.

73.2.1. Coverity Connect 2018.12

Coverity Connect is a component of the Coverity Platform installation package.

73.2.1.1. Important Coverity Connect Information

There has been one deprecation for Coverity Connect this release:

- Coverity Connect support is deprecated in the Pacific release for Linux Kernel 2.6.31 and earlier. (SAT-27278)

73.2.1.1.1. New and changed features

- The `WRITE_DEFECTS` variable has been renamed `WRITE_ISSUES_JSON`. (RG-1049)
- Added support for Tomcat 8.0.53. (IM-22827)

73.2.1.1.2. Bug fixes

The following bugs are fixed for Coverity Connect:

IM-23089

Fixed an RBAC issue which caused an infinite refresh loop on the *Visitor* role when viewing a project.

IM-22973

For concurrent commits with a `commitPoolThreads` value of less than five, the value will then be changed to 5, which is the default commit pool size.

IM-22919

Improved the Coverity Connect instance startup and availability.

IM-22820

Fixed a bug in the code that sent email notifications about commit action completion. This was causing commit actions to hang in cases where **Email Configuration** had incorrect authentication data.

IM-22790

Fixed an issue that was causing excessive memory usage when browsing functions in *High CCM (>15)* view. We also improved the performance for browsing functions in this view.

IM-22751

When restoring large backup dump files, `pg_restore` might throw an error such as "pg_restore: [parallel archiver] could not create worker process: Cannot allocate memory". We've fixed this by adding more memory.

IM-21330, 107884

Fixed an issue where the *Configuration - Users & Groups* dialog would close immediately after being opened by an *Administrators* group member.

IM-19218, 85655

It is now possible to create two LDAP configurations that have the same server and port.

INS-2428

Fixed an issue where the `cov-platform` installer would not properly create a `keystore` in new Coverity instances if the hostname had been changed during **Backup and Restore**.

INS-2412

Resolved an issue where installing `cov-analysis` into *Program Files* on Windows without the Extend SDK would cause *Incremental Updates* to fail.

INS-2409

Fixed an issue where a `stacktrace` would be printed to the console when upgrading a Coverity instance with a disabled HTTP proxy server.

INS-2346

The Linux Coverity Connect installer will now fail if it is run with root permissions. It will also provide the user with an explanation for the failure.

INS-2399

Improved the error handling and messaging for argument parsing of unattended installers.

INS-1910

Resolved an issue where a *Backup and Restore* of Coverity on Windows could not be installed into a non-existent directory.

INS-1689, 83924

Fixed an error that occurred for some Windows users where the Coverity updater failed to work properly. Prior to the fix, the updater failed to remove older files and would not properly create any new files.

INS-1523

When performing a Backup and Restore, the trust store from the old Coverity Connect is now migrated to the new instance.

INS-1395

The Coverity Connect installer has been updated so that, when performing a *Backup and Restore*, the user is prompted for the previous installation directory before the new installation directory. The installer will attempt to determine the previous installation directory, and prefills the selection, where possible.

RG-999

You can now use a `.JSON` file to configure security reports.

73.2.1.1.3. Known issues and solutions

Coverity Connect has the following known issues:

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.

INS-2307, 118662

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-2133, 112939

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.

IM-19690, 89946

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-19685, 89897

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

IM-19048, 84453

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

IM-18710, 82648

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-18707, 82643

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.

CPU-38, 82579

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.

CPU-17, 80045

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.

IM-17701, 75559

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

IM-17660, 75263

An error occurs when a custom role is created using a multi-word rolename that is the same as a built-in rolename, even if there are case differences between the two rolenames.

INS-1477, 73401

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 (1.8), or by specifying a max heap setting for `cov-im-daemon`.

IM-16076, 67748

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.

INS-1274, 63454

Although the upgrade doc 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, fall back to backup-and-restore upgrade.

73.2.2. Coverity Policy Manager 2018.12

Coverity Policy Manager is a component of the Coverity Platform installation package.

There are no new or changed features, and no bug fixes or known issues for this release.

73.3. Coverity Analysis 2018.12

This section provides updates about Coverity Analysis components.

73.3.1. Important Coverity Analysis information

There have been several deprecations and EOLs this release:

- Support for the Scratchbox compiler has been dropped. This EOL includes removal of the `cov-build-sbox` and `cov-configure-sbox` commands. (CAP-1325, 121094)
- Support for Windows Server 2008 SP2 has been deprecated as of this release. (CMPG-2865)
- Support for the HP-UX platform has been deprecated as of this release. (CMPG-2857)
- The following SCMs are no longer supported as of this release:
 - Accurev 6.0 and 6.1
 - Clearcase 7.0.x, 7.1.x and 8.0.x
 - Perforce 2007.2 – 2014.1
 - TFS 2008 (COVP-2062)
- Support for NetBSD 6.0 and 6.1 has been deprecated as of this release. (COVP-2075)

- Support for AIX 6.1 has been deprecated as of this release. (COVP-2071)
- Support for the following source control management systems (SCMs) is deprecated as of this release:
 - Accurev 6.2
 - Perforce 2014.2 and 2015.1
 - TFS 2010 (COVP-2062)
- Coverity Analysis support for Linux Kernel 2.6.31 and earlier has been dropped. (SAT-27278)

73.3.1.1. New and changed features

Coverity Analysis has the following new and changed features:

- Added support for .NET Core 2.0-2.1 applications on Windows. (SAT-27606; SAT-20558, 96584; CMPG-2864)
- Added support for TypeScript. (SAT-16388, 79255)
- Added support for C# 7.1–7.3. (CMPG-2863)
- Added support for Swift 3.3 and 4.1.x. (CMPSWIFT-187)
- The new `cov-emit --enable_user_sections` option enables the user sections compiler extension, allowing variable placement at specific addresses in memory. Compilers that support this extension include the IAR ARM compiler which uses the `@` operator for this purpose, and the CodeWarrior compiler which uses `:`. Please consult your compiler manual for more information.

This option supersedes `cov-emit`'s deprecated `--allow_declare_at_address` option. (CMPCPP-6362, 102666)

- Added support for NetBSD 7.0. (COVP-2073)
- Added support for macOS 10.14. (COVP-2038)
- `cov-capture`: This new command captures source files for analysis from the file system or from an SCM repository, without a build. (BLC-183)

73.3.1.2. Bug fixes

SAT-28027; SAT-27921

Fixed a class of `OVERRUN` false positives that occurred when accessing a buffer within a decreasing loop in a callee.

SAT-27792

Improved the performance of the `PROPERTY_MIXUP` checker for auto-implemented properties.

SAT-27783

Fixed a performance issue with running MSIRA C++-2008 rules on some code base where the analysis hangs on some work units.

SAT-27555

Fixed a `cov-analyze` crash in Java webapp security checkers.

SAT-27482

Implemented new coding standard ISO TS17961.

SAT-27455

Improved the coverage of the AUTOSAR C++14 coding standard. Refer to the Checker Reference for more information about AUTOSAR C++ coverage.

SAT-27353

Fixed a `cov-analyze` recoverable error that occurred when parsing EL in JavaServer Pages (JSPs).

SAT-27292

Fixed a class of `OVERRUN` false positives where a buffer was accessed within a parameter-bounded loop in a callee and when the loop could exit early.

SAT-27213

Fixed an analysis crash in Coverity Desktop that produced the following error message: "Suppressible assertion failed at analysis/work-unit/work-unit.cpp: tu_index.is_present()".

SAT-27206

Added custom models for the `strcpy` and `strlcat` functions.

SAT-27061

Added support for reporting `OVERRUN` defects in loops where the buffer index is a field of a local variable.

SAT-27040; SAT-27005

Fixed a web application security analysis crash that occurred when tainting members of a class that were missing a definition.

SAT-27038

Fixed a `cov-analyze` crash in the `WRAPPER_ESCAPE` checker where the same variable was cast to multiple different wrapper types.

SAT-27010

Fixed an analysis crash caused by unexpected field types during a `deep write`.

SAT-26986

Fixed a `cov-analyze` crash that produced the following message when analyzing Visual Basic .NET: "Invalid relationship type: dyn==".

SAT-26952

Fixed a crash that occurred when decompiling some .NET assemblies.

SAT-26937

Fixed a crash that occurred in `cov-commit-defects`, where the use of the `--strip-path` option caused multiple files to have the same name. The crash would produce the following error message: "File symbols should already have been handled".

SAT-26894

Fixed a `cov-format-errors` crash involving the use of C++11 variadic templates.

SAT-26880

Fixed a class of `RESOURCE_LEAK` false negatives involving self-pointers (such as patterns like `obj->field = obj + offset`).

SAT-26882

Fixed an error with a function that caused an unconditional exit, where, if that function contained a loop, our analysis would incorrectly fail to notice the unconditional program exit.

SAT-26877

`new(nothrow)` correctly returns a null pointer, and therefore reports on more defects.

SAT-26836

Fixed the line count metric for function lines of cases where a function was called with a default argument value. In some cases, the location of the default argument's definition would incorrectly be included in the line span, yielding a very large line count.

SAT-26776

The documentation for `cov-install-updates` has been updated to clarify that the `check`, `list`, and `install` subcommands require connection options sufficient to access update information from a connected Coverity server.

SAT-26658

The C# `PATH_MANIPULATION` checker was enhanced to recognize that filtering paths using `GetInvalidPathChars()` is insufficient to sanitize against upwards directory traversal.

SAT-26593

Fixed a `cov-analyze` crash that produced the following message: "In `init_fn(PASS_BY_VALUE)`: `opt-uint.hpp:124`: assertion failed: Expected a value to be present for optional integer".

SAT-26456

The `XML_EXTERNAL_ENTITY` checker reports on any insecurely configured parser, regardless of whether its input is untrusted or not, when `cov-analyze` is run with `--enable-audit-mode`.

SAT-26437, 121689; SATW-2513; IM-19280, 86101

Fixed a crash in `cov-commit-defects` and `cov-format-errors` where the following error message was produced: "Cannot parse XML: Char 0xFFFF out of allowed range".

SAT-26413

Improved the precision of `UNINIT` for classes accessed through subclasses.

SAT-26404

Fixed a class of `RESOURCE_LEAK` false positives when an integer representing a resource (including a pointer cast to an integer, using `allow_cast_to_int`) is passed on to an unimplemented function.

SAT-26402, 121720

Fixed a crash on Windows that occurred when a defect file grew larger than 4GB.

SAT-26378

Fixed a class of `NULL_RETURNS` false positives that occurred when a function returned a C++ reference type.

