



Klocwork 2024.2 Release Notes

Document Number: KW2024_2 _009

Certified version: Klocwork 2024.2

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
ABedford	5.2	Updated for Klocwork 2020.3	2020-09-14
ABedford	5.3	Updated for Klocwork 2020.4	2021-02-24
ADunster	6.0	Updated for Klocwork 2021.1	2021-04-26
ADunster	6.1	Updated for Klocwork 2021.2	2021-08-17

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

Contents

Release Notes Klocwork 2024.2	6
What's new in Klocwork 2024.2	6
Fixed issues in Klocwork 2024.2	18
Limitations	24
Release Notes Klocwork 2024.1	43
What's new in Klocwork 2024.1	43
Fixed issues in Klocwork 2024.1	49

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 50128	EN 50128:2011/A2:2020
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_2_001	Functional Safety Manual for Klocwork
KW2024_2_003	Klocwork ISO 26262 / IEC61508 / EN 50128 / EN50716 Certified Checkers

KW2024_2_005	Klocwork Checker Qualification Pack
KW2024_2_006	Klocwork Architecture

Release Notes Klocwork 2024.2

These release notes cover Klocwork 2023.4 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

2022 licenses are not compatible with Klocwork 2023.4. You need a new license to use the latest version of the product. Contact 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 2024.2

Here are the highlights for Klocwork 2024.2. If you're upgrading, see the [Limitations](#) for items that affect how you use Klocwork. (also available at <https://help.klocwork.com/current/en-us/concepts/whatsnewmain.htm>)

Enhanced security and user experience with SAML and OIDC authentication

You can now integrate your identity provider (IdP) with Validate using Security Assertion Markup Language (SAML) or OpenID Connect (OIDC) authentication to enjoy benefits such as:

- Enhanced security through centralized authentication
- Simplified user management and experience through single sign-on (SSO)

Validate has been tested with the following identity providers:

- SAML: Keycloak, Okta, AWS, Cisco, and Github
- OIDC: Keycloak, Google, Microsoft Entra (formerly Azure AD), and AWS

To learn how to set up and configure SAML and OIDC, see [Setting up SAML access control](#) and [Setting up OIDC access control](#).

Authenticate using application tokens

IMPORTANT: In release 2024.2, [ltokens](#) have been replaced by [application tokens](#). If you relied on ltokens for scripting before release 2024.2, see [Transitioning from ltokens to application tokens](#).

RESTRICTION: To authenticate with a Validate server that is supported by SAML or OpenID Connect, install version 24.2 or newer of kwauth from one of the following packages: kwauthtools, kwbuildtools, or the kw-cmd-installer found in kw-desktop-tools. See [Installing the Auth Tools package](#).

You can now create application tokens in Validate to securely authenticate with SAML or OIDC supported servers for the following tasks:

- **Sign in to the command line tools using [kwauth](#) or [validate auth](#).** This is helpful on headless machines, where you cannot sign in with a username and password in a browser. See [Authentication using application tokens](#).
- **Import projects from Validate or the Web API.** You will need an application token to authenticate instead of a username and password. You will still need a username and password to import projects from servers that use classic authentication. See [Import or migrate projects using application tokens](#) and [Import your existing projects into a new projects root](#).

Note that SAML or OIDC device authorization (including for the desktop plug-ins) happens through the Validate login page, and uses an access code generated by kvaauth or validate auth. To learn more, see [Accessing Validate by Perforce](#).

Manage user sessions and tokens in Validate

Administrators can now manage individual user sessions through Validate. With this permission is enabled, you can log users out of their Validate sessions and revoke user tokens. To learn more, see [Managing user sessions and tokens](#).

Enhanced password security in Validate

If you use basic authentication, you can now implement a secure password policy for Validate accounts. This requires your password to meet the following criteria:

- A minimum of 8 characters
- At least one uppercase letter
- At least one lowercase letter
- At least one number
- At least one special character (such as !, @, # or \$)

Existing passwords are not affected by this new policy. To learn more, see [Enabling secure passwords](#).

