



# Klocwork 2024.4 Release Notes

**Document Number:** KW2024\_4 \_009

**Certified version:** Klocwork 2024.4

Author	Revision	Comments	Date
MTooke	0.1	Initial draft created	May-30-13
MTooke	0.2	Added table describing MISRA archives	July-02-13
AWeekes	0.3	Updated for 2015 re-certification	May-29-15
SBommaganti	1.0	Updated for Klocwork 2016	Mar-3-16
SBommaganti	1.1	Updated fixed issues to include content from latest Klocwork 2016 build (11.0.1)	Mar-22-16
SBommaganti	1.2	Updated for Klocwork 2016.1	June-25-16
SBommaganti	1.3	Updated for Klocwork 2016.3	Nov-02-16

SBommaganti	1.4	Added Fixed Issues for Klocwork 2016.3	Jan-18-17
SBommaganti	2.0	Updated for Klocwork 2017	Mar-2-17
SBommaganti	2.1	Updated for Klocwork 2017.1	July-5-17
SBommaganti	2.2	Fixed a typo – reference to Klocwork 2017.2 in Klocwork 2017.1 document	Sep-7-17
SBommaganti	2.3	Updated for Klocwork 2017.2	Oct-16-17
SBommaganti	2.4	Updated for Klocwork 2017.3	Nov-2-17
MTofinetti	3.0	Updated for Klocwork 2018	2018-05-08
MTofinetti	3.1	Updated for Klocwork 2018.1	2018-07-05
MTofinetti	3.2	Updated for Klocwork 2018.2	2018-09-27
MTofinetti	3.3	Updated for Klocwork 2018.3	2018-12-06
MTofinetti	4.0	Updated for Klocwork 2019	2019-03-22
MTofinetti	4.1	Updated for Klocwork 2019.1	2019-07-04
MTofinetti	4.2	Updated for Klocwork 2019.2	2019-07-30
MTofinetti	4.3	Updated for Klocwork 2019.3; IEC 62304 added	2019-12-12
LRobertson	5.0	Updated for Klocwork 2020.1	2020-03-12
ABedford	5.1	Updated for Klocwork 2020.2	2020-06-29
<b>ABedford</b>	5.2	Updated for Klocwork 2020.3	2020-09-14
<b>ABedford</b>	5.3	Updated for Klocwork 2020.4	2021-02-24
<b>ADunster</b>	6.0	Updated for Klocwork 2021.1	2021-04-26
<b>ADunster</b>	6.1	Updated for Klocwork 2021.2	2021-08-17

<b>ADunster</b>	6.2	Updated for Klocwork 2021.3	2021-11-30
<b>ADunster</b>	6.3	Updated for Klocwork 2021.4	2022-01-18
<b>JBritton</b>	6.4	Updated for Klocwork 2022.2 and Klocwork 2022.1 plus rebranding	2022-06-30
<b>JBritton</b>	6.5	Updated for Klocwork 2022.3 and Klocwork 2022.4	2022-12-17
<b>JBritton</b>	6.6	Updated for Klocwork 2022.4.SR1	2023-04-25
<b>JBritton</b>	6.7	Updated for Klocwork 2023.2	2023-07-26
<b>JBritton</b>	6.8	Updated for Klocwork 2023.4	2023-12-29
<b>JBritton</b>	6.9	Updated for 2024.2 and add EN 50716	2024-07-24
<b>JBritton</b>	7.0	Updated for Klocwork 2024.4	2024-12-29

## Contents

Release Notes Klocwork 2024.4 .....	6
What's new in Klocwork 2024.4.....	6
Fixed issues in Klocwork 2024.4.....	19
Limitations .....	24
Release Notes Klocwork 2024.3 .....	41
What's new in Klocwork 2024.3.....	41
Fixed issues in Klocwork 2024.3.....	56

## Referenced Standards

Standards referenced in this document refer to the following versions:

Standard	Version
ISO 26262	ISO 26262:2018
IEC 61508	IEC 61508:2010
IEC 62304	IEC 62304:2006/AMD1:2015
EN 50716	EN50716:2023

## Trademarks

"MISRA", "MISRA C" and "MISRA C++" are registered trademarks of The MISRA Consortium Limited.  
Windows is a registered trademark of Microsoft Corporation.

## Related Documents

Document ID	Title
KW2024_4_001	Functional Safety Manual for Klocwork
KW2024_4_003	Klocwork ISO 26262 / IEC61508 / EN50716 Certified Checkers
KW2024_4_005	Klocwork Checker Qualification Pack



## Release Notes Klocwork 2024.4

These release notes cover Klocwork 2024.4 and include information about what's new in this release, issues we've fixed since the last release, and any limitations you should be aware of.

(Also available at <https://help.klocwork.com/current/en-us/concepts/releasenotes.htm>)

### Changes affecting migration

This section details product changes that affect how Klocwork data is migrated from a previous version. For general information about upgrading, see [Upgrading from a previous version](#).

### Licensing changes

2023 licenses are not compatible with Klocwork 2024.4. You need a new license to use the latest version of the product. Contact [license@perforce.com](mailto:license@perforce.com) to obtain a new license.

### Disabled checkers

If you chose to migrate your `projects_root` directory, verify that you have the same checker configuration as the previous release before your first integration build analysis.

### What's new in Klocwork 2024.4

Here are the highlights for Klocwork 2024.4. If you're upgrading, see the [Limitations](#) for items that might affect your upgrade and usage. (also available at <https://help.klocwork.com/current/en-us/concepts/whatsnewmain.htm>)

## Klocwork and Validate enhancements

This release includes the following enhancements

### **Back up projects and server information with minimal downtime**

To minimize downtime, you can safely back up information without ever having to stop your servers. See hot backup [Method 1: Use supported scripts](#) to learn how to back up individual projects or Validate server configurations and restore them later.

### **Back up projects and server information with minimal downtime**

To minimize downtime, you can safely back up information without ever having to stop your servers. See hot backup [Method 1: Use supported scripts](#) to learn how to back up individual projects or Validate server configurations and restore them later.

### **Improved workflow for application token authentication**

Authenticating a client in automated environments is now more streamlined and secure. This enhancement is especially useful for setups like Docker container deployments.

You can securely authenticate a client by storing the application token in a secret storage system such as Docker Secrets, then use your system's automated interaction methods to pass the token using the `kwauth -t` command.

For more information, see [Authentication using application tokens](#).

### **Added support for regular expressions when creating modules in Java**

Regular expressions are now supported along with Glob for pattern matching when you create modules in Java. A wide range of standard regex characters are supported for precise pattern matching. For more information and examples, see [Creating a module](#).

### **Updated taxonomy page for improved navigation**

To reduce visual clutter and simplify navigation, taxonomy categories are now collapsed by default. Categories will expand when you click on them or search the page. To expand or collapse all categories at once, use the top menu buttons.

### **Added confirmation step for editing multiple issues at once**

To help ensure that your bulk changes are intentional, a confirmation box now appears when you edit multiple issues at once from the search screen.

### **Improved CI build logs for better visibility and troubleshooting**

To provide a comprehensive view of the build process, CI build logs now contain both the analysis and the import build logs, making it easier to debug and track the progress of CI builds.

## **C and C++ enhancements**

In this release, we:

- Added full C and C++ support for the Bazel build system on Windows and Linux
- Replaced existing community MISRA checkers with Klocwork supported checkers

## **Java enhancements**

In this release, we:

- Added full Java support for the Bazel build system on both Windows and Linux
- Added support for differential analysis in kwciagent
- Upgraded Jetty to version 9.4.56

## **Plug-ins and extensions**

In this release, we:

- Upgraded the JetBrains IDEs (IDEA, CLion, and Android Studio) plug-ins to version 2024.2



- Added SAML/OIDC authentication to the VS Code plug-in, allowing you to authenticate securely with the Validate server and reducing the need for multiple passwords

### Expanded coverage for coding standards

This release includes new and expanded coverage for the following coding standards:

- CERT C and C++ (includes 100% coverage for the L1 rules)
- CERT Java
- MISRA C 2012 with Amendment 2 (includes 100% coverage)
- 

### Checker improvements

#### New checkers

The following checkers were added in this release:

Checker	Description
ABV.NON_ARRAY	This checker finds defects when any non-array object is used as an array.
CERT.EXCEPTION.OVER.BOUNDARY	This CERT checker provide support for CERT ERR59-CPP: Do not throw an exception across execution boundaries.
CERT.FIO.FGETS	This CERT checker provide support for CERT FIO37-C: Do not assume that fgets() or fgetws() returns a nonempty string when successful.
CERT.MEM.OBJ_LIFETIME_CTOR CERT.MEM.OBJ_LIFETIME_DTOR	These CERT checkers provide support for CERT MEM53-CPP: Explicitly construct and destruct objects when manually managing object lifetime.