SAT-26362

Fixed a `EXCEPTION_STACK_OVERFLOW` crash that sometimes occurred when MISRA C-2012 Rule 18.2 or CERT C ARR36-C checkers were enabled for `cov-analyze`.

SAT-26361, 121561

Fixed a class of `NO_EFFECT` false positives which incorrectly reports misused comma operators with Visual Studio 2017 `va_start`.

SAT-26299

Fixed a crash that occurred when parsing some invalid `coverity.conf` files with the `extend_checkers` configuration.

SAT-26289, 121177

Fixed a class of `OVERRUN` false positives where a buffer was accessed within a loop in a callee. That access was guarded by an extra condition which prevented the overrun.

SAT-26264, 121087

Improved the model for the `mbrlen` function and its related functions, so that it can report on UNINIT problems.

SAT-26253, 121068

Improved the `OVERRUN` checker to detect more cases where a buffer argument was accessed using a constant provided in a callee.

SAT-26227, 120968

Fixed a class of `STREAM_FORMAT_STATE` false positives with a function that uses a saver class, and calls `flags`, `setf`, or `setiosflags`.

SAT-26197; SAT-26006, 120030

Fixed an `OVERRUN` false negative so that the pointer is compared against `NULL` before the overrun occurs.

SAT-26179, 120813; SAT-24151, 113861

Added the `require_exact_zero` option (default true) to the `DIVIDE_BY_ZERO` checker. This option controls whether the checker should report a defect only when the denominator is known to be exactly zero on the path being analyzed.

SAT-26093, 120534

Fixed a class of `MISMATCHED_ITERATOR` false positives where `std::move` was applied to STL containers.

SAT-26089, 120529

Fixed a class of `ALLOC_FREE_MISMATCH` false negatives where the allocations were done conditionally.

SAT-26008, 120048

Improved the precision of the `OVERRUN` checker for cases where a callee checks bounds.

SAT-26007

Improved the `OVERRUN` checker so that it handles cases where an allocation uses the number of bytes that is a multiple of its argument.

SAT-26005, 119887

Improved the `OVERRUN` checker to report cases where the address of a buffer or buffer length is passed as an argument.

SAT-25974, 119940

Fixed a class of `DEADCODE` false positives with the `report_redundant_tests` option and try/catch blocks.

SAT-25954, 119887; SAT-22476, 107159

Fixed a class of `NULL_RETURNS` false positives that occurred where the `operator new` function returned a null pointer.

SAT-25952, 119879

Fixed a class of `CHECKED_RETURN` checker false positives caused by the `--aggressiveness-level high` setting overriding the effect of setting `CHECKED_RETURN:stat_threshold` directly.

SAT-25832, 119577; SAT-25793, 119491

Improved the `OVERRUN` checker so that it reports more cases where the buffer or its index was copied through local variables.

SAT-25733, 119341

Fixed `STRING_NULL` false negatives where `memcpy` produced non null-terminated strings.

SAT-25624, 118989

Added models for `zlib`, improving precision of the analysis for code that uses the `zlib` library.

SAT-25620

Fixed a class of `STRING_NULL` false positives where a non null-terminated string's address is printed using `%p`.

SAT-25999

`cov-run-desktop` no longer sets the `100-Continue` expectation, fixing compatibility issues with some proxies that do not handle this expectation correctly.

SAT-25996, 119998

Improved the `OVERRUN` checker to report more cases involving multiple levels of callees.

SAT-25270, 117911

Fixed a class of `ARRAY_VS_SINGLETON` false positives that occurred where the pointer to a class was incremented while the pointer was also being cast to a base class.

SAT-24885; SAT-24795; SAT-27379; SAT-13026

Fixed a class of `OVERRUN` false positives, where the index was an expression (such as an addition or multiplication), and that same expression was checked beforehand.

SAT-23518, 111543

Fixed a class of `OVERRUN` false negatives where `strncpy` was called with a fixed-sized destination buffer and the number of characters to be copied was specified by a function argument.

SAT-23097, 109669

Fixed RESOURCE_LEAK false negatives by enhancing the models for `sqlite_exec` and `sqlite3_exec` functions.

SAT-21656, 103571

Fixed a class of OVERRUN false positive with `min` calculations.

SAT-21285, 101611

Fixed a class of BAD_SHIFT false positives where the shifted bit count was the result of subtracting two correlated values.

SAT-18553, 88365

Improved the OVERRUN checker to report cases where multiple different arguments were used to access the same buffer.

SAT-16427, 79412; SAT-12701, 64883

Improved the OVERRUN checker to report more cases where a buffer is accessed within a loop in a callee.

SAT-13527, 68031

Fixed a class of OVERRUN false positives where the buffer index was modified in a callee.

SAT-12095, 62476

Improved the USE_AFTER_FREE checker to report some cases where a freed pointer is compared to another pointer in a callee.

SAT-6039, 37223

Fixed RESOURCE_LEAK false positives where a pointer to a struct was stored by copying a pointer to its first field.

SAT-5906, 36321

Improved the OVERRUN checker to report more cases where getter functions are used.

SATW-2822

Fixed a crash that occurred when multiple MISRA configuration files that contained HIS Metrics were passed to `cov-analyze`.

SATW-2704

Fixed a false positive in MISRA C-2012 Rule 11.9 where a macro `NULL` was used as a null pointer constant.

SATW-2682

Corrected the classification of two MISRA rules in the Checker Reference Guide.

SATW-2661

Fixed a false positive in CERT EXP37-C where an incorrect number of arguments was reported for C `++` code.

SATW-2659

Fixed a false positive in CERT PRE31-C where a function calls `assert`.

SATW-2655

Improved the error message for CERT DCL37-C so that it is clearer.

SATW-2654

Fixed a false positive in CERT PRE31-C, where a function call was misinterpreted as invoking an unsafe function-like macro.

SATW-2653

Fixed a false positive in CERT INT32-C involving a cast to a wider integer type.

SATW-2652

Fixed a false positive in CERT INT32-C, where defects were reporting on already checked expressions.

SATW-2651

Fixed a false positive in CERT EXP37-C involving Coverity's internal types.

SATW-2650

Fixed a false positive in CERT EXP36-C that occurred while converting pointers of a struct type that had the same alignment.

SATW-2630

Fixed some false positives that occurred in MISRA C++-2008 Rule 3-2-2 and Rule 3-2-4 when a `static const` class member with an in-class initialization was present.

SATW-2606

Fixed a false positive in MISRA C-2012 Rule 2.2 for a function that contained an intentional infinite loop with a side effect.

SATW-2597

Fixed a false positive in MISRA C-2012 Rule 13.2 when calling a function with two function calls without side effects as arguments.

SATW-2554

Fixed a crash that occurred when CERT-C checkers were enabled.

SATW-2527

Enhanced the defect presentation of MISRA C-2012 Rule 10.1.

SATW-2525

Fixed a MISRA C++-2008 Rule 0-1-10 false positive, where a global or static class object was defined.

SATW-2516

Improved the defect presentation of MISRA C-2012 Rule 10.4.

SATW-2512

Fixed an "Invalid property" assertion failure for `cov-analyze`.

SATW-2511

Fixed a memory leak in the CERT DCL58-CPP check.

SATW-2506

Fixed some `EVALUATION_ORDER` and MISRA 2012 Rule 13.2 false positives that claimed volatile access on variables declared with a custom alignment.

SATW-2505

Fixed a false negative in MISRA C-2012 Rule 19.2 where a keyword was not reported when being used without a typedef.

SATW-2479

Added a new coding standards configuration file to turn on CERT-C rules that CERT requires to run with CERT-C++.

SATW-2478, 121782

Fixed an issue that caused MISRA C-2012 Rule 17.2 to crash when it was run with languages other than C/C++.

SATW-2472, 121610

Fixed several false negative cases in MISRA C-2012 Rule 14.3 involving constant conditional expressions.

SATW-2376, 120361

Fixed a false positive case for MISRA C-2012 Rule 7.2 involving literals in assembly code.

SATW-2365, 120407

Fixed a false positive in MISRA C++-2008 Rule 0-1-10 where a destructor was called.

SATW-2187, 120047

Fixed false negatives of CERT FIO46-C with a new checker.

SATW-2185, 120045

Fixed a false negative in CERT ENV31-C where the environment pointer was accessed indirectly through a function call.

SATW-2180, 120038

Fixed an issue reporting on simple cases for CERT STR32-C.

SATW-2179, 120037

Fixed a false negative case which uses pointer arithmetic to access structure members in CERT ARR37-C.

SATW-2177, 120035

Fixed a false negative in CERT-EXP32-C where the assignment allowed the valid code to reference the value of the volatile object through a nonvolatile reference.

SATW-2172, 119985

Fixed a false negative in CERT PRE31-C involving an assignment from an unsafe function-like macro.

SATW-2171, 119979

Fixed a false positive in MISRA C 2012 Dir 4.4.

SATW-2166, 119665

Fixed a false positive in MISRA C-2012 Rule 10.3 where the initializer `{0}` could be used to initialize an aggregate or union type.

SATW-2163

Fixed a false positive in MISRA C-2012 Rule 5.2 where the language standard was not correctly handled.

SATW-2162

Updated the MISRA 2012 Rule 22.8 and MISRA 2012 Rule 22.9 checkers to detect direct `errno` modification within a function.

SATW-1400, 109708

Fixed a false positive in MISRA C++-2008 Rule 0-1-4, which declared a variable with an initialization, assigning it a new value.

SATW-1374, 108970

Fixed a false positive in MISRA C-2012 Directive 4.8 where there is no pointer to the structure or in the translation unit.

73.3.1.3. Known issues and solutions

COVP-1717, 81025

Older versions of FlexLM might produce the wrong hostid version. Please make sure you have installed a version of FlexLM which supports Windows 10.

INS-1694, 84234: Installing into an existing folder

Coverity Analysis cannot be installed into an existing empty folder. Please select a non-existing folder.

INS-1792, 89937

The Coverity Analysis installer fails when the installer path contains Japanese characters.

SAT-26530

On 64-bit Windows platforms, the length of the command string that can be passed to the Fortran syntax analyzer is limited (internally) to 32768 characters. If this limit is exceeded, `cov-run-fortran` fails and reports an "Argument list too long" error.

SAT-26651

Coverity Fortran Syntax Analysis fails with a memory access violation when run under Clear Linux 4.14-64. There is a possible incompatibility with the Fortran runtime library on the Clear Linux platform.