Klocwork utility enhancements

You can now specify which [Java Virtual Machine \(JVM\)](#) the Klocwork Java tools use, by setting the KW_JAVA environment variable.

When this variable is set, the Klocwork Java tools will run using the JVM defined by KW_JAVA instead of the default JVM. This allows for greater flexibility and compatibility with different Java environments.

C/C++

The Klocwork analysis engine for C/C++ can be run using classic, standard, or modern mode. If you do not specify an option, standard mode (recommended) is used by default. To learn more, see [Specifying the C/C++ analysis engine mode](#). C#

Java

Instead of having to modify the build specification to focus on a select set of Java files for analysis, you can now use the `--ignore-files` option in `kwandroid`.

Plug-in and tool enhancements

The following enhancements were made to the Klocwork plug-ins and tools:

- Depending on the version of your Validate server, plug-in, and tools, you can now connect to a project or stream in any plug-in using either classic authentication, or [SAML or OIDC authentication](#). Simply refer to the instructions in the documentation for your [desktop analysis tool](#), and follow the prompts on your screen.
- To streamline the deployment of your Klocwork analysis tools in automated environments, the continuous integration tools are now included in the Build Tools package.

Expanded coverage for coding standards

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

- CWE for Kotlin

Plug-ins and extensions

Depending on the version of your Validate server, plug-in, and tools, you can now connect to a project or stream in any plug-in using either classic authentication, or SAML or OIDC authentication. Refer to the instructions in the documentation for your desktop analysis tool, and follow the prompts on your screen.

Checker improvements

New checkers

The following checkers were added in this release:

Checker	Description
MISRA.TOKEN.WRONGESC.C.2004 and MISRA.TOKEN.WRONGESC.CPP.2008	These MISRA checkers provide support for MISRA-C Rule 4.1 (required): Incorrect escape sequence in a literal and MISRA-C++ Rule 2-13-1 (required): Only those escape sequences that are defined in ISO/IEC 14882:2003 shall be used.

Modified checkers

Checker	Description
A_UNUSED.GEN	Finds fewer false positives
AUTOSAR.ADD.ENUM.OP	Finds fewer false positives
FUNCRET.GEN	Finds fewer false positives
LOCRET.RET	Finds fewer false positives
MISRA.ETYPE.INAPPR.CAST.2012	Finds fewer false positives

Checker	Description
MISRA.TOKEN.WRONGESC	Finds fewer false positives
MISRA.VAR.UNIQUE.STATIC	Finds additional defects
MLK.MUST	Finds fewer false positives
NNTS.MUST	Finds fewer false positives
NPD.CHECK.MIGHT	Finds fewer false positives
NPD.FUNC.MIGHT	Finds fewer false positives
NPD.FUNC.MUST	Finds fewer false positives
SV.BRM.HKEY_LOCAL_MACHINE	Overall improvements to the checker
UNINIT.CTOR.MUST	Finds fewer false positives
UNINIT.HEAP.MUST	Finds fewer false positives
UNINIT.STACK.MIGHT	Finds fewer false positives
UNINIT.STACK.MUST	Finds fewer false positives
SV.STRBO.NOBOUND_COPY	New defects detected

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, we offer several custom taxonomy files that map our checkers to coding standards such as MISRA, CWE, OWASP, and DISA STIG.

Taxonomy	New/updated
ert_c_all.tconf and cert_c_all_ja.tconf	Added or modified checker mappings to the following rules: <ul style="list-style-type: none"> ▪ CERT POS02-C
cert_cpp.tconf and cert_cpp_ja.tconf	Substantial reorganization of the cert_cpp.tconf and cert_cpp_ja.tconf taxonomies.
cwe_all_kt.tconf and cwe_all_kt_ja.tconf	Added new taxonomies that map Klocwork Kotlin checkers to the CWE IDs.
Helix QAC taxonomies	Updated the Helix QAC taxonomies to Helix QAC version 2024.2.
misra_c_2023_c99.tconf and misra_c_2023_c99_ja.tconf misra_c_2023_c90.tconf and misra_c_2023_c90_ja.tconf misra_c_2023_c11.tconf and misra_c_2023_c11_ja.tconf misra_c_2012_with_amd2_c99.tconf and	Substantial reorganization of The MISRA C and C++ taxonomies. Each taxonomy is now defined by using a rule-first approach, where the checkers are subcategories of rules in the taxonomies