Checker	Description
CERT.MEM.SMART_PTR.OWNED CERT.MEM.SMART_PTR.OWNED.THIS	These CERT checkers provide support for CERT MEM56-CPP: Do not store an already-owned pointer value in an unrelated smart pointer.
ITER.ADVANCE.NONADJACENT	This CERT checker provides support for CERT CTR55-CPP: Do not use an additive operator on an iterator if the result would overflow.
ITER.END.OUTPARAM.MIGHT ITER.END.OUTPARAM.MUST	These CERT checkers provide support for CERT CTR52-CPP: Guarantee that library functions do not overflow.
JAVA.ASSERT.ARG	This CERT checker provides support for CERT MET01-J: Never use assertions to validate method arguments.
JAVA.BIGDEC.FLOAT	This CERT checker provides support for CERT NUM10-J: Do not construct BigDecimal objects from floating-point literals.
JAVA.COMPARE.NAN	This CERT checker provides support for CERT NUM07-J: Do not attempt comparisons with NaN.
JAVA.CTOR.EXCEPT	This CERT checker provides support for CERT OBJ11-J: Be wary of letting constructors throw exceptions.
JAVA.DEBUG.ENTRY	This CERT checker provides support for CERT ENV06-J: Production code must not contain debugging entry points.
JAVA.FINAL.STATIC.VAR	This CERT checker provides support for CERT OBJ11-J: Be wary of letting constructors throw exceptions.
JAVA.INF.LOOP.EMPTY	This CERT checker provides support for CERT MSC01-J: Do not use an empty infinite loop.

Checker	Description
JAVA.LOOP.CTR.FLOAT	This CERT checker provides support for CERT NUM09-J: Do not use floating-point variables as loop counters.
JAVA.NATIVE.PUBLIC	This CERT checker provides support for CERT JNI00-J: Define wrappers around native methods.
JAVA.NESTED.EXPOSE	This CERT checker provides support for CERT OBJ08-J: Do not expose private members of an outer class from within a nested class.
JAVA.SERIALIZE.INNER	This CERT checker provides support for CERT SER05-J: Do not serialize instances of inner classes.
JAVA.THREADGROUP	This CERT checker provides support for CERT THI01-J: Do not invoke ThreadGroup methods.
JAVA.WAIT.IN.LOOP	This CERT checker provides support for CERT THI03-J: Always invoke wait() and await() methods inside a loop.
MISRA.MACRO.IDENT.DISTINCT.C90.2012 MISRA.MACRO.IDENT.DISTINCT.C99.2012	This MISRA checker provides support for MISRA C 2012 Rule 5.5: Identifiers shall be distinct from macro names.
MISRA.UNUSED_MACRO.2012	These MISRA checkers provide support for MISRA 2012 Rule 2.5 (Advisory): A project should not contain unused macro declarations.

## Modified checkers

Checker	Description
ABV.GENERAL	Finds fewer false positives and false negatives

Checker	Description
AUTOSAR.ADD.AUTO.SPECIFIER	Finds fewer false positives
CERT.OOP.CTOR.INIT_ORDER	Finds fewer false positives
MISRA.BITS.OPERAND	Finds fewer false positives
MISRA.CTOR.BASE	Finds fewer false positives
MLK.MUST	Finds fewer false positives
NPD.FUNC.MIGHT	Finds fewer false positives
RH.LEAK	Finds fewer false negatives
UNINIT.STACK.MUST	Finds fewer false positives

## Enabled or disabled checkers

No checkers were added to or removed from the default `enabled` field of the checker configuration files for this release.

## Taxonomy improvements

As part of our installation, you will find several custom taxonomy files that map Klocwork checkers to coding standards such as MISRA, CWE, OWASP, and DISA STIG.

Taxonomy	New/updated
autosar_cpp_18_10.tconf and autosar_cpp_18_10_ja.tconf autosar_cpp_18_10_strict.tconf and autosar_cpp_18_10_strict_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> <li>A5-2-5</li> </ul>

Taxonomy	New/updated
cert_c_all.tconf and cert_c_all_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> <li>▪ ARR30-C</li> <li>▪ FIO37-C</li> </ul>
cert_c_rules.tconf and cert_c_rules_ja.tconf	Substantial reorganization of the cert_c_rules.tconf and cert_c_rules_ja.tconf taxonomies.
cert_cpp_rules.tconf and cert_cpp_rules_ja.tconf	The taxonomies were renamed from cert_cpp.tconf and cert_cpp_ja.tconf to cert_cpp_rules.tconf and cert_cpp_rules_ja.tconf, respectively. Recommendations were removed so that the taxonomies contain only rules.
cert_java.tconf and cert_java_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> <li>▪ ENV06-J</li> <li>▪ JNI00-J</li> <li>▪ MET01-J</li> <li>▪ MSC01-J</li> <li>▪ NUM07-J</li> <li>▪ NUM09-J</li> <li>▪ OBJ11-J</li> <li>▪ SER05-J</li> <li>▪ THI01-J</li> <li>▪ THI03-J</li> </ul>
cwe_2019_top_25_cxx.tconf and cwe_2019_top_25_cxx_ja.tconf cwe_2020_top_25_cxx.tconf and cwe_2020_top_25_cxx_ja.tconf cwe_2021_top_25_cxx.tconf and cwe_2021_top_25_cxx_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> <li>▪ CWE-119</li> <li>▪ CWE-125</li> <li>▪ CWE-787</li> </ul>

Taxonomy	New/updated
cwe_2022_top_25_cxx.tconf and cwe_2022_top_25_cxx_ja.tconf cwe_2023_top_25_cxx.tconf and cwe_2023_top_25_cxx_ja.tconf	
cwe_all_cxx.tconf and cwe_all_cxx_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> <li>▪ CWE-119</li> <li>▪ CWE-120</li> <li>▪ CWE-122</li> <li>▪ CWE-125</li> <li>▪ CWE-787</li> <li>▪ CWE-788</li> <li>▪ CWE-805</li> <li>▪ CWE-806</li> </ul>
disa_stig_v4_cxx.tconf and disa_stig_v4_cxx_ja.tconf disa_stig_v5_cxx.tconf and disa_stig_v5_cxx_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> <li>▪ APSC-DV-002590</li> <li>▪ APSC-DV-003170</li> </ul>
Helix QAC taxonomies	Updated the Helix QAC taxonomies to Helix QAC version 2024.3.
helix_qac_c_cpp.tconf and helix_qac_c_cpp_ja.tconf	Added or modified checker mappings to the following categories: <ul style="list-style-type: none"> <li>▪ Portability - ISO C99 Language features</li> <li>▪ Portability - Language Extension</li> </ul>
hkmc_c.tconf and hkmc_c_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> <li>▪ C-ARR-001</li> <li>▪ C-ARR-008</li> <li>▪ C-ARR-009</li> </ul>