SAT-27581

When `--webapp-security-aggressiveness-level` is set to `high`, it has the effect of setting the `distrust_all` checker option for many checkers. In this case, trusting individual taints using `--trust-<taint-type>` options does not override the `distrust_all` checker option. Note that `--enable-audit-mode` sets `--webapp-security-aggressiveness-level=high` by default. This describes the current behavior. It might change in future releases and should not be relied upon.

73.3.2. Coverity Analysis checkers and user directives in 2018.12

The following sections describe new and updated features, bug fixes, and known issues for Coverity checkers and associated elements.

73.3.2.1. New and updated checkers and directives

MISRA C 2004 and MISRA C 2012 compliance standard support

We've added MISRA C 2004 and MISRA C 2012 compliance standard support for the Clang compiler. (CMPCPP-7133, 115796)

The following table lists new checkers and the languages they support.

Checker	Languages
ANGULAR_BYPASS_SECURITY	JavaScript, TypeScript
ANGULAR_ELEMENT_REFERENCE	JavaScript, TypeScript
BLACKLIST_FOR_AUTHN	Ruby
CONFIG.SEQUELIZE_ENABLED_LOGGING	JavaScript
CSS_INJECTION	JavaScript
DC.PREDICTABLE_KEY_PASSWORD	C/C++
DYNAMIC_OBJECT_ATTRIBUTES	Ruby
FLOATING_POINT_EQUALITY	C/C++
INSECURE_DIRECT_OBJECT_REFERENCE	Ruby
INSUFFICIENT_LOGGING	JavaScript
LOCALSTORAGE_WRITE	JavaScript, TypeScript
PRINTF_ARGS	C/C++
RAILS_DEFAULT_ROUTES	Ruby
RAILS_DEVISE_CONFIG	Ruby
RAILS_MISSING_FILTER_ACTION	Ruby
REGEX_MISSING_ANCHOR	Ruby
RUBY_VULNERABLE_LIBRARY	Ruby
SESSION_MANIPULATION	Ruby
TRUST_BOUNDARY_VIOLATION	Java, C#, Visual Basic
UNESCAPED_HTML	Ruby
UNSAFE_SESSION_SETTING	Ruby
UNSAFE_BASIC_AUTH	Ruby
URL_MANIPULATION	JavaScript

The following table documents added language support for existing checkers.

Languages	Checkers	Checkers
C/C++	FLOATING_POINT_EQUALITY	PRINTF_ARGS

Languages	Checkers	Checkers
	OS_CMD_INJECTION PATH_MANIPULATION	SQL_INJECTION
Ruby	BAD_CERT_VERIFICATION CSRF DIVIDE_BY_ZERO HARDCODED_CREDENTIALS INSECURE_COOKIE OPEN_REDIRECT OS_CMD_INJECTION PATH_MANIPULATION REGEX_INJECTION	RESOURCE_LEAK SCRIPT_CODE_INJECTION SENSITIVE_DATA_LEAK SQLI UNSAFE_DESERIALIZATION UNSAFE_REFLECTION WEAK_PASSWORD_HASH XSS

New and changed checkers

AUDIT.SPECULATIVE_EXECUTION_DATA_LEAK

AUDIT.SPECULATIVE_EXECUTION_DATA_LEAK has a new option:
`speculative_uninitialized_use:<bool>`

If true, the checker will report defects if the index used in a nested memory access is only initialized inter-procedurally. Depending on the code produced by the compiler, a speculative store bypass might occur. This would allow the memory access to occur before the initialization, resulting in the use of an uninitialized value as the load address. This could enable an attacker to read sensitive information. (SAT-27561)

ANGULAR_BYPASS_SECURITY

Calls to the `bypassSecurityTrust*` functions from the Angular DomSanitizer API. (SAT-27672)

ANGULAR_ELEMENT_REFERENCE

Reports uses of the `ElementRef` API where the underlying DOM element is accessed and used in a sensitive way. (SAT-27672)

BLACKLIST_FOR_AUTHN

Reports defects when a filter on a web controller is specified using a list of actions to which the filter applies, rather than a list of actions to which the filter does not apply. (SAT-27000)

CONFIG.SEQUELIZE_ENABLED_LOGGING

Finds cases where a `sequelize` connection is created with logging enabled. In this case, SQL queries would be logged to the console and might leak sensitive data because console outputs are often streamed to log files when the application is deployed. (SAT-27605)

CSS_INJECTION

Reports a defect when a user-controlled string is able to modify the CSS of an HTML element. (SAT-26373, 121614)

DC.PREDICTABLE_KEY_PASSWORD

Detects calls to crypto APIs that result in the generation of weak or predictable keys. (SAT-26050)

DYNAMIC_OBJECT_ATTRIBUTES

Finds vulnerabilities that occur when a resource is updated with attribute names and values using uncontrolled dynamic data (CWE-915).

FLOATING_POINT_EQUALITY

Reports on floating-point expressions being tested for equality or inequality. This checker is adapted from MISRA C++2008 Rule 6-2-2. (SAT-25993, 119829)

INSECURE_DIRECT_OBJECT_REFERENCE

Finds code that might allow attackers to directly retrieve records via a simple identifier (CWE-639).

INSUFFICIENT_LOGGING

Reports a defect in code that handles a security event or error condition but does not properly log the event. Logging important security events facilitates the earlier detection of security incidents and a better response to them. (SAT-27562)

LOCALSTORAGE_WRITE

Reports whenever any data is written to `localStorage`. (SAT-27672)

OMR_NULL_LOAD

The `OMR_NULL_LOAD` checker is now enabled by default. (SAT-24594, 115330)

OS_CMD_INJECTION

Trust options are not supported for C and C++ in this release.

(They have been documented in the localized versions of the Checker Reference.)

Note

Some of the defects reported by this checker were previously reported as `TAINTED_STRING` defects.

(SAT-28029)

PATH_MANIPULATION

Trust options are not supported for C and C++ in this release.

(They have been documented in the localized versions of the Checker Reference.)

Note

Some of the defects reported by this checker were previously reported as `TAINTED_STRING` defects.

(SAT-28029)

PRINTF_ARGS

Reports on invalid printf format strings, or invalid arguments to those strings. (SAT-23905, 112760)

RAILS_DEFAULT_ROUTES

Identifies vulnerabilities resulting from the failure to mark a controller method as private. (SAT-27000)

RAILS_DEVISE_CONFIG

Reports on a number of best practices when configuring a Ruby on Rails application using the Devise authentication library. (SAT-26285)

RAILS_MISSING_FILTER_ACTION

Finds code where a filter specifies an action that does not exist. (SAT-26285)

REGEX_MISSING_ANCHOR

Finds regular expressions where proper anchors to the beginning and end of the string are not specified (CWE-777).

RUBY_VULNERABLE_LIBRARY

Reports a defect if your application uses a library that might be affected by one of a given set of Ruby-on-Rails related vulnerabilities. (SAT-26998)

SESSION_MANIPULATION

Indicates that uncontrolled dynamic data is used to specify a key in a session. (SAT-27000)

SQLI

Trust options are not supported for C and C++ in this release.

(They have been documented in the localized versions of the Checker Reference.)

 **Note**

Some of the defects reported by this checker were previously reported as TAIANTED_STRING defects.

(SAT-28029)

TAIANTED_SCALAR

Trust options are not supported for this release.

(They have been documented in the localized versions of the Checker Reference.)

(SAT-28029)

TAIANTED_STRING

Some defects that were previously reported by this checker are now reported by the following checkers:

OS_CMD_INJECTION

PATH_MANIPULATION

SQL_INJECTION

 **Note**

Trust options are not supported for this release. (They have been documented in the localized versions of the Checker Reference.)

(SAT-24466)

TRUST_BOUNDARY_VIOLATION

Reports a defect when tainted data is stored in a location that is generally trusted. (SAT-19949, 93494)

UNESCAPED_HTML

Reports possible instances of cross-site scripting vulnerabilities. (SAT-27000)

UNSAFE_BASIC_AUTH

Reports use of Basic Authentication: the Basic Authentication scheme sends unencrypted credentials with every request from the web browser to the web server. (SAT-27000)

UNSAFE_SESSION_SETTING

Reports unsafe settings related to web server sessions. (SAT-27000)

URL_MANIPULATION

Detects instances where a URL or URI is constructed unsafely. (SAT-26541)

XPATH_INJECTION

Trust options are not supported for C and C++ in this release.

(They have been documented in the localized versions of the Checker Reference.)

(SAT-28029)

XSS

The `xss` checker is now also supported for Ruby. (for SAT-26997)

73.3.2.2. Known issues and solutions

SAT-17490, 84256: `INTEGER_OVERFLOW` churn

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

SAT-7224, 43971: `xss`

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

73.3.3. Compiler configuration, Build capture, and Compiler Integration Toolkit (CIT) 2018.12

This section lists new features, bug fixes, and known issues related to Coverity-supported compilers (including configuration), and the Compiler Integration Toolkit (CIT).

73.3.3.1. Important Compiler Integration Toolkit (CIT) information

There were several deprecations and EOLs for Compiler Integration Toolkit (CIT) this release:

- Deprecated support for NetBSD 6.1 and earlier. (CMPG-2889)
- Deprecated support for the HI-TECH PICC compiler. (CMPG-2856)
- Deprecated support for the TriMedia TCS compiler. (CMPG-2855)
- Deprecated support for the SNC C/C++ and SNC GNU C/C++ compilers. (CMPG-2854)
- Deprecated support for Visual Studio 2008. (CMPG-2853)
- Deprecated support for GNU GCC 2.x. (CMPG-2852)
- Dropped support for Apple Clang versions 2.1-5.1. (CMPG-2812, 120910)

73.3.3.2. New and changed features

- Added support for the Green Hills Optimizing C and C++/EC++ ARM 2015.1.4 compiler. (CMPCPP-7386, 118589)
- Added support for ARM Clang 6.10.1. (CMPCPP-7582, 121633)
- Added support for ARM NEON builtin types and intrinsic functions for gcc compilers. (CMPCPP-6477, 104248)
- Added support for C++17. (CMPG-2182, 85696)
- JavaScript files containing decorator syntax are now supported for capture and emit. (CMPJS-549)
- The JavaScript front end now supports the ECMAScript 8 `async` function and `await` syntax. (CMPJS-423, 103284)
- JavaScript source code containing JSX syntax, which is often used for writing React applications, is now supported for capture and emit. Note that source code using Flow syntax is not supported. (CMPJS-649; CMPJS-525; CMPJS-415, 101071)

We've also added the `-no-jsx` option to the `cov-configure` command. This option disables the filesystem capture of JSX files.

- Capture and emit of Angular application code, written in TypeScript, are now supported. (CMPJS-185, 92790)
- Capturing and emitting TypeScript source files is now supported. (CMPJS-184, 92789)

Added the new option `--no-typescript` to the `cov-configure` command. This option disables the filesystem capture of TypeScript files.

73.3.3.3. Bug fixes

The following bugs are fixed for compilers and the Compiler Integration Toolkit (CIT) for Coverity Analysis analysis in 2018.12:

CAP-1294, 117069

An issue where `cov-build` could miss compiler invocations in some Mac OS builds has been resolved.

CMPCPP-8085

Fixed an "Unexpected attempt to load an enum definition" assertion.

CMPCPP-8116

Fixed an assertion in `cov-emit` when emitting the initializer for a non-trivial type array with an explicit zero bound.

CMPCPP-8147; CMPCPP-7761

Fixed an issue in the GCC configuration that caused spurious errors when using certain intrinsics.

CMPCPP-7984

Fixed a parsing error and subsequent error recovery crash in `cov-emit` involving a failure to resolve an overloaded function when assigning a member function pointer.

CMPCPP-7950

Fixed a spurious "no instance of overloaded function matches the specified type" error in `cov-emit` that could occur in C++17 mode when specializing a template class template constructor.

CMPCPP-7945

Fixed an issue where `cov-emit` sometimes produced the wrong value for `std::is_pod` in a class with inheriting constructors.

CMPCPP-7931

An internal error that resulted in an assertion failure, stating "missing default rescan info" for `edg/src/exprutil.c`, has been fixed.

CMPCPP-7890

There was an issue with the CERT PRE30-C defect detection when compiling with an MSVC compiler and the C language level was left unspecified. This has been fixed.

CMPCPP-7845

In the WindRiver Diab compiler support, identifiers beginning with "packed" were being misinterpreted as using the packed keyword. This has been corrected.

CMPCPP-7842

Functions containing ASM statements were not being properly emitted in Windows. There was also no specific parse error message. Instead, there was only a message indicating that the function was not emitted. This has been corrected.

CMPCPP-7823

An `edg/src/class_decl.c` assertion failure, which occurs in C++ code when using a Microsoft extension, has been fixed.

CMPCPP-7759

Use of the Microsoft `__super` extension no longer results in failures when captured code is compiled with Clang.

CMPCPP-7742, 121092

Fixed an emit filename corruption that occurred on Windows when a file is included with an absolute path without drive specification in a PCH file.

CMPCPP-7710

Fixed a spurious redeclaration error in `cov-emit` that could occur in the presence of a prior using declaration for the same method.

CMPCPP-7708

Fixed an issue in the GCC compiler configuration that caused errors when using various STL templates added in C++17, such as `std::void_t` and `std::disjunct`.

CMPCPP-7696

Fixed a spurious error that occurred in `cov-emit` when emulating versions of GCC newer than 4.2. The error would occur for an out-of-class definition of a nested template class member.

CMPCPP-7666

Fixed an issue where the wrong value for the `__cplusplus` macro was used for some Intel compilers.

CMPCPP-7640

An `edg/src/lower_name.c` assertion failure has been fixed.

CMPCPP-7636

Fixed an internal error in `cov-emit` that occurred when an assertion was triggered during error recovery.

CMPCPP-7626

Fixed an internal error that occurred in `cov-emit` when instantiating a template that contained a `constexpr friend` declaration.

CMPCPP-7565, 121330

The `cov-emit` database was getting corrupted when parse messages exceeded 100M. This has been fixed.

CMPCPP-7563, 121321

An assertion failure with `edg/src/expr.c` has been fixed.

CMPCPP-7501, 120533

Fixed a spurious error that could occur in `cov-emit` when using the `xmemory` header with Visual Studio 2017 in `/std:c++17` or `/std:c++latest` mode.

CMPCPP-7457, 119667

Fixed a `cov-emit` crash that resulted from mangling anonymous structs in a template function.

CMPCPP-7396, 118697

Fixed a spurious error in `cov-emit` that occurred when template argument lists were followed by `>>`.

CMPCPP-7334, 117732

Fixed an issue where `cov-emit` would crash when the compiler was parsing template constructors.

CMPCPP-6494, 104541

Fixed a crash in `cov-internal-emit-clang`, which involved C99 designated initializers for aggregate fields.

CMPCPP-6486, 104410

Fixed an assertion failure that occurred for Clang compilers when the subscript operator was used on an rvalue expression of a vector type.

CMPCPP-6429, 103648

The use of class template deduction guides no longer results in a compilation failure.

CMPCPP-6377, 102835

Fixed a spurious error in `cov-emit` when initializing a `constexpr` static data member of a class marked `dllimport`.

CMPCPP-6333, 102290

Fixed an issue where `__attribute__((packed))` was not being respected on enums.

CMPCPP-5969, 96622

Added support for the `-fgnu89-inline` switch to the GCC configuration.

CMPCPP-5837, 94956

Fixed a parse error in `cov-emit` involving variadic templates and templates with default arguments.

CMPCPP-5745, 93038

Declaring a member function with the same name as a base member function no longer causes parse failures with Microsoft compilers.

CMPCPP-5634, 90675

Fixed an issue where `__builtin_va_list` was unrecognized when emulating some Intel compilers.

CMPCPP-3493, 55666

Use of C++11 generalized attributes (such as `[[clang::fallthrough]]`) no longer results in failures when capture code is compiled with Clang-based compilers.

CMPCSH-928

Addressed several crashes in `cov-emit-cs` related to using local functions with lambdas and delegates. Most common cases appeared as "Unknown serf pointer" and "assertion failed: `delegatedFn->is_static_method() || delegatedFnInstanceExpr`".

CMPCSH-994

Fixed an issue with Coverity Analysis where a nullable literal value in C# would trigger an assertion in `cov-analysis`.

CMPG-2830; CMPCPP-8033

C++14 and C++17 support has been significantly improved.

CMPG-2845

Fixed an error when using the replay feature with a Clang-based compiler and `--emit-complementary-info` option (which is activated when using compliance standards).

CMPVB-47

Addressed several crashes in `cov-emit-vb` that caused time-consuming error recovery invocations.

CMPJ-1041

Fixed a `NullPointerException` in `cov-emit-java` that could occur when compiling non-modular code with JAR files that contain module definitions. This could result in entire compilations being lost.

CMPJ-1011, 118603

Fixed a Java version detection failure when the `JAVA_TOOL_OPTIONS` or `_JAVA_OPTIONS` environment variable was set in the native build environment.

CMPJ-980, 116304

Removed the `--enable-java-annotation-framework-support` option from `cov-build`. By default, this behavior is now enabled.

73.3.3.4. Known issues and solutions

CAP-1176, 97630

`cov-build --instrument` has a known issue when running the `xdcmake.exe` tool of Visual Studio 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 either:

- Modify the build such that `xdcmake.exe` is run from a 64-bit process.
- Ignore the `xdcmake.exe` process by adding `--capture-ignore xdcmake.exe` to your `cov-build` invocation.

CMPJ-368, 65669, 65721

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 `LANG` or `LC_CTYPE` environment variables.

CMPJS-286, 95651

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 used with some frameworks.

73.3.4. Coverity Analysis 2018.12 Commands

73.3.4.1. Commands related to the build process

This section lists new features, bug fixes, and known issues for `cov-build` and related commands, including `emit` and `translate` commands.

73.3.4.1.1. New and changed features

The following new and changed features were added for commands related to the build process (including emit and translate commands) in 2018.12:

- The `cov-build` command has added the `--js-template-da` option: When specified, it causes the Javascript template dynamic-analysis to be run for all directories specified in `--fs-capture-search`. (SAT-27752)
- The new `cov-capture` command allows you to run a "buildless" capture of your source. That is, it captures source code without requiring a build. For more information, see the Coverity Analysis User and Administrator Guide. (BLC-212)
- Previously, `cov-capture` was used for capturing Test Advisor test coverage. That functionality has been consolidated into `cov-build --test-capture`. See the Command Reference for more information. (BLC-87)

73.3.4.1.2. Bug fixes

There were no bugs fixed for build-related commands (including emit and translate commands) in 2018.12.

73.3.4.1.3. Known issues and solutions

Build-related commands have the following known issues and solutions:

CMPCPP-4764, 72964

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.

CAP-812, 64428

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 hang occurs, 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 either:

- Uninstall KB2919355

OR

- Add the `--instrument` flag to your `cov-build` invocation:

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

SAT-12174, 62745

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.

CAP-332, 26881, 38175

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

Error message:

```
[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
```

73.3.4.2. Commands related to analysis

This section lists new features, bug fixes, and known issues for `cov-analyze` and related commands.

73.3.4.2.1. New and changed features

The following new features were added or changed for commands related to the analysis process in 2018.12:

- The `--enable-audit-mode` option has been added to the `cov-analyze` command. This option sets the impact level to `Audit` and enables audit-mode analysis, which is intended to expose more potential security vulnerabilities by considering additional potential data sources that could be used in an exploit. It sets `--webapp-security-aggressiveness-level=high` and `--distrust-all`, enables four additional checkers, and models additional taint sources in supported languages. (SAT-27768, IM-23168)
- The `cov-analyze` command has two new options:
 - `--enable-brakeman`, which enables Brakeman Pro checkers (Default)
 - `--disable-brakeman`, which disables Brakeman Pro checkersThese two new options are used to enable and disable the new Ruby checkers. (SAT-27544)

The `cov-analyze` command has two new options:

- `--enable-brakeman`, which enables Brakeman Pro checkers (Default)
- `--disable-brakeman`, which disables Brakeman Pro checkers

These two new options are used to enable and disable the new Ruby checkers. (SAT-27544)

73.3.4.2.2. Bug Fixes

There are no bug fixes for analysis-related commands in 2018.12.

73.3.4.2.3. Known issues and solutions in 2018.12

Analysis-related commands have the following known issues and solutions:

CMPG-1741, 70845, 71216

The `cov-run-desktop` command sometimes fails on large Java compilations, potentially causing emit database corruption on Windows platforms. This can manifest as a `cov-analyze` crash. More commonly, `cov-emit-java` itself will fail with access violation crashes or errors concerning a failure to acquire a lock. These will appear in `cov-run-desktop-log.txt`. If this issue occurs, you can work around it by specifying `-j 1` with `cov-run-desktop`.

73.3.4.3. Commands related to Test Advisor

This section lists new features for Test Advisor.

73.3.4.3.1. Bug fixes in 2018.12

There were no bugs fixed for Test Advisor-related commands in this release.