Taxonomy	New/updated
hkmc_cpp.tconf and hkmc_cpp_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> <li>▪ P-CTR-001</li> <li>▪ P-CTR-003</li> <li>▪ P-CTR-006</li> <li>▪ P-MEM-004</li> <li>▪ P-MEM-006</li> </ul>
iso_iec_ts_17961_c.tconf and iso_iec_ts_17961_c_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> <li>▪ 5.22</li> </ul>
kw_quality_std_cxx.tconf and kw_quality_std_cxx_ja.tconf	Added or modified checker mappings to the following categories: <ul style="list-style-type: none"> <li>▪ Buffer Overflow</li> </ul>
misra_c_2012_with_amd2_c11.tconf and misra_c_2012_with_amd2_c11_ja.tconf misra_c_2012_with_amd2_c90.tconf and misra_c_2012_with_amd2_c90_ja.tconf misra_c_2012_with_amd2_c99.tconf and misra_c_2012_with_amd2_c99_ja.tconf	Added or modified checker mappings to the following rules and directives: <ul style="list-style-type: none"> <li>▪ Dir 4.1</li> <li>▪ Rule 2.5</li> <li>▪ Rule 5.5</li> </ul>
misra_c_2023_c11.tconf and misra_c_2023_c11_ja.tconf misra_c_2023_c90.tconf and misra_c_2023_c90_ja.tconf misra_c_2023_c99.tconf and misra_c_2023_c99_ja.tconf	Added or modified checker mappings to the following rules and directives: <ul style="list-style-type: none"> <li>▪ Dir 4.1</li> <li>▪ Rule 1.3</li> <li>▪ Rule 2.5</li> <li>▪ Rule 5.5</li> </ul>
pci_3_2_1_cxx.tconf and pci_3_2_1_cxx_ja.tconf	Added or modified checker mappings to the following categories: <ul style="list-style-type: none"> <li>▪ 6.5.1</li> <li>▪ 6.5.2</li> </ul>

## Improvements to supported compilers

You'll find additional or improved support for the following compilers:

- Clang
- GNU

For the full list of supported C and C++ compilers, see [C/C++ compilers supported for build integration](#).

## Licensing

Klocwork supports Reprise License Manager (RLM).

2023 licenses are not compatible with Klocwork 2024.1 or newer. To use the latest version of the product, obtain a new license by contacting Perforce at [license@perforce.com](mailto:license@perforce.com).

For more information, see [Supported versions of RLM](#) and [Operating systems that support RLM dongles](#).

## Changes to system requirements

In this release, we added support for:

- Amazon Linux 2 (2.0.20241014.0 Update)
- Android Studio Jellyfish (up 2023.3.1 Patch 1), Ladybug (up to 2024.2.1 Patch 1)
- CLion 2024.2 (up to 2024.2.3)
- Debian 12.0 to 12.7
- Google Chrome 119.x to 131.x
- IntelliJ IDEA 2024.2 (up to 2024.2.4)



- Microsoft Edge 119.x to 131.x
- Mozilla Firefox 120.x to 132.x
- Ubuntu 24.04 to 24.04.1 LTS
- Visual Studio 2017 (up to v 15.9.68), 2019 (up to v 16.11.42), 2022 (up to 17.12.1)
- VS Code 1.85.2 to 1.95.3
- Windows 11 (v 24H2)

In this release, we ended support for:

- Debian 10.0 to 10.13
- Google Chrome 117.x to 118.x
- Microsoft Edge 117.x to 118.x
- Mozilla Firefox 118.x to 119.x
- SUSE Enterprise 12 SP4 to 12 SP5, 15 SP1 to SP4
- VS Code 1.85.1 to 1.84.2
- Windows Server 2012 to R2

For the complete list of supported versions, see [System Requirements](#).

### Discontinuation of NIS access control starting in Klocwork 2024.3

Starting in Klocwork 2024.3, NIS access control will no longer be supported. Some functionalities may be affected in Klocwork 2024.2.

When migrating from an earlier version to Klocwork 2024.2, you will need to switch to a different authentication method. It is recommended that you switch authentication methods before migrating, to ensure that you can continue to sign in after the upgrade. For migration information, see [Setting up NIS access control](#).

### End of life notice for CentOS Linux 7 starting in Klocwork 2024.3

Starting in Klocwork 2024.3, the following operating systems and installers are not supported:

- CentOS Linux 7

### Removal of the Jenkins plug-in starting in 2024.2

Starting in Klocwork 2024.2, the Jenkins plug-in has been removed from Klocwork and the installation package is no longer provided.

### Removal of Validate Code Review starting in 2024.2

Starting in Klocwork 2024.2, the Code Review function and its associated command line tools have been removed from Validate.

### Discontinuation of docs.roguewave.com in 2024

The docs.roguewave.com site was discontinued in early 2024. For Klocwork versions 2021 and earlier, see the offline documentation that is included with the product.

### Discontinuation of Klocwork Server installations in release 2023.4

Starting from release 2023.4, Klocwork Server installations have been discontinued. You can transition to a [Validate installation](#), which is designed to provide a more streamlined and integrated experience.

When transitioning from Klocwork to Validate:

- Stop your Klocwork instance and back up the **projects\_root** directory.
- During Validate install, set the **projects\_root** directory location to your current **projects\_root** directory.
- If you are currently using non-default values for ports or license server, be sure to set the same values when you install Validate.

## Fixed issues in Klocwork 2024.4

The following issues were fixed in Klocwork 2024.4.

(also available at <https://help.klocwork.com/current/en-us/concepts/fixedissues.htm>)

### General issues

Number	Description
00878983	Added full coverage for CERT C/C++ L1 rules.
01166818	CI Builds list in Validate now shows counts of all issues and open issues.
01105036, 01134711	Added support for differential analysis for Java projects.
01139050	Fixed a Validate UI bug that triggered a Java null exception when clicking on Issues for failed or in-progress builds.
01245055	Added the ability to remove orphaned issue records.
01162795	Fixed an issue that was causing uploads to Validate to fail during the CI build process.
01159962	Added validation of suppression configuration (sconf) file to ensure analysis will not start if the sconf file format is invalid.

Number	Description
01150587	Added the missing guidance for compliance report creation in Klocwork 24.1 documentation.
01172889	Fixed an issue with project descriptions in Validate not correctly showing multi-line formatting.
01202843	Removed recommendations from the CERT C++ taxonomy.
01163774	Fixed an intermittent communication issue between the Validate server and the license server.
01227506	Updated erroneous Japanese translation in licensing documentation.
01223834, 01196924	Fixed an issue where kwinject was not functioning correctly with the make utility in the S32DS IDE.
01216737	Fixed an issue causing slow synchronization of IAR projects and other compilers using kwinject on Windows.
01286567	Improved robustness of the clef process for C/C++ analysis
01288587	Upgraded third-party dependency to resolve an issue preventing the remote desktop agent from starting when running Klocwork Desktop in remote mode.
01246360	Fixed an issue where the kwciagent list --replace-path option did not function correctly when a file was specified as an argument.

Number	Description
01288218	Fixed a permissions issue that prevented a stream administrator from assigning roles to other users.
01289785, 01308468	Fixed an issue where clicking "View Log" in Validate caused an error for users without the project root admin role.
01290200	Updated the documentation with the latest known issues and limitations.
01297469	Fixed an issue where the Klocwork version was displayed incorrectly on the installed plugins screen for Visual Studio 2019 and earlier.
01306487	Added level attribute for rules to the cert_c_rules.tconf taxonomy.
01299928	Updated the documentation with a new example of the customized kwmysql.ini file.
01308076	Clarified documentation on kwauth usage with SAML and OIDC authentication.

## Checker issues

Number	Description
00636086	Reduced false positives for the checker MISRA.BITS.OPERAND.
00717550	Reduced false positives for the checker UNINIT.STACK.MUST.
00696219	Reduced false positives for the checker MLK.MUST.
01005013, 01045414, 01045914	Introduced new checker ABV.NON_ARRAY to detect out-of-bounds array access.
01045928	Reduced false negatives for the checker RH.LEAK.
01065986, 01242039	Reduced false positives and false negatives for the checker ABV.GENERAL.
01228010	Reduced false positives for the checker AUTOSAR.ADD.AUTO.SPECIFIER.
01228209	Reduced false positives for the checker CERT.OOP.CTOR.INIT_ORDER.

Number	Description
01228209	Reduced false positives for the checker MISRA.CTOR.BASE.
01235847	Reduced false positives for the checker NPD.FUNC.MIGHT.
01045913	Improved defect detection for the CXX.SUSPICIOUS.INDEX.CHECK* checkers.

## Limitations

This section contains limitations added in both this release and in previous releases.

(also available at <https://help.klocwork.com/current/en-us/concepts/kwlimitations.htm>)

### Limitations for installation, upgrade, and deployment

#### MySQL "Socket Error" appears during backup or restore

You may encounter a MySQL "socket error" while [running the backup and restore scripts](#) on Linux if the database host is not specified correctly in the command.