73.3.5. Coverity Wizard 2018.12

This section lists new features, bug fixes, and known issues related to Coverity Wizard.

73.3.5.1. Important Coverity Wizard Information

There have been no deprecations or EOLs this release.

73.3.5.2. New and changed features

Coverity Wizard has the following new and changed features in 2018.12:

- Added support for OSX 10.14 to Coverity Wizard. (PRD-10591, 121360)

73.3.5.3. Bug fixes

The following bugs were fixed for Coverity Wizard in 2018.12:

PRD-10618

Added an option, allowing customers to disable all `webapp-security` checkers.

PRD-10555

Fixed an issue where `cov-wizard` wouldn't allow users to resize the **Edit File List** window in the *Capture* step of Coverity Wizard.

INS-1832

The user is no longer required to install JVM6 on OSX.

73.3.5.4. Known issues and solutions

Coverity Wizard has the following known issues in version 2018.12:

PRD-9245, 90621

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

PRD-9208, 90489

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-8453, 83450

When using a self-signed certificate, if the user chooses not to trust a certificate, they might be prompted multiple times in a row (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-8227, 82196

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-7595, 77742

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.

PRD-6832, 70361

The guided policy creation wizard "Documentation" link fails to open properly on Linux. Open the *Coverity Wizard 2020.12 User Guide* separately to view this documentation.

PRD-6760, 69815

The *Guided Test Advisor Policy Creation Wizard* uses Java regex validation instead of the Perl regex validation that Coverity Analysis Test Advisor users. 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-5770, 59676

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-5387, 54143

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

PRD-5290, 53367

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.

73.3.6. Test Advisor 2018.12

Coverity Test Advisor is a component of the Coverity Analysis installation package.

73.3.6.1. New and changed features

Test Advisor has the following new and changed features in 2018.12:

- Support for the Accurev source control management system has been extended to Accurev version 7.2. (TADE-1958)
- Support for Team Foundation Server 2018 has been added. (TADE-1928)
- Automatic version detection for Team Foundation Server has been added. It can be used by replacing `--scm tfs2013`, `--scm tfs2015`, etc., in your invocations of Coverity tools with `--scm tfs`. (TADE-1928)

73.3.6.2. Bug fixes

The following bugs are fixed for Test Advisor in 2018.12:

TADE-1704, 89465

Previously, the Coverity SCM tools (`cov-extract-scm`, `cov-import-scm`, `cov-blame`, `cov-run-desktop`) would fail to parse output from svn if `--use-merge-history` was being passed (via `--command-arg --use-merge-history`, `--scm-command-arg --use-merge-history`, or `--scm-param annotate_arg=--use-merge-history`). This has been fixed.

73.3.6.3. Known issues and solutions

Test Advisor has the following known issues and solutions in 2018.12:

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.

73.3.7. Dynamic Analysis 2018.12

Dynamic Analysis is a component of the Coverity Analysis installation package.

73.3.7.1. Known issues and solutions

Dynamic Analysis has the following known issues and solutions in 2018.12:

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.

JDA-681, 20788

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

JDA-694, 21417

Specifying certain combinations of the `instrument-arrays`, `instrument-collections`, `detect-races`, and `detect-deadlocks` options to the Dynamic Analysis agent have 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, then Dynamic Analysis reports races on neither collections nor arrays.

JDA-720, 22148

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

73.3.8. Architecture Analysis 2018.12

Coverity Architecture Analysis is a component of the Coverity Analysis installation package.

73.3.8.1. Important Architecture Analysis information

Architecture Analysis has no changed features and support in 2018.12.

73.3.9. Extend SDK 2018.12

Coverity Extend Software Development Kit is a component of the Coverity Analysis installation package.

73.3.9.1. Known issues and solutions

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.

73.4. Coverity Desktop 2018.12

The Coverity Desktop plug-in is available for various platforms from the Coverity Connect *Downloads* menu.

73.4.1. Important information

There have been several deprecations and EOLs this release:

- Coverity Desktop support has been dropped for Eclipse 4.5. (PRD-10681)
- Coverity Desktop support has been deprecated for Android Studio 2.2. (PRD-10681)
- Coverity Desktop support has been deprecated for IntelliJ IDEA 2016.1 and 2016.3. (PRD-10681)
- Coverity Desktop support has been deprecated for the 2016.3 version of RubyMine, WebStorm, PyCharm, and PhpStorm. (PRD-10681)
- Dropped support for TFS 2008. (PRD-10650; PRD-10648)

Deprecated support for TFS 2010.

- Deprecated the `misra_config` settings. Users can use the `coding_standard_config` option instead. Note that the `coding_standard_config` option supports multiple files. (PRD-10584, 120571)

73.4.2. Coverity Desktop for Eclipse in 2018.12

73.4.2.1. New and changed features

Coverity Desktop for Eclipse has the following new and changed features in 2018.12:

- Added support Eclipse 4.9. (PRD-10683)
- Added support for `.c` files. (PRD-10677)
- Added support for OSX 10.14 to the Eclipse Coverity Desktop plugin. (PRD-10695; PRD-10591, 121360)
- The Coverity Visual Studio extension now supports multiple *Coding Standard Configuration* files. (PRD-10583)
- Added support for Android Studio 3.2. (PRD-10560)
- *Security Audit* issues from a remote stream can now be viewed in the Coverity plugins.
In Visual Studio, these issues will be displayed without an impact icon. (PRD-10549)
- Coverity Visual Studio Extension now supports .NET Core projects. (PRD-10446)

73.4.2.2. Bug fixes

There following bugs were fixed for Coverity Desktop for Eclipse in 2018.12:

PRD-10666

Any *Coding Standard Configuration* setting from the `coverity.conf` file was displayed as read-only in the **Analysis Page** (under **Analysis Configuration**). This has been fixed.

PRD-10598

Fixed an issue where the *IssuesView* column did not properly display.

73.4.2.3. Known issues and solutions

Coverity Desktop for Eclipse has the following known issues in version 2018.12:

PRD-10711

Eclipse customers using Plastic SCM may see a failure during **Analyze Modified Files**, as Eclipse is unable to locate their `cm.exe` 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/`.

PRD-10694

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

73.4.3. Coverity Desktop for Microsoft Visual Studio in 2018.12

73.4.3.1. New and changed features

Coverity Desktop for Microsoft Visual Studio has the following new and changed features in 2018.12:

- Coverity Desktop now supports Microsoft TFS 2018.

The user interface has also been modified so that users are no longer required to specify the TFS version for SCM analysis. (PRD-10630; PRD-10629)

73.4.3.2. Bug fixes

Coverity Desktop for Microsoft Visual Studio has the following bug fixes in 2018.12:

PRD-10712

Fixed an encoding issue which caused Visual Studio to crash.

PRD-10701

As of the Pacific release, the user no longer needs to select a specific TFS version in order to use TFS for SCM analysis.

PRD-10603

The Coverity Visual Studio extension was crashing due to an invalid OEM code page, which was used to retrieve the encoding. We've fixed this by using default encoding in the OEM code page.

PRD-10602

Fixed a `System.InvalidOperationException` error message which was thrown when a wait dialog was displayed.

73.4.3.3. Known issues and solutions

Coverity Desktop for Visual Studio has no known issues in version 2018.12.

73.4.4. Coverity Desktop for IntelliJ IDEA in 2018.12

73.4.4.1. New and changed features

Coverity Desktop for IntelliJ IDEA has the following new and changed feature(s) in 2018.12:

- Added support for IntelliJ 2018.2, CLion 2018.2, PhpStorm 2018.2, PyCharm 2018.2, RubyMine 2018.2, and Webstorm 2018.2. (PRD-10617; PRD-10559)
- Added support for Java 9+ project module path. (PRD-10610; PRD-10609)
- Added support for OSX 10.14 to the IntelliJ Coverity Desktop plugin. (PRD-10591, 121360)

73.4.4.2. Bug fixes

Coverity Desktop for IntelliJ has no bug fixes in 2018.12.

73.4.4.3. Known issues and solutions

Coverity Desktop for IntelliJ IDEA has the following known issues in version 2018.12:

PRD-10553, 119444

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-10076, 105052

When using whole program checkers in IntelliJ, a warning about missing class files might be seen 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-8397, 83106

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

PRD-8042, 80698

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.

PRD-8038, 80693

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

PRD-7991, 80599

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-7980, 80573

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

PRD-7453, 76907

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.

73.5. Coverity Documentation 2018.12

The following new documents and changes were made in 2018.12:

RG-1052

You can now generate a security report from a script. For more information, see the chapter on Security reports in the Coverity Platform User and Administrator Guide.

RG-1048

The OWASP Mobile Top 10 report has been added. This report details the assessments that were done, provides a summary of findings, and specifies the remediations needed. Information from this report is of special interest to application security assurance teams and their clients.

RG-1047

The PCI DSS report has been added. It analyzes your source and reports violations of standards defined by the Payment Card Industry Data Security Standard (PCI DSS), which was developed to encourage and enhance cardholder data security and facilitate the broad adoption of consistent data security measures globally.

RG-1046

The OWASP Top Ten report was added. It details the assessments that were done, provides a summary of findings, and specifies the remediations needed. Information from this report is of special interest to application security assurance teams and their clients.

SAT-27606; SAT-20558, 96584; CMPG-2864

Added instructions for running a security analysis on an ASP.NET Core Web application. See "2.2.2. Running a security analysis on an ASP.NET Web application" in the Coverity Analysis 2018.09 User and Administrator Guide for more information.

SAT-26930

Supported standards per language (for example, CWE Top 25) are now documented in "Chapter 1.3. Language Support" of the Coverity Analysis 2018.12 User and Administrator Guide.

Chapter 74. Coverity 2018.09-15 Release Notes

Table of Contents

74.1. Coverity Analysis bug fixes	274
---	-----

74.1. Coverity Analysis bug fixes

CMPCPP-9164

A performance problem related to XREF generation has been corrected.

Chapter 75. Coverity 2018.09-14 Release Notes

Table of Contents

75.1. Coverity Compiler Integration Toolkit (CIT) bug fixes 275

75.1. Coverity Compiler Integration Toolkit (CIT) bug fixes

CMPCPP-9040

Fixed a case of emit DB corruption triggered by use of GNU multiversioning on a member function.

Chapter 76. Coverity 2018.09-13 Release Notes

Table of Contents

76.1. Coverity Analysis bug fixes	276
---	-----

76.1. Coverity Analysis bug fixes

SATW-3066

Fixed a false positive in CERT EXP62-CPP that occurred when using `memset` on an array of pointers to class objects.