*Workaround:* To connect the script to the correct MySQL host, add the --db-host parameter to the command, followed by the fully qualified domain name (FQDN) or the IP address of the host where the database is running. For example:

```
backup --db-host 127.0.0.1
```

#### Validate server fails to start after command line installation on Windows

Starting in release 2024.2, the Validate server may fail to start if installed via command line on Windows, particularly when run from a network drive. The issue arises from a file-path lookup library that cannot resolve network paths for Java.

*Workaround:* We currently do not recommend running Klocwork or Validate tools from a network drive. If you do run the Validate server executables from a network drive, make sure to install Java 17 locally and either add it to the PATH or KW\_JAVA environment variable.



### **Non-expiring application tokens that were created in 2024.2 actually expire after 30 days**

A minor bug caused application tokens that were created in 2024.2 with no expiration date to actually expire after 30 days. If you attempt to authenticate using a non-expiring token after 30 days, the authentication will fail. This issue is fixed in Validate 2024.3.

*Workarounds:*

- Create an application token that has a custom expiration date (Perforce recommends two years).
- Upgrade to Validate 2024.3 or later (if available). Then, create a new application token and set the field Expires at to Never.

### **Issue with kwauthconfig prompting for credentials when using SAML or OIDC authentication**

If you have configured your Validate Server to use SAML or OIDC authentication, attempting to set or reset the database password using kwauthconfig will incorrectly prompt you for user name and password credentials when connecting to the server.

*Workaround:* Run kwauth --url <Validate server>:<port> to authenticate with the Validate Server. After successful authentication, restart kwauthconfig. It will not prompt you for credentials.

### **Cannot authenticate SAML or OIDC when using pre-2024.2 Klocwork tools and plug-ins**

When using Klocwork tools or plug-ins version 2024.1 or earlier on a SAML- or OIDC-enabled server, you must use [kwauth](#) tools from release 2024.2 or higher in order to authenticate.

Additionally, you are required to use the --insecure flag during authentication for your older tools to be able to authenticate with the server. This flag allows users to save their credential information in an insecure manner in their ltoken and should only be used when necessary.

### **kwconfigeditor displays an unused Severity 0**

This is expected and relates to the use of Severity 0 in QAC projects. Since no Klocwork projects use Severity 0, it can be ignored.

### **Mixed Klocwork and QAC projects will display severity levels based on the last uploaded project**

Mixed projects are not technically supported. It should be noted that if you attempt to upload both Klocwork and QAC results to the same project, the severities displayed in the Issues List in Validate will depend on which project type was uploaded last. If QAC results were uploaded last, then QAC severities will be used, and vice versa.

### **Unable to use kwgcheck after upgrading to Debian 10.7**

If you upgrade to Debian 10.7, you may be unable to launch kwgcheck for Klocwork Desktop Analysis due to missing libraries.

*Workaround:* To install the missing libraries, run the following command:

```
sudo apt-get install --reinstall libgtk2.0-0
```

### **dbvalidate cleanup utility can result in data loss**

Since the dbvalidate cleanup tool can result in data loss, it is not recommended to run dbvalidate or dbvalidate cleanup. When upgrading from a previous version, it is suggested that you migrate using the documented procedure. If an issue arises after the upgrade, please raise a support ticket..

### **Klocwork desktop tools cannot connect to previous servers**

The 2023.4 Klocwork desktop tools cannot connect to previous (23.3 and earlier) Klocwork servers, due to internal changes in the tools and the server APIs they communicate with.

### **Project build hangs on upload**

Network connection problems between the client and server, caused by infrastructure-related configurations, can cause kwadmin load to hang indefinitely.

*Workaround:* Use the hidden option --hard-timeout for kwadmin load to specify a timeout value in hours, minutes, or seconds. If a request from the client to the server goes unanswered for the duration of the timeout value, the load command will fail.

Example: `kwadmin load myProject myTables --hard-timeout 10m`.

### Limitations for Checker configuration migration

Note the following limitations with checker configuration files during the upgrade process (via the import process):

- Only modifications to default checker configuration files are imported. If you had a non-default checker enabled in an earlier installation and it was renamed in a new version, you will not see the checker in new builds. You must manually re-enable the checker in the new version of Klocwork.
- If a checker that was enabled by default was renamed in the new version of Klocwork, you will not see new codes until the first system build of the new installation.

### **kwcollect fails on tables generated by new analysis engine**

*Workaround:* If your project has been built with Klocwork 2018's new analysis engine, you must include the, '--all-sources' option on the command-line. This requirement does not apply if your project was built without Klocwork 2018's new analysis engine. To determine if your project was built with the new analysis engine, examine the output of the build process in the build.log, contained in the root of the build's output tables folder. Find the line that begins with 'Selected Engines'. Your project has been built using Klocwork 2018's new analysis engine if 'MODERN' appears between square brackets.

### **Debian 10.x and later cannot run the license server**

Debian no longer supports the Linux Standard Base core (lsb-core) as of version 10.x. Therefore, the license server cannot be run on the Debian 10.x and later platforms.

### **kwauth doesn't properly set HTTP/1.1 header**

Sometimes when the Klocwork Server IP address is associated with multiple host names or located behind a reverse proxy, kwauth does not properly resolve the FDQN of the Klocwork Server.

*Workaround:* To resolve this problem, we added a conditional host resolution based on a parameter in a specified configuration file. If you set it to 'false', then you can specify FQDN for the URL of the remote server. To set host resolution to 'false', you need to create a configuration file on the client side with the following address:

```
{client_tools_install_folder}\config\client_config.xml
```

The file must have the following structure:

```
<?xml version="1.0" encoding="UTF-8"?><params>    <host resolveHost="false" /></params>
```

### 64-bit REPCXX Memory Limitations

With 2020.4 SR1, 64-bit REPCXX (our classic C/C++ analysis engine) by default is capped at using 16 Gigabytes (GB) of memory. An attempt to exceed this limitation will result in the REPCXX process being terminated with exit code 121 and the following message:

```
"memory: REPCXX has exceeded the memory threshold of 16000000 Kb, mem_usage={0} Kb.  
Memory threshold can be increased by passing a value (in GB) to REPCXX through the  
'--memory-limit=' option or through the 'KW_CLASSIC_ENGINE_MEMORY_LIMIT' environment  
variable."
```

*Workaround:* As the error message indicates, the new --memory-limit REPCXX option or 'KW\_CLASSIC\_ENGINE\_MEMORY\_LIMIT' environment variable may be used to override the 16 GB memory limit. For example, to increase the memory limit to 32 GB per REPCXX process, you can run kwbuildproject with option --add-compiler-options '--memory-limit=32'.

As confirmation of the memory limit increase, the following message will be output to the build log when a REPCXX process is initiated:

```
"REPCXX memory input set to 32 GB (32000000 KB) "
```

### 'kwloaddb: Java heap space' error when trying to load a build

During a load or a build from a connected project, or when running kwcheck run, an 'OutOfMemory' exception may occur.

*Workaround:* To fix this error, you can try to increase the amount of memory for kwloaddb in your java\_wrappers\_memory.conf file. If you continue to see the error, contact [Customer Support](#).

### **dbvalidate outputs error(s) when attempting to validate a database that contains streams**

For projects with streams, when attempting to run dbvalidate as follows: `java -jar /<path-to-server>/class/dbvalidate.jar --projects-root /<path-to-projects-root>/projects_root --project a`

dbvalidate will incorrectly output error messages on steps a1, k, and occasionally j and l. These messages can be ignored.

### **Using 'sconf' files on Linux does not filter defects in the file specified.**

If you are using windows style paths in an 'sconf' file, connected projects can yield inconsistent results.

**Workaround:** Avoid using of windows style file paths (For example, 'C:\Checkout\PRs\PR-59822\test.cpp' or 'C:\\Checkout\\PRs\\PR-59822\\test.cpp'). You can use UNIX-style paths across all platforms, for example:

Windows: `C:/Checkout/PRs/PR-59822/test.cpp`

Linux: `/checkout/mydir/test.cpp`

**Note:** Use of the 'C:\Checkout\PRs\PR-59822\test.cpp' format should be avoided on **all platforms** as it is not supported.

### **Limitations for build integration**

#### **Cannot load Android 4.4 (KitKat) using the default memory settings for kwloaddb, kwadmin and kwjava**

When building the Android platform, you may need to increase the memory settings for certain Klocwork tools on the machine invoking the load process. These values can be modified in the `<klocwork_install>/config/java_wrappers_memory.conf` file.