SATW-3068

Fixed a false positive in CERT FLP32-C where user-defined functions were mistaken for standard math functions.

SATW-3070

Fixed a false positive in CERT OOP51-CPP where an expected object slicing did not occur.

SATW-3072

Fixed a false positive in CERT OOP57-CPP that occurred when using `memset` on an array of pointers to class objects.

Chapter 77. Coverity 2018.09-12 Release Notes

Table of Contents

77.1. Coverity Analysis bug fixes	277
---	-----

77.1. Coverity Analysis bug fixes

IM-23890

Fixed occasional commit failures that occurred during high concurrency commit scenarios.

Chapter 78. Coverity 2018.09-11 Release Notes

Table of Contents

78.1. Coverity Compiler Integration Toolkit (CIT) bug fixes	278
---	-----

78.1. Coverity Compiler Integration Toolkit (CIT) bug fixes

CMPCPP-8514

Fixed a case of memory corruption in `cov-emit` when a call was made to an `extern inline` method, and the target of the call was wrapped in parentheses. For example: `(f)()`.

Chapter 79. Coverity 2018.09-9 Release Notes

Table of Contents

79.1. Coverity Platform bug fixes	279
---	-----

79.1. Coverity Platform bug fixes

IM-23263

Fixed a bug in the *Purge Snapshot Details* functionality which caused some Issue occurrences to be lost for snapshots that were not directly purged. With the fix, new snapshots will have correct occurrences, but existing snapshots might still be missing some data.

Chapter 80. Coverity 2018.09-7 Release Notes

Table of Contents

80.1. Coverity Analysis bug fixes	280
---	-----

80.1. Coverity Analysis bug fixes

SAT-28127

Fixed a bug where the `enabled-checkers` setting in the `ANALYSIS.metrics.xml` file did not include regular quality or security checkers even though they were enabled.

Chapter 81. Coverity 2018.09-6 Release Notes

Table of Contents

81.1. Coverity Connect bug fixes	281
--	-----

81.1. Coverity Connect bug fixes

IM-23205

You can now specify email addresses for custom top-level-domains (TLDs) during an LDAP user import.

Chapter 82. Coverity 2018.09-5 Release Notes

Table of Contents

82.1. Coverity Analysis bug fixes	282
---	-----

82.1. Coverity Analysis bug fixes

CMPCPP-8087

Added support for the `-fgnu89-inline` switch to the GCC configuration.

SAT-27794

Fixed an analysis crash caused by unexpected field types during a deep write.

Chapter 83. Coverity 2018.09-4 Release Notes

Table of Contents

83.1. Coverity Analysis bug fixes	283
---	-----

83.1. Coverity Analysis bug fixes

CMP CPP-7935

Added support for the LLVM Clang 7.0, Apple Clang 9.1 (Xcode 9.3/9.4), and Apple Clang 10 (Xcode 10.0) compilers.

CMP J-1064

Fixed a `NullPointerException` in `cov-emit-java` that could be encountered while compiling non-modular code when JAR files on the classpath contained module declarations. This caused entire compilations to be lost.

Chapter 84. Coverity 2018.09-3 Release Notes

Table of Contents

84.1. Coverity Platform bug fixes	284
---	-----

84.1. Coverity Platform bug fixes

IM-23188

This update causes a database incompatibility with releases 2018.09 to 2018.09-3. Please upgrade to 2018.09-4 or later.

IM-23135

Fixed an issue where the *Configuration - Users & Groups* dialog would close immediately after being opened by an `Administrators` group member.

Chapter 85. Coverity 2018.09-2 Release Notes

Table of Contents

85.1. Coverity Analysis bug fixes	285
---	-----

85.1. Coverity Analysis bug fixes

SAT-27193

Fixed the line count metric for function lines of cases where a function was called with a default argument value. In some cases, the location of the default argument's definition would incorrectly be included in the line span, yielding a very large line count.

Chapter 86. Coverity 2018.09-SP1 Release Notes

Table of Contents

86.1. Coverity Analysis bug fixes 286

86.1. Coverity Analysis bug fixes

SAT-27158, 121720

Fixed an issue where writing files larger than 4GB (such as files containing large amounts of defect output) on Windows would cause Coverity to crash.

Chapter 87. Coverity 2018.09 Release Notes

Table of Contents

87.1. Important information for 2018.09	287
87.2. Coverity Platform 2018.09	289
87.3. Coverity Analysis 2018.09	293
87.4. Coverity Desktop 2018.09	302
87.5. Coverity Documentation 2018.09	304

87.1. Important information for 2018.09

Due to a change in our bug tracking system, items are now identified by two bug numbers:

- One reflecting the identity of the bug in our **old** bug tracking system, formatted like this: XXXXXX. (For example, 374568.)
- One reflecting the identity of the bug in our **new** bug tracking system, formatted like this: CODE-XXXXX. (For example, IM-22788.)

 **Note**

Bugs with only a CODE-XXXXX number do not have an old number.

87.1.1. Deprecated and End-of-Life (EOL) Products in Coverity 2018.09

Support for the following products, features, platforms, and third-party tools is classified as deprecated or end-of-life as of the Coverity 2018.09 release.

87.1.1.1. Deprecated Products

Support for the following products and features is deprecated as of the Coverity 2018.09 release.

Table 87.1. Deprecated products

Product	Comments
Accurev 6.0 and 6.1 support	Coverity Test Advisor Support SCM Systems
Apple Clang v2.0 - 5.1 support	Supported Compilers: Coverity Analysis for C/C++
Coverity Connect DesktopDeveloper role	The DesktopDeveloper role has been deprecated and is now replaced with the Developer role.
cov-build option: --treat-as-64bit	The --treat-as-64bit option is deprecated. The related COVERITY_INSTRUMENT_64BIT_EXES variable is also deprecated because --instrument

Coverity 2018.09 Release Notes

Product	Comments
	capture now works in a way that can adapt to 64-bit .NET binaries (that appear to run as 32-bit .NET binaries).
ClearCase v7.0.x, 7.1.x and 8.0.x	Coverity Test Advisor Support SCM Systems
Desktop plugin support for Eclipse v4.5	Coverity Desktop Eclipse platform support
Developer Streams feature	Support for the Developer Streams feature is now deprecated and will be removed in a future release.
GCC on Windows for Extend SDK checker development	For Extend SDK checker development on Windows, use of the GCC compiler (available in the MinGW environment) to compile Extend SDK checkers has been deprecated as of this release. Support will be removed and replaced in a future release.
Linux Kernel 2.6.31 and earlier deprecated for Coverity Analysis	Supported Platforms for Coverity Analysis
Mac OS X versions 10.10 and 10.11 deprecated for Coverity	Supported Platforms for Coverity Analysis
Perforce 2007.2-2014.1	Coverity Test Advisor Support SCM Systems
PostgreSQL 9.5 as external database for Coverity Connect	Coverity Software Requirements
Safari versions no longer supported by Apple	Support for versions of Safari that Apple no longer supports is deprecated for Coverity Connect.
Scratchbox support deprecated for Coverity Analysis	Supported Compilers: Coverity Analysis for C/C++
Windows 7 support deprecated for Coverity Connect	Coverity Desktop Supported platforms
Windows Server 2008 support deprecated for Coverity Connect	Coverity Desktop Supported platforms
Xbox 360 compiler	Supported Compilers: Coverity Analysis for C/C++
Xcode gcc 4.2 and llvm-gcc 4.2 support	Supported Compilers: Coverity Analysis for C/C++

87.1.1.2. End-of-Life Products

Support for the following products and features is dropped in the Coverity 2018.09 release.

Table 87.2. End-of-Life Products

Product	Comments
Java versions 1.5 and 1.6 for Coverity Analysis and Coverity Desktop	Coverity Desktop Supported platforms

Product	Comments
Java 1.7 running the Coverity Desktop plugin within Eclipse, IntelliJ, and Android Studio	Supported Platforms for Coverity Desktop
Java 9 support	Supported Platforms for Coverity Analysis

87.2. Coverity Platform 2018.09

This section provides release notes for Coverity Platform components.

87.2.1. Coverity Connect 2018.09

Coverity Connect is a component of the Coverity Platform installation package.

87.2.1.1. Important Coverity Connect Information

There have been no deprecations or EOLs for Coverity Connect this release.

87.2.1.1.1. New and changed features

Coverity Connect has the following new and changed features:

- **We've integrated Coverity into Synopsys' eLearning system.** User accounts are provided by one of our Sales and Support team members once a Synopsys license has been purchased by your organization. To log in to the eLearning portal, visit elearning.synopsys.com. (COVP-2045, 639930)
- **Added TLSv1.2 support to `cov-manage-im`.** (IM-22209, 116603)
- **The `cov-commit-defects` command has a new option: `--url`.** This option allows you to commit analysis results to a Coverity Connect instance that has a context path in its HTTP(S) URL. This option replaces the `--host`, `--port`, `--https-port`, and `--dataport` options. The `--url` option is provided to accommodate the use of a context path and to deal with the setting up of Coverity Connect behind a reverse proxy. (IM-22788, 117291)
- **The behavior of importing groups has changed slightly:** When a user selects an LDAP group to import into Coverity Connect, by default, it imports the top level group and all nested LDAP group members with the specified Group Filter. The user may set the scope of the Group Filter to either *Top level groups only* or *Top level groups and nested groups*. (IM-21347, 107963)

87.2.1.1.2. Bug fixes

The following bugs are fixed for Coverity Connect:

RG-954, 120773

Updated the wording in the CERT Report Compliance Scorecard to list the first rule violation (of the selected Target Compliance Level) so that it appears in the Standard: CERT C 2016 table.

RG-947, RG-918, 120246, 118599

A few MISRA C:2012 rules were categorized under `Required`. They are now correctly categorized under `Advisory`.

RG-917, 118534

Defects classified as `Intentional` or `False Positive` were incorrectly classified in the Security Report. This has been fixed.

RG-906, RG-781, 117449, 109450

Updated the MISRA Report Generator's JSON files to ensure that the PDF version of the report displays the same MISRA rules being shown in the UI.

RG-778, 108813

The **Target Integrity Level** in the Coverity Integrity Report did not display correctly in the Japanese locale. This has been fixed.

RG-774, 108243

Added an authentication key to the Security Report Generator. Users must now re-enter their authentication key before generating the report. This update minimizes security vulnerabilities.

RG-771, 108037

Updated the MISRA Report Help documentation to indicate that MISRA C++2008 Rule 8-3-1 is now supported.

RG-764, 106262

Improved the performance of report generators by limiting the number of defect instances that are returned when calling WS methods.

RG-509, 82942

Fixed an issue where projects were not visible if they were added to Coverity Connect after the report generator had started.

INS-2348, 121247

Resolved an issue in the Coverity Connect installer where the wrong description for "In Place upgrade mode" was displayed in console mode.

IM-22704, 121627

Fixed an issue where upgrading Coverity from version 8.7.1 to 2018.06 would result in a `Database Integrity Check` failure. This no longer happens.

IM-22699, 121603

Updated the `cov-admin-db check-integrity` feature to ensure that important database constraints are present even when unspecified. This reduces the risk of any performance issues.

IM-22691, 121552

Users upgrading from Coverity 2018.01 to 2018.06 no longer receive a column "defect_instance_details_id" contains null values error message.

IM-22673, 121364

Fixed an issue where updating or deleting streams would sometimes produce foreign key constraint violations.

IM-22265, 117150

The TLSv1 protocol is now disabled by default because of its security vulnerabilities. The only security protocol enabled by default is TLSv1.2.

Note that some previous versions of Coverity Connect clients (such as `cov-wizard` and `cov-manage-im`) do not support TLSv1.2 and might not work with a new server unless TLSv1 is enabled. To enable TLSv1 in the Coverity Connect server, read the "Enable TLSv1" instructions provided in the `${SERVER_INSTALL_DIR}/server/base/conf/server.xml` file.

IM-22629, 120983

The Coverity Connect server would sometimes disable users that were previously imported via LDAP if they were unable to connect to a directory service. This has been fixed.

IM-22593, 120574

Improved the synchronization of projects between Coordinator and Subscriber nodes in the Coverity Connect cluster. As a result, the risk of synchronization failures, requiring manual intervention to be resolved, has been reduced.

IM-22271, 117214

Improved the logging in the code path that creates analysis summaries. Analysis summaries that are not found for a specified hash are now properly handled.

IM-22267, 117185

We can now successfully import nested group members of groups with more than 1500 members.

IM-22572, 120258

Added a mechanism that prevents Coverity Connect backup jobs from running concurrently. With the fix, the backup jobs now run sequentially.

IM-22257, 117096

Improved the Coverity Connect server startup process, so that the server no longer switches to LDAP authentication despite Kerberos authentication being configured and enabled.

IM-22487, 119313

Improved the synchronization of triage stores between Coordinator and Subscriber nodes in the Coverity Connect cluster. As a result, the risk of synchronization failures, requiring manual intervention to be resolved, has been reduced.

IM-22458, 119008

Updated the `getUser` and `getUsers` API requests so that responses now include a timestamp of when the user last logged in.

IM-22229, 116954

Fixed an issue where the incorrect database size was being displayed when users ran System Diagnostics from the Help page. The correct database size is now calculated and displayed.

IM-22220, 116870

Fixed an importing issue in Coverity Connect that resulted from component map changes made prior to importing a new file.

IM-21892, 113949

Fixed a data overflow issue where `cov-commit-defects` would hang when processing large intermediate directories. The server memory footprint has been reduced and `cov-commit-defects` no longer produces a data processing error.

87.2.1.1.3. Known issues and solutions

Coverity Connect has the following known issues:

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.

INS-2307, 118662

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-2133, 112939

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.

IM-19690, 89946

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-19685, 89897

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

IM-19048, 84453

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

IM-18710, 82648

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-18707, 82643

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.

CPU-38, 82579

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.

CPU-17, 80045

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.

IM-17701, 75559

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

IM-17660, 75263

An error occurs when a custom role is created using a multi-word rolename that is the same as a built-in rolename, even if there are case differences between the two rolenames.

INS-1477, 73401

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 (1.8), or by specifying a max heap setting for `cov-im-daemon`.

IM-16076, 67748

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.

INS-1274, 63454

Although the upgrade doc 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, fall back to backup-and-restore upgrade.

87.2.2. Coverity Policy Manager 2018.09

Coverity Policy Manager is a component of the Coverity Platform installation package.

There are no new or changed features, and no bug fixes or known issues for this release.

87.3. Coverity Analysis 2018.09

This section provides updates about Coverity Analysis components.

87.3.1. Important Coverity Analysis information

There have been several deprecations and EOLs this release:

- The `--chcmdline-type` option to the `cov-build` and `cov-capture` command has been deprecated. (CAP-1343, CAP-609, 47799)
- Dropped support for Java 9. (COVP-2051)

87.3.1.1. New and changed features

Coverity Analysis has the following new and changed features:

- **Added support for Java 10.** (SAT-26217, 120926)

- **The new `--tmpdir <tmp>` and `-t <tmp>` options to the `cov-manage-emit` command specify the temporary directory to use.** On UNIX, the default is `$TMPDIR`, or `/tmp` if that variable does not exist. On Windows, the default is to use the temporary directory specified by the operating system. (SAT-19969, 93555)

87.3.1.2. Bug fixes

SAT-26673, SAT-26251, 121055

Updated the models for Java classes in the `java.util.zip` namespace, allowing the analysis to report "Zip Slip" vulnerabilities.

SATW-1404, 109859

Updated issue types to compliance rule-based main event tags and added more descriptive Coverity Connect issue names.

SAT-26338, 121468

Fixed an assertion failure in the web application security analysis, involving certain C# methods with `ref` parameters.

SAT-26644

Updated models for the Java class `java.util.Scanner`, allowing the analysis to report more vulnerabilities.

CAP-1302, CAP-1289, CAP-899, 118355, 116523, 71628

Different failures in Poky, Yocto, and BitBake builds would sometimes occur when run under `cov-build`. These failures would produce the following error message: "ERROR: ld.so: object '/path/to/cov-analysis-kit/bin/libcapture-linux64-\${PLATFORM}.so' from LD_PRELOAD cannot be preloaded". These issues have been fixed.

CAP-758, 60268

On 64-bit Linux systems, C++ builds run in Eclipse would sometimes produce the following error message: "ERROR: ld.so: object '/path/to/cov-analysis-linux64-version/bin/libcapture-linux64-.so' from LD_PRELOAD cannot be preloaded: ignored". This has been fixed.

87.3.1.3. Known issues and solutions

SAT-26530

On 64-bit Windows platforms, the length of the command string that can be passed to the Fortran syntax analyzer is limited (internally) to 32768 characters. If this limit is exceeded, `cov-run-fortran` fails and reports an "Argument list too long" error.

INS-1792, 89937

The Coverity Analysis installer fails when the installer path contains Japanese characters.

INS-1694, 84234: Installing into an existing folder

Coverity Analysis cannot be installed into an existing empty folder. Please select a non-existing folder.

87.3.2. Coverity Analysis checkers and user directives in 2018.09

The following sections describe new and updated features, bug fixes, and known issues for Coverity checkers and associated elements.

87.3.2.1. New and updated checkers and directives

The following table documents added language support for existing checkers.

Languages	Checkers	Checkers
Visual Basic	ASPNET_MVC_VERSION_HEADER	OS_CMD_INJECTION

New and changed checkers

ASPNET_MVC_VERSION_HEADER

The ASPNET_MVC_VERSION_HEADER checker now supports Visual Basic. (SAT-26408, 121741)

OS_CMD_INJECTION

The OS_CMD_INJECTION checker now supports Visual Basic. (SAT-26389, 121676)

87.3.2.2. Bug Fixes

SATW-2369, 120500

Fixed a stack overflow issue with MISRA C-2012 Rule 13.1.

SAT-26489

For C and Objective-C, Coverity Analysis now treats an access of the first field of a struct like an access of the struct itself. As a result, we've fixed RESOURCE_LEAK false positives.

87.3.2.3. Known issues and solutions

SAT-26651

Clear Linux is not supported in the 2018.09 release.

SAT-17490, 84256: INTEGER_OVERFLOW churn

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

SAT-7224, 43971: XSS

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

87.3.3. Compiler configuration, Build capture, and Compiler Integration Toolkit (CIT) 2018.09

This section lists new features, bug fixes, and known issues related to Coverity-supported compilers (including configuration), and the Compiler Integration Toolkit (CIT).

87.3.3.1. Important Compiler Integration Toolkit (CIT) information

There have been no Compiler Integration Toolkit (CIT) deprecations or EOLs this release.

87.3.3.2. New and changed features

Compilers and the Compiler Integration Toolkit (CIT) for Coverity Analysis has the following new features:

- **Added Java 10 support for Coverity Analysis.** (CMPG-2809, 120696)
- **Added support for the Nintendo Switch SDK compiler.** (CMPCPP-6923, 112469)

87.3.3.3. Bug fixes

The following bugs are fixed for compilers and the Compiler Integration Toolkit (CIT) for Coverity Analysis analysis in 2018.09:

CMPCPP-7564, 121324

Fixed a regression (introduced in LLVM/Clang 6) which would result in a spurious "-Werror=nsconsumed-mismatch is currently enabled, but was not in the PCH file" error.

CMPOCCPP-190, 117204

The use of the Objective-C keyword `@available` (or `__builtin_available` for C code) no longer causes `cov-internal-emit-clang` to crash.

CMPCPP-7601, 55660

C/C++ compilers that are configured with the `--clang` option now support Coverity annotations.

87.3.3.4. Known issues and solutions

CAP-1176, 97630

`cov-build --instrument` has a known issue when running the `xdcmake.exe` tool of Visual Studio 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 either:

- Modify the build such that `xdcmake.exe` is run from a 64-bit process.
- Ignore the `xdcmake.exe` process by adding `--capture-ignore xdcmake.exe` to your `cov-build` invocation.

CMPJS-286, 95651

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 used with some frameworks.

CMPJ-368, 65669, 65721

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 `LANG` or `LC_CTYPE` environment variables.

87.3.4. Coverity Analysis 2018.09 Commands

87.3.4.1. Commands related to the build process

This section lists new features, bug fixes, and known issues for `cov-build` and related commands, including `emit` and `translate` commands.

87.3.4.1.1. New and changed features

There are no new and changed features for commands related to the build process (including `emit` and `translate` commands) in 2018.09.

87.3.4.1.2. Bug fixes

There are no bug fixes for build-related commands (including `emit` and `translate` commands) in 2018.09.

87.3.4.1.3. Known issues and solutions

Build-related commands have the following known issues and solutions:

CMPCPP-4764, 72964

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.

CAP-812, 64428

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 hang occurs, 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 either:

- Uninstall KB2919355

OR

- Add the `--instrument` flag to your `cov-build` invocation:

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

SAT-12174, 62745

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.

CAP-332, 26881, 38175

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

Error message:

```
[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
```

87.3.4.2. Commands related to analysis

This section lists new features, bug fixes, and known issues for `cov-analyze` and related commands.