## Limitations for C# analysis

The following features are not supported for C# integration projects:

Feature	Details
Build integration	<ul style="list-style-type: none"><li>▪ <a href="#">Build specification templates</a></li></ul>
<a href="#">Validate</a>	<ul style="list-style-type: none"><li>▪ "Show implementation", "Show declaration", and <a href="#">Source Cross-Reference</a></li></ul>

The following features are not supported for C# desktop analysis:

- Display of server issues in Visual Studio
- File-level analysis in [Visual Studio](#) (only solutions and projects can be analyzed)
- Using [knowledge bases](#)

## Limitation for Python

In release 2023.3, the following checkers listed in the Python taxonomy are not supported. They are disabled by default and can be enabled manually:

E0001	F0001	I0001	I0020
E0013	F0002	I0010	I0021
E0014	F0010	I0011	I0022

E0015	F0011	I0013	I0023
-------	-------	-------	-------

## Limitations for Validate

### Backups cannot be restored with a different Validate version

A backup created with one version of [validate backup](#) cannot be restored with a different version of [validate restore](#). For example, a backup created with Validate 24.4 cannot be restored using Validate 24.3 or 25.1.

*Workaround:* Use the same version of Validate to back up and restore a project or server. To avoid compatibility issues when you [migrate to a new server version](#), we recommend that you create new project and server backups after you finish the migration.

### Unable to sign into a Validate 2024.1 or earlier server

You are unable to sign in to a Validate 2024.1 or earlier server, or you encounter the error Trying to rerun async when not logged in.

*Workaround:* Clear your browser cache and cookies for the Validate server.

### CI build paths are not shown in the tree structure of the 'Modules' page in Validate/Validate

When creating a new module, the file tree always shows files from the latest Server build (even if the latest uploaded build was CI build from kwciagent).

*Workaround:* If you want to include files or paths from a CI build, you need to manually input them using 'Path patterns' instead of using the 'Use tree' field.

## Limitations for the Visual Studio extension

### **Authentication failure with Visual Studio Extension 2024.1 or earlier with Validate Server with modern authentication enabled**

An issue arises when using the Visual Studio extension version 2024.1 or earlier with a Validate server upgraded to use modern authentication (modern-jit or modern-pp). When prompted to log in and enter your user name and password, authentication will fail.

*Workaround:*

1. Install the 2024.2 authtools package or a later version.
2. Run Device Authorization with the new kwauth, adding the --insecure flag. Your user name might have changed, so you may need to contact your system administrator.

You can continue to use Visual Studio Extension 2024.1 or upgrade to the latest plugin. Note that the latest plugin defaults to a new analysis engine based on kwcheck.

### **The number of issues changes for each analysis**

For each Visual Studio extension analysis that you run, a different number of issues appear.

*Workaround:* To resolve this issue, use the default kwcheck engine.

### **'One or more extensions were loaded using deprecated APIs' warning message in Visual Studio 2019**

Visual Studio 2019 may give a warning message regarding deprecated APIs. If you select the recommended option to not allow deprecated API usage, this will disable the Klocwork plug-in and you will no longer be able to access the Klocwork tools in VS.



*Workaround:* Select the 'Don't show this message for current extensions' option to safely ignore this warning and continue to use the Klocwork plug-in.

### **'Visual Studio stopped responding for X seconds.' warning message in Visual Studio 2019**

Visual Studio 2019 may give a warning message regarding slower performance in relation to use of the Klocwork plug-in. If you select the option to 'disable this extension', it will disable the Klocwork plug-in and you will no longer be able to access the Klocwork tools in VS.

*Workaround:* Select the 'Don't show this message for current extensions' option to safely ignore this warning and continue to use the Klocwork plug-in.

### **Visual Studio 2015 preventing analysis of C++**

A known issue with Visual Studio 2015 running on Windows Server 2012 prevents the Klocwork extension for Visual Studio from analyzing C++ projects.

*Workaround:* Either modify the build configuration version to x86, or switch the analysis mode to Classic. To switch to Classic mode, go to **Tools > Extensions and Updates > Klocwork > Options**, and select the **Analysis** tab. Under **Analysis Mode**, select the Use Classic Mode check box.

### **Visual Studio extension may be disabled**

When installing the Klocwork extension for Visual Studio into Visual Studio 2015, the extension may initially be disabled. If so, go to **Tools > Extensions & Updates** and select **Installed**. Select the **Klocwork Extension for Visual Studio**. You may need to restart Visual Studio.

### **Visual Studio hang**

The Klocwork development team is tracking a support request with the Visual Studio Technical Support team where user actions cause Visual Studio to hang under a number of conditions. These Visual Studio hangs occur whether or not the Klocwork VS Extension is installed. For example, when navigating into the definition of a function that is defined in a

source file that is not currently open in a tab in Visual Studio, Visual Studio opens that file in a temporary tab. When this temporary tab is open, if you then navigate to the definition of a different function, Visual Studio hangs.

### **'kwcc' error in Visual Studio after upgrading**

If, before 2020.1, you deployed the MISRA checkers to your project using kwdeploy, and have a misra.xml file in your %USERPROFILE%\klocwork\plugins folder, you might see errors similar to the following:

```
kwcc: Error: C:\Users\username\.klocwork\plugins\misra.xml:5783:  
Trying to describe error 'MISRA.STDLIB.ILLEGAL_WRITE.2012_AMD1' several times.  
Repeated descriptions are ignored
```

*Workaround:* To fix this issue, delete the misra.xml file located in your %USERPROFILE%\klocwork\plugins folder before performing the upgrade.

### **Help for Klocwork community checkers cannot be accessed directly from Visual Studio**

If you attempt to access the help for a community checker by right-clicking the checker and selecting **View Checker Documentation**, you will get a 'Cannot find requested topic on your computer' error message.

*Workaround:* Offline help for the community checkers is available by using the portal. [Online help](#) is also available.

### **The filter by severity option in the Microsoft Visual Studio extension may not display custom severities for C++ projects**

For C++ projects where you have defined custom severities, the severity filter list may not display the correct items. The list may display default severity names, or in the case where you have a mixed C++ and C# project, the list will display the C# severities. You can still use the filter, but the severity names displayed in the issue tree may not match the items you selected in the list (as the filter is applied by severity number).

### **Klocwork server option fails to retrieve projects when you use a hard-coded IP address**

If you use a hard-coded IP address in the Klocwork server dialog under the Klocwork options menu, the Klocwork extension for Visual Studio fails to retrieve the list of projects.

*Workaround:* Use the host name instead of the IP address; if this is not an option, you can add an entry in the hosts file for the IP address.

### **Options dialog is unresponsive if the license host box contains an invalid character**

If you enter the details for your license host and type an invalid character (such as a ';' or '#'), the dialog box turns red and you can't change any other settings within the Options dialog box.

*Workaround:* To fix this issue, replace the content of the license host box with a valid character (any letter of the alphabet).

### **Defect column does not update correctly unless tabs are converted to spaces**

If you press **tab** to indent code, this will insert a tab character or a configurable number of spaces. When the Klocwork extension shows the column number where a defect occurs, it counts a tab as one space. If you do not configure Visual Studio to convert tabs to spaces on source files that have tabs and when you press the **tab** key, the correct column information for defects may be incorrect.

*Workaround:* To ensure that Klocwork counts the correct number of columns when it displays where a defect has occurred, go to **Tools > Options > Text Editor > All Languages > Tabs** and select **Insert spaces**.

### **Klocwork Visual Studio plugin installation does not detect Visual Studio 2015**

If you try to install the Klocwork Visual Studio plugin on Visual Studio 2015 and you have a higher version of Visual Studio (such as 2017 or 2019) already installed, the VSIX installer does not automatically detect the Visual Studio 2015 installation. The error "This extension is not installable on any currently installed products" appears.

*Workaround:* To install the Klocwork plugin on Visual Studio 2015, run the compatible VSIX installer from the command prompt:

<path-to-Visual-Studio-2015-compatible-VSIX-Installer> \VSIXInstaller.exe kw-vsplugin.win64.vsix

```
"C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE\VSIXInstaller.exe"  
C:\KW_Installer\23.1\kw-desktop-tools.23.1.0.61.win64\kw-vsplugin.win64.vsix
```

## Limitations for VS Code

### **No stream defects are shown by default for the 2024.1 versions of the Perforce Static Analysis extension and kwcheck**

If you are using the Perforce Static Analysis extension for VS Code 2024.1 and it is pointing to a 2024.1 version of kwcheck, no stream defects will be shown by default.

*Workaround:* From your project directory, run the following command:

```
kwcheck set show_streams_as_system=true
```

### **Do not add the build specification location to the build command. It will be automatically collected from the 'Build Specification Location' setting.**

'Build Specification Generation Command' should not contain the builds spec location (-o <location>) when using the 'Automatically Update Build Specification' option.

*Workaround:* Specify the location with the Build Specification Location setting.

### **Build Specification Generation Command cannot contain single quoted arguments.**

If you need quoted parameters, use double quotes (same as a command prompt/terminal).

*Workaround:* Ensure your quoted parameters use double quotations.

## Limitations for Klocwork Desktop

### Analysis is not supported with any of the following configurations:

- When a project with symbolic links is configured with an external build specification that does not have the attribute "no-resolve". If a project uses symbolic links, the user must configure the project using an external build specification, and the external build specification must be created with the "no-resolve" option passed to kwinject.
- When a project with symbolic links is configured to use the Eclipse CDT toolchain. The Eclipse plug-in does not allow the user to set a "no-resolve" option.
- When a project contains a symbolic link to a directory. The plug-in supports symbolic links to files only.

## Limitations for the IntelliJ IDEA plug-in

### Difficult to connect to a project or stream with a long name

If you want to connect to a project or stream that has a long path or name, you may not see the full path of the project or stream because the Klocwork Project dropdown list might be too narrow to display the entire path and name.

*Workaround:* You can fix this problem by doing the following:

1. Open your IDE with Klocwork installed and then open a project. Doing so ensures the .idea folder gets generated in your project.
2. Close your IDE.
3. In your project's .idea folder, edit the 'misc.xml' file. Edit the following line by adding the project/stream name in the value parameter (The name appears with slashes included if it's a stream):  
`<option name="kwProject" value="project-name/stream-name" />`

4. Open your IDE. If you open your Klocwork settings, you can verify that you are connected to the correct project or stream.

## Limitations for the CLion plug-in

### **"Requires plugin 'com.intellij.modules.cidr.lang' to be installed" warning appears after restarting CLion**

After you restart version 2024.1 or 2024.2 of the CLion IDE, the warning "Requires plugin 'com.intellij.modules.cidr.lang' to be installed" appears.

#### *Workaround:*

By default, the Klocwork CLion plug-in is installed at the following location:

- Windows: %appdata%\Roaming\JetBrains\CLion2024.x\plugins\klocworkclionplugin
  - Linux: ~/.local/share/JetBrains/CLion2024.x/plugins/klocworkclionplugin
1. In klocworkclionplugin\lib, open the klocworkclionplugin-<kw version>.jar archive (for example, using 7-Zip).
  2. In 7-Zip, open the META-INF\plugin.xml file in a text editor.
  3. To update the META-INF\plugin.xml file, replace the line <depends>com.intellij.modules.cidr.lang</depends> with <depends optional="true">com.intellij.modules.cidr.lang</depends> and save the file.
  4. Close 7-Zip.
  5. Restart the CLion IDE. The warning will not appear.

## **Clion plug-in crashes for connected projects using older versions of the Klocwork desktop tools**

The 2023.4 version of the Klocwork Clion plug-in is only compatible with the latest version (2023.4) of the desktop tools.

## **Limitations for Klocwork extensibility**

### **Delay in build monitoring of projects when using kwinject (Windows)**

Synchronization of projects on Windows using kwinject are noticeably slower than running a native build.

*Workarounds:*

- For Klocwork 2024.3 and earlier: Before running kwinject, run the following command to set an environment variable:  
\_NO\_DEBUG\_HEAP=1
- For Klocwork 2024.4 and later: By default, this optimization on Windows is enabled. If there is a problem when using kwinject on Windows, you can set the environment variable:  
KWTRACE\_DISABLE\_FAST\_PIPE=1

### **C/C++ Path checker compilation makefile compatibility**

The makefile generated by kwcreatechecker on Unix systems requires GNU make to build the checker. On Windows, the makefile generated by kwcreatechecker requires nmake to build the checker.

*Workaround:* None.

### **Custom checker help is not found when searching in offline help**

As of 2022.1, it is not possible to search for custom checker help from the offline documentation accessed from the Klocwork Portal.

*Workaround:* You can search for custom checker help from the Configuration tab in the Klocwork Portal instead. From your project, click **Configuration**, select any taxonomy, and use the search box above the checker list. The help for any custom checker appears in the right pane.

### **Custom checker help cannot be displayed in some IDEs**

Custom checker help does not display for Eclipse, Klocwork Desktop, IntelliJ IDEA, and Android Studio.

*Workaround:* You can view custom checker help by accessing it through the Configuration tab in the portal. From your project, click **Configuration**, select any taxonomy, and use the search box above the checker list. The help for any custom checker appears in the right pane.



## Release Notes Klocwork 2024.3

These release notes cover Klocwork 2024.3 and include information about what's new in this release, and issues we've fixed since the last release,

### Changes affecting migration

This section details product changes that affect how Klocwork data is migrated from a previous version. For general information about upgrading, see [Upgrading from a previous version](#).

### Licensing changes

2023 licenses are not compatible with Klocwork 2024.3. You need a new license to use the latest version of the product. Contact [license@perforce.com](mailto:license@perforce.com) to obtain a new license.

### Disabled checkers

If you chose to migrate your projects\_root directory, verify that you have the same checker configuration as the previous release [before your first integration build analysis](#).

## What's new in Klocwork 2024.3

Here are the highlights for Klocwork 2024.3. If you're upgrading Klocwork, see the [Limitations](#) for items that might affect your upgrade and usage.

### Validate / Klocwork enhancements

This release includes the following enhancements.

## Basic Klocwork administration tasks no longer consume build license

In this release, we've updated the build license consumption model for kwadmin and validate admin. Basic administrative tasks such as list, get and export will no longer consume a build license. This change simplifies the licensing process and enhances alignment between QAC and Klocwork.

For more information on build license consumption, see [kwadmin](#) or [validate admin](#).

## C and C++ enhancements

This release includes the following enhancements:

- Benefit from improved defect detection, with enhanced support for tracking conditionally freed memory.
- Enjoy increased support for the Bazel build system on both Windows and Linux. See [kwbazel](#).

## Java enhancements

Starting in this release, kwauth defaults to storing credentials in secure storage with Java KeyStore. The environment variable KLOCWORK\_SECURE\_TOKEN\_STORAGE is set to JAVA\_SECRET\_STORAGE for new tokens and is set to empty for pre-existing tokens.

For more information, see [Secure authentication tokens](#).

## Expanded coverage for coding standards

This release includes new and expanded coverage for the following coding standards:

- AUTOSAR
- SEI CERT C and C++
- HKMC C++
- MISRA C 2012 with Amendment 2 (100% coverage can now be obtained with a tool available on-demand from Klocwork Support)
- MISRA C 2004 and 2023
- MISRA C++ 2008 and 2023

## Plugins and extensions

The following features were added to the plug-ins and extensions:

- **Device authorization:** Instead of running kauth externally for the Validate 24.2 and later servers, you can now authenticate directly within the Visual Studio and CLion plug-ins. For Visual Studio, the token is stored in the ltoken file. For CLion, the token can be stored in the credentials or ltoken file, depending on your analysis tool version.

## Checker improvements

### New checkers

Checker	Description
CERT.ALIGN.OF CERT.ANONYMOUS.STRUCT CERT.ANONYMOUS.UNION CERT.ARRAY.LENGTH.ZERO CERT.BITFIELD.NOT.INT CERT.BITFIELD.SIGN.MODIFIER CERT.EMPTY.AGR.INIT CERT.ENUM.LITERAL.TYPE.INT CERT.ENUM.TYPE.SPECIFIER CERT.FIXED.MEMORY.ADDRESS CERT.LITERAL.BINARY CERT.LITERAL.MULTI-BYTE.CHAR	These CERT checkers provide support for CERT MSC14-C: Do not introduce unnecessary platform dependencies.