87.3.4.2.1. Bug Fixes

There are no bug fixes for analysis-related commands in 2018.09.

87.3.4.2.2. Known issues and solutions in 2018.09

Analysis-related commands have the following known issues and solutions:

CMPG-1741, 70845, 71216

The `cov-run-desktop` command sometimes fails on large Java compilations, potentially causing emit database corruption on Windows platforms. This can manifest as a `cov-analyze` crash. More commonly, `cov-emit-java` itself will fail with access violation crashes or errors concerning a failure to acquire a lock. These will appear in `cov-run-desktop-log.txt`. If this issue occurs, you can work around it by specifying `-j 1` with `cov-run-desktop`.

87.3.4.3. Commands related to Test Advisor

This section lists new features for Test Advisor.

87.3.4.3.1. Bug fixes in 2018.09

The following bugs are fixed for Test Advisor-related commands:

87.3.5. Coverity Wizard 2018.09

This section lists new features, bug fixes, and known issues related to Coverity Wizard.

87.3.5.1. Important Coverity Wizard Information

There has been one support removal this release:

- In order to improve security, we no longer bundle GTK+ with cov-wizard. As a result, the appearance of cov-wizard might vary depending on the version and GTK+ theme. We recommend that you use the Ambiance theme, if you are experiencing UI issues. (PRD-10571, 119795, PRD-4854, 49756)

87.3.5.2. New and changed features

Coverity Wizard has no new and changed features in 2018.09.

87.3.5.3. Bug fixes

Coverity Wizard has no bug fixes in 2018.09.

87.3.5.4. Known issues and solutions

Coverity Wizard has the following known issues in version 2018.09:

PRD-9245, 90621

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

PRD-9208, 90489

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-8453, 83450

When using a self-signed certificate, if the user chooses not to trust a certificate, they might be prompted multiple times in a row (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-8227, 82196

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-7595, 77742

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.

PRD-6832, 70361

The guided policy creation wizard "Documentation" link fails to open properly on Linux. Open the *Coverity Wizard 2020.12 User Guide* separately to view this documentation.

PRD-6760, 69815

The *Guided Test Advisor Policy Creation Wizard* uses Java regex validation instead of the Perl regex validation that Coverity Analysis Test Advisor users. 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-5770, 59676

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-5387, 54143

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

PRD-5290, 53367

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.

87.3.6. Test Advisor 2018.09

Coverity Test Advisor is a component of the Coverity Analysis installation package.

87.3.6.1. New and changed features

Test Advisor has no new and changed features in 2018.09.

87.3.6.2. Bug fixes

There are no fixed bugs for Test Advisor in 2018.09.

87.3.6.3. Known issues and solutions

Test Advisor has the following known issues and solutions in 2018.09:

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.

TADE-1733, 90584

Using a Bullseye small runtime newer than version 8.9.26 with Test Advisor Developer Edition might not work correctly.

TADE-1686, 88602

Function Coverage Instrumentation network coverage collection does not support IPv6, and might fail to collect coverage data if the emit server address contains a hostname which resolves to an IPv6 address. This problem may be avoided by explicitly specifying an IPv4 address, or by using a hostname which only resolves to an IPv4 address.

TADE-1567, 83788

If you encounter the following errors when running your Java build/tests under `cov-build/cov-capture` for Test Advisor, then your native build is likely already using JaCoCo as part of the build:

```
Caused by: java.lang.RuntimeException: Class java/util/UUID could not be
instrumented.
...
```

```
Caused by: java.lang.NoSuchFieldException: $jacocoAccess
...
```

In order to run this with Test Advisor you will need to disable the native JaCoCo in your build. This will depend on the build system used, but for common build systems, like Maven, this can be as simple as adding "-Djacoco.skip=true".

TADE-1037, 69957

When using Desktop Analysis with Accurev, if your setup requires access to different servers, you will need to configure this through an Accurev `wspaces` files. For convenience, here is a link to the Accurev documentation: <http://www.borland.com/Products/Change-Management/AccuRev> .

TADE-631, 60051

`cov-commit-defects` might fail when:

- it is committing the results of a Test Advisor analysis, and
- it is run on a Windows machine with a non-utf8 Japanese locale, such as `shift-jis`

TADE-635, 60094

Coverage for Java tests can sometimes erroneously flag the closing brace of a finally block as coverable and uncovered. This is caused by the way the Java compiler generates the byte-code for the finally block, and is a limitation of the underlying coverage tool.

This issue can be resolved either by using a Test Advisor policy and code comments to "ta-ignore" the area of the code where the coverage is incorrect, or by writing a test that causes a `RuntimeException` to be thrown from inside the try block.

PRD-5678, 58996

In Coverity Wizard, on the Download Test Metrics screen in the Test Prioritization workflow, the *Test metrics download summary* status indicator might incorrectly display the status "Test metrics files are present" when the test metrics files have actually been deleted.

Test metrics files can be deleted in the following ways:

- Perform a new build with the *Clear intermediate directory before each build* check box activated.
- Delete the intermediate directory.
- Directly delete individual test metrics files within the intermediate directory.

PRD-5482, 56425

For Linux users creating new policy files with the Coverity Wizard Test Advisor Policy Editor, the File → Save and File → Save As actions will not be enabled until the new document has been modified. Evaluating the policy, or returning to Coverity Wizard will prompt users to save the new policy file.

CMPCPP-3461, 55514

The source code locations of template functions of template classes are incorrect. This might lead to incorrect results from Test Advisor, as coverage might be misattributed to such functions, or missing from such functions, due to the incorrect locations.

87.3.7. Dynamic Analysis 2018.09

Dynamic Analysis is a component of the Coverity Analysis installation package.

87.3.7.1. Known issues and solutions

Dynamic Analysis has the following known issues and solutions in 2018.09:

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.

JDA-681, 20788

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

JDA-694, 21417

Specifying certain combinations of the `instrument-arrays`, `instrument-collections`, `detect-races`, and `detect-deadlocks` options to the Dynamic Analysis agent have 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, then Dynamic Analysis reports races on neither collections nor arrays.

JDA-720, 22148

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

87.3.8. Architecture Analysis 2018.09

Coverity Architecture Analysis is a component of the Coverity Analysis installation package.

87.3.8.1. Important Architecture Analysis information

Architecture Analysis has no changed features and support in 2018.09.

87.3.9. Extend SDK 2018.09

Coverity Extend Software Development Kit is a component of the Coverity Analysis installation package.

87.3.9.1. Known issues and solutions

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.

87.4. Coverity Desktop 2018.09

The Coverity Desktop plug-in is available for various platforms from the Coverity Connect *Downloads* menu.

87.4.1. Important information

There has been one EOL this release:

- Dropped support for Java 1.6 and 1.7. Note that Java 6 (1.6) and Java 7 (1.7) code may still be analyzed using the Eclipse, IntelliJ, and Android Studio plugins. (PRD-10576, 120315)

87.4.2. Coverity Desktop for Eclipse in 2018.09

87.4.2.1. New and changed features

Coverity Desktop for Eclipse has the following new and changed features in 2018.09:

- **Added support for Eclipse v4.8.** (PRD-104447, BZ 113633)

87.4.2.2. Bug fixes

There are no bugs for Coverity Desktop for Eclipse in 2018.09.

87.4.2.3. Known issues and solutions

Coverity Desktop for Eclipse has no known issues in version 2018.09.

87.4.3. Coverity Desktop for Microsoft Visual Studio in 2018.09

87.4.3.1. New and changed features

Coverity Desktop for Microsoft Visual Studio has no new and changed features in 2018.09.

87.4.3.2. Bug fixes

There are no bugs for Coverity Desktop for Microsoft Visual Studio in 2018.09.

87.4.3.3. Known issues and solutions

Coverity Desktop for Visual Studio has no known issues in version 2018.09.

87.4.4. Coverity Desktop for IntelliJ IDEA in 2018.09

87.4.4.1. New and changed features

Coverity Desktop for IntelliJ IDEA has the following new and changed feature(s) in 2018.09:

- Added CLion version support (2017.3 to 2018.1) for IntelliJ plugins. (PRD-10514, BZ 118222)

87.4.4.2. Bug fixes

Coverity Desktop for IntelliJ has no bug fixes in 2018.09.

87.4.4.3. Known issues and solutions

Coverity Desktop for IntelliJ IDEA has the following known issues in version 2018.09:

PRD-10553, 119444

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-10076, 105052

When using whole program checkers in IntelliJ, a warning about missing class files might be seen 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-8397, 83106

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

PRD-8042, 80698

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.

PRD-8038, 80693

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

PRD-7991, 80599

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-7980, 80573

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

PRD-7453, 76907

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.

87.5. Coverity Documentation 2018.09

The following new documents and changes were made in 2018.09:

SAT-26702

Coverity CodeXM documentation has been corrected and expanded. The CodeXM QuickStart Tutorial introduces some features that are new to Coverity 2018.09.

SAT-22378, 106737

The `cov-import-results` entry has been updated in the Command Reference. It replaces the individual language-specific options. The value for the `--lang` option `<lang>` is one of `cpp`, `cs`, `java`, `javascript`, `objc`, `php`, `python2`, `python3`, `ruby`, `scala`, `swift`, `text-files`, or `vb`. This option sets the source language and analysis domain in the output `error.xml` file and replaces language-specific options like `--cpp` and `--java`. For more information, see the Command Reference.

SAT-25769, 119434

The Coverity Checker Reference now includes a table that shows Coverity checker coverage for the OWASP 2016 Mobile Top Ten.

SAT-24898, 116594

The Coverity CodeXM Common Library Reference, a new document in HTML form, has also been added to document some new functions that are available to all supported languages.

SAT-21823, 104440

The `--occurrences` option, which filters output based on the occurrence count for issues, is now documented in the "Output and filtering options" section for the `cov-run-desktop` command.

SAT-25718, 119283

Added a new section about upgrade recommendations and MISRA changes in the Upgrade Guide. The new MISRA engine might affect the number and quality of the defect reports.

PRD-10371, 110950

Version-specific IDE and Java requirements for Coverity Desktop have been centralized in "IDE and Java Version Support" (Section 7.5.2) of the Coverity Desktop 2018.09 Installation and Deployment Guide. These version-specific requirements are no longer duplicated in other books and sections of the documentation set.

Appendix A. Legal Notice

Table of Contents

A.1. Legal Notice	306
-------------------------	-----

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

Legal Notice

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

Legal Notice

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:

Legal Notice

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

Legal Notice

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.