Checker	Description
CERT.LITERAL.SUFFIX.I64 CERT.NAME.DOLLAR.CHAR CERT.STMT.EXPR CERT.STR.RANGE.FORMATTER CERT.TYPEOF CERT.UNNAMED.MEMBER	
CERT.CHROOT CERT.CHROOT.CHDIR	These CERT checkers provide support for POS05-C: Limit access to files by creating a jail.
CERT.LITERAL.ARRAY CERT.LITERAL.BITFIELD CERT.LITERAL.CHAR.CONST CERT.LITERAL.FLT.CONST CERT.LITERAL.INT.CONST CERT.LITERAL.STR.CONST	These CERT checkers provide support for CERT DCL06-C:(L3) Use meaningful symbolic constants to represent literal values.
CERT.LITERAL.OCTAL	This CERT checker provides support for CERT MSC14-C: Do not introduce unnecessary platform dependencies.
CERT.MEM.PLACEMENTNEW.MISALIGNED CERT.MEM.PLACEMENTNEW.TOOSMALL	These CERT checkers provide support for CERT MEM54-CPP: Provide placement new with

Checker	Description
	properly aligned pointers to sufficient storage capacity.
CERT.MULTI.FUNC.ARG.CALLS	This CERT checker provides support for CERT EXP10-C: Do not depend on the order of evaluation of subexpressions or the order in which side effects take place.
CERT.RTN.FLT.CAST.DBL CERT.RTN.FLT.IMPLICIT.CAST.DBL	These CERT checkers provide support for CERT FLP07-C: Cast the return value of a function that returns a floating-point type.
CERT.SIG.SIG_HANDLER.ASYNC_SAFE	This CERT checker provides support for SIG30-C: Call only asynchronous-safe functions within signal handlers.
CERT.STATIC.SINGLE.USE CERT.TU.UNUSED.GLOBAL.DECL	These CERT checkers provide support for CERT DCL19-C: Minimize the scope of variables and functions.

Checker	Description
CXX.ID_VIS.GLOBAL_VARIABLE.EXTERN CXX.ID_VIS.GLOBAL_VARIABLE.STATIC	These MISRA checkers provide support for MISRA C 2012 and 2023 Rule 8.9 (Advisory): An object should be declared at block scope if its identifier only appears in a single function.
MISRA.DECL.EXTERNAL.MULTIPLE	This MISRA checker provides support for MISRA C 2012 and 2023 Rule 8.5, and for MISRA C 2004, Rule 8.8: An external object or function shall be declared once in one and only one file.
MISRA.DEFINE.EXTERNAL.MULTIPLE	This MISRA checker provides support for MISRA C 2012 and 2023 Rule 8.6, and for MISRA C 2004, Rule 8.9: An identifier with external linkage shall have exactly one external definition.
MISRA.ELIF.DEFINED MISRA.EXPANSION.NARGS MISRA.EXPANSION.UNSAFE MISRA.IF.DEFINED MISRA.INCOMPLETE.STRUCT MISRA.INCOMPLETE.STRUCT.UNNAMED	These MISRA checkers provide support for MISRA C 2012 Rule 1.3: There shall be no occurrence of undefined or critical unspecified behaviour.

Checker	Description
MISRA.INCOMPLETE.UNION MISRA.INCOMPLETE.UNION.UNNAMED	
MISRA.EXT.IDENT.DISTINCT.2012.C90 MISRA.EXT.IDENT.DISTINCT.2012.C99	These MISRA checkers provide support for MISRA C 2012 and 2023 Rule 5.1: External identifiers shall be distinct.
MISRA.EXT.LINKAGE.REDUNDANT.2012	This MISRA checker provides support for MISRA 2012 and 2023 Rule 8.7: Functions and objects should not be defined with external linkage if they are referenced in only one translation unit.
MISRA.MACRO_ARG.EXPRESSION.2012	This MISRA checker provides support for MISRA 2012 and 2023 Rule 20.7: Expressions resulting from the expansion of macro parameters shall be enclosed in parentheses.
MISRA.STRUCT_DEF.HIDDEN.2012	This MISRA checker provides support for MISRA C 2012 Directive 4.8: If a pointer to a

Checker	Description
	structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden.
MISRA.TOKEN.COMMENTED.CODE	This MISRA checker provides support for MISRA C 2004 Rule 2.4; for MISRA C++ 2008 Rules 2-7-2 and 2-7-3; for MISRA C++ 2023 Directive 5.7.2; and for MISRA C 2012 and 2023 Directive 4.4: Sections of code should not be “commented out”.

## Modified checkers

Checker	Description
AUTOSAR.ADD.LITERAL	Finds fewer false positives



Checker	Description
AUTOSAR.ADD.REDEF.DERIVED.FUNC	Finds fewer false positives
CERT.OOP.COPY_MUTATES	Finds fewer false positives
MISRA.IF.UNDEF	Finds fewer false positives
MISRA.LOGIC.NOT_BOOL	Finds fewer false positives
MLK.MUST	Finds fewer false positives
UFM.DEREF.MIGHT	Finds additional defects
UFM.DEREF.MUST	Finds additional defects

### Enabled or disabled checkers

No checkers were added to or removed from the default enabled field of the checker configuration files in this release.

## Taxonomy improvements

As part of the installation, you will find several custom taxonomy files that map Klocwork checkers to [coding standards](#) such as MISRA, CWE, OWASP, and DISA STIG.

Taxonomy	Improvements
autosar_cpp_18_10.tconf and autosar_cpp_18_10_ja.tconf autosar_cpp_18_10_strict.tconf and autosar_cpp_18_10_strict_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"><li>▪ A2-7-2</li></ul>
cert_c_all.tconf and cert_c_all_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"><li>▪ MSC14-C</li><li>▪ POS05-C</li><li>▪ DCL06-C</li><li>▪ EXP10-C</li><li>▪ FLP07-C</li><li>▪ DCL19-C</li><li>▪ MEM00-C</li></ul>
cert_c_rules.tconf and cert_c_rules_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"><li>▪ SIG30-C</li></ul>
cert_cpp.tconf and cert_cpp_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"><li>▪ MEM54-CPP</li></ul>

Taxonomy	Improvements
cwe_all_cxx.tconf and cwe_all_cxx_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> <li>▪ CWE-401</li> </ul>
disa_stig_v4_cxx.tconf and disa_stig_v4_cxx_ja.tconf disa_stig_v5_cxx.tconf and disa_stig_v5_cxx_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> <li>▪ APSC-DV-002400</li> <li>▪ APSC-DV-003170</li> <li>▪ APSC-DV-003320</li> </ul>
disa_stig_v5_java.tconf and disa_stig_v5_java_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> <li>▪ APSC-DV-002440</li> <li>▪ APSC-DV-002550</li> <li>▪ APSC-DV-001820</li> </ul>
Helix QAC taxonomies	Updated the Helix QAC taxonomies to Helix QAC version 2024.3.
iso_iec_ts_17961_c.tconf and iso_iec_ts_17961_c_ja.tconf	Added or modified checker mappings to the following categories: <ul style="list-style-type: none"> <li>▪ 5.18</li> </ul>
misra_c_2012_with_amd2_c11.tconf and misra_c_2012_with_amd2_c11_ja.tconf	Added or modified checker mappings to the following rules and directories: <ul style="list-style-type: none"> <li>▪ Rule 1.3</li> <li>▪ Rule 5.1</li> <li>▪ Rule 8.5</li> </ul>

Taxonomy	Improvements
	<ul style="list-style-type: none"> <li>▪ Rule 8.6</li> <li>▪ Rule 8.7</li> <li>▪ Rule 8.9</li> <li>▪ Rule 20.7</li> <li>▪ Rule 22.1</li> <li>▪ Dir. 4.4</li> <li>▪ Dir. 4.8</li> </ul>
<p>misra_c_2012_with_amd2_c90.tconf and misra_c_2012_with_amd2_c90_ja.tconf</p>	<p>Added or modified checker mappings to the following rules and directories:</p> <ul style="list-style-type: none"> <li>▪ Rule 1.3</li> <li>▪ Rule 5.1</li> <li>▪ Rule 5.2</li> <li>▪ Rule 5.4</li> <li>▪ Rule 5.5</li> <li>▪ Rule 8.5</li> <li>▪ Rule 8.6</li> <li>▪ Rule 8.7</li> <li>▪ Rule 8.9</li> <li>▪ Rule 20.4</li> <li>▪ Rule 20.7</li> <li>▪ Rule 22.1</li> <li>▪ Dir. 4.4</li> <li>▪ Dir. 4.8</li> </ul>
<p>misra_c_2012_with_amd2_c99.tconf and misra_c_2012_with_amd2_c99_ja.tconf</p>	<p>Added or modified checker mappings to the following rules and directories:</p> <ul style="list-style-type: none"> <li>▪ Rule 1.3</li> </ul>

Taxonomy	Improvements
	<ul style="list-style-type: none"> <li>▪ Rule 5.1</li> <li>▪ Rule 8.5</li> <li>▪ Rule 8.6</li> <li>▪ Rule 8.7</li> <li>▪ Rule 8.9</li> <li>▪ Rule 20.7</li> <li>▪ Rule 22.1</li> <li>▪ Dir. 4.4</li> <li>▪ Dir. 4.8</li> </ul>
misra_c_2023_c90.tconf and misra_c_2023_c90_ja.tconf	<p>Added or modified checker mappings to the following rules and directories:</p> <ul style="list-style-type: none"> <li>▪ Rule 5.1</li> <li>▪ Rule 5.2</li> <li>▪ Rule 5.4</li> <li>▪ Rule 5.5</li> <li>▪ Rule 8.5</li> <li>▪ Rule 8.6</li> <li>▪ Rule 8.7</li> <li>▪ Rule 8.9</li> <li>▪ Rule 20.4</li> <li>▪ Rule 20.7</li> <li>▪ Rule 22.1</li> <li>▪ Dir. 4.4</li> <li>▪ Dir. 4.8</li> </ul>
misra_c_2023_c99.tconf and misra_c_2023_c99.tconf misra_c_2023_c11.tconf and misra_c_2023_c11_ja.tconf	<p>Added or modified checker mappings to the following rules and directories:</p>

Taxonomy	Improvements
	<ul style="list-style-type: none"> <li>▪ Rule 5.1</li> <li>▪ Rule 8.5</li> <li>▪ Rule 8.6</li> <li>▪ Rule 8.7</li> <li>▪ Rule 8.9</li> <li>▪ Rule 20.7</li> <li>▪ Rule 22.1</li> <li>▪ Dir. 4.4</li> <li>▪ Dir. 4.8</li> </ul>
misra_cpp_2008.tconf and misra_cpp_2008_ja.tconf	<p>Added or modified checker mappings to the following rules and directories:</p> <ul style="list-style-type: none"> <li>▪ Rule 2-7-2</li> <li>▪ Rule 2-7-3</li> </ul>
misra_cpp_2023.tconf and misra_cpp_2023_ja.tconf	<p>Added or modified checker mappings to the following rules and directories:</p> <ul style="list-style-type: none"> <li>▪ Dir. 5.7.2</li> </ul>

## Improvements to supported compilers

You'll find additional or improved support for the following compilers:

- GNU compiler (gcc, g++ and gcc-based cross-compilers)
- Clang compiler (clang, clang++)
- TI ARM Clang (tiarmclang)
- QNX compiler (qcc)
- iccarm compiler (iccarm)
- Tricore compiler (cctc, ctc, cptc)

For the full list of supported C and C++ compilers, see [C/C++ compilers supported for build integration](#).

## Changes to system requirements

In this release, we've added support for

- Amazon Linux 2 (2.0.20240903.0 Update)
- Ant 1.10.15
- Chrome 117.x to 128.x
- CLion 2023.2.5
- Debian 11.11
- Eclipse 4.33 (2024-09)
- Firefox 118.x to 130.x
- Glibc 2.15 to 2.40
- Gradle 8.10
- IntelliJ IDEA 2023.1.8
- Maven 3.9.9
- Microsoft Edge 117.x to 128.x
- Oracle Linux 8.10, 9.4
- Red Hat Enterprise Linux 8.10
- Suse Enterprise Enterprise 15 SP6

- Ubuntu 22.04.5 LTS
- Visual Studio 2017 version 15.9.66
- Visual Studio 2019 version 16.11.40
- Visual Studio 2022 version 17.11.4
- VS Code 1.83.1 to 1.93.1

In this release, we ended support

- CentOS Linux 7.x
- Chrome 115.x to 116.x
- Firefox 115.x to 117.x
- Microsoft Edge 115.x to 116.x
- Oracle Linux 7.x
- Red Hat Enterprise Linux 7.x
- VS Code 1.80.2 to 1.82.3

For the complete list of supported versions, see [System Requirements](#).

## Fixed issues in Klocwork 2024.3

The following issues were fixed in Klocwork 2024.3.

### General issues

Number	Description
00859971, 00749557, 00751344, 01119112,	Improved support for the QNX compiler with C++ 11/14.



Number	Description
01150496, 01137814, 01158992, 01162319	
00840347, 00798167, 01064861, 01155254	Improved support for the TI Arm Clang compiler.
01146934	Fixed an issue with displaying updated issue counts in Validate when the default view search parameter changes.
01061087, 01055416	Eliminated the need for kwadmin to consume a build license for basic administration actions.
00913219	Added a feature that stores compressed build log files on the server to reduce storage usage.
01061102, 01172754, 01192832, 01172754	Fixed an issue with the dbvalidate cleanup tool.
01105271	Added the option to set runtime timeouts for all C/C++ analysis steps.
01066032	Improved support for the Arm extensions to fix an issue with missing object files.
01121550	Improved the reliability of C++ analysis when using the modern analysis engine.

Number	Description
01124381	Fixed an issue related to SCONF file path matching.
01143937	Updated the icons for Severity and Support Level for English and Japanese in the Validate portal.
01154713	Fixed an issue with uploading CI builds to the project server.
01139893	Fixed an issue with Klocwork projects hanging during analysis.
01163039	Improved the handling of build options when using the C/C++ modern analysis mode.
01157667	Improved support for the IARC/C++ compiler.
01190436	Fixed an issue where the same defect, across streams of the same family, showed different statuses.
01161340	Improved support for Klocwork C/C++ analysis on Android 14.
01200853	Updated the documentation for the Klocwork 2024.2 release notes and Java virtual machine requirements.
01213090, 01210352	Updated the documentation for the MISRA C:2012 and MISRA C:2023 rules mapped to Klocwork checkers.

Number	Description
01210352	Updated the format of the Date Originated column in CSV file downloads to improve usability.
01210352	Updated the documentation for the fields included when exporting a list of issues from Validate.
01215710, 01237080	Fixed an issue with handling of paths for modules in Validate.
01237252	Updated the documentation to include descriptive browser tab names for each page.
01235300	Fixed an issue with displaying renamed project names correctly in Validate reports.

### Checker issues

Number	Description
00912764	Added the new checker MISRA.MACRO_ARG.EXPRESSION.2012.
00624808, 01164040	Reduced false positives for the checker MLK.MUST.

Number	Description
01045924, 01045915	Reduced false negatives for the checker UFM.USE.MUST.
01045919	Reduced false negatives for the checker UFM.USE.MIGHT.
01045924	Reduced false negatives for the checker MISRA.IF.UNDEF.
01198952	Reduced false positives for the checker AUTOSAR.ADD.REDEF.DERIVED.FUNC.
01201231	Reduced false positives for the checker MISRA.LOGIC.NOT_BOOL.
01226335	Reduced false positives for the checker AUTOSAR.ADD.LITERAL.
01201231	Reduced false positives for the checker CERT.OOP.COPY_MUTATES.



This document, as well as the software described in it, is furnished under license and may only be used or copied in accordance with the terms of such license. The information contained herein is the exclusive property of RogueWave Software, Inc. a Perforce company. No part of this documentation may be copied, translated, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Perforce Software, Inc. If you find any problems in the documentation, please report them to us in writing.

Klocwork is a registered trademark of RogueWave Software, Inc., a Perforce company.

All other trademarks are the property of their respective owners. All help content for Klocwork's MISRA checkers is copyright by MISRA Consortium.