Taxonomy	New/updated
<p>misra_c_2012_with_amd2_c99_ja.tconf</p> <p>misra_c_2012_with_amd2_c90.tconf and</p> <p>misra_c_2012_with_amd2_c90_ja.tconf</p> <p>misra_c_2012_with_amd2_c11.tconf and</p> <p>misra_c_2012_with_amd2_c11_ja.tconf</p> <p>misra_c_2004.tconf and</p> <p>misra_c_2004_ja.tconf</p> <p>misra_cpp_2023.tconf and</p> <p>misra_cpp_2023_ja.tconf</p> <p>misra_cpp_2008.tconf and</p> <p>misra_cpp_2008_ja.tconf</p>	

Improvements to supported compilers

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

- Clang
- Clang-cl
- GCC
- IAR
- Renesas

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).

2022 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.

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:

- AlmaLinux 9.4
- Amazon Linux 2 (2.0.20240529.0 Update)
- Android Studio Iguana (2023.2.1 Patch 2)

- Chrome 115.x to 126.x
- CLion 2023.1.7, 2023.2.4
- Eclipse 4.32 (2024-06)
- Fedora 40
- Firefox 115.x to 127.x, 115.x ESR
- Glibc 2.39
- Gradle 8.8
- IntelliJ IDEA 2023.1, 2023.2.7
- Microsoft Edge 115.x to 126.x
- openSUSE Leap 15.6
- Oracle Linux 9.4
- Red Hat Enterprise Linux 9.4
- Rocky Linux 9.4
- Ubuntu 22.04 to 22.04.4 LTS
- Visual Studio 2017 version 15.9.63
- Visual Studio 2019 version 16.11.37
- Visual Studio 2022 version 17.10.3
- VS Code 1.80.2 to 1.90.2

In this release, we ended support for:

- Chrome 111.x to 114.x
- Fedora 38
- Firefox 111.x to 114.x
- Jenkins plug-in
- Microsoft Edge 111.x to 114.x
- VS Code 1.76.2 to 1.80.1

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

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 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

Maintenance ending for Klocwork 2022

Maintenance (including end of maintenance and end of sale) for all 2022 versions of Klocwork ended on March 31, 2024. To learn about the support available for all Klocwork releases, see the Klocwork Product Lifecycle.

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.

Discontinuation of issue grouping

Starting from Klocwork 2023.3, issue grouping is turned off by default for new projects.

To help avoid issues, turn off issue grouping before you upgrade Klocwork.

Fixed issues in Klocwork 2024.2

The following issues were fixed in Klocwork 2024.2.

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

General issues

Number	Description
SUPPORT-10910	Improved handling of macro evaluations by the Klocwork compiler.
SUPPORT-28580	Improved handling of <code>if</code> statements with initializers defined in C++ 17 by the Klocwork compiler.
00637318, 00902961, 01025722, 00637583, 01174466, 01063795, 00635318	Added support for integration with external identity providers for authentication in Validate.
00749853	Improved Klocwork C/C++ analyzer stability on Linux.
00924395, 01021186, 01114322, 01141656	Fixed an issue with <code>kwininspect</code> reporting missing entities as errors instead of warnings.

Number	Description
01028712, 00982409	Added an option to <code>kwandroid</code> to exclude Java files when generating a build trace.
01053044	Fixed a discrepancy between the number of system files analyzed show in Reports versus the XRef tab in Validate.
01113787	Added an option to set a threshold for the number of issues of the same kind per file that will be accepted for a build to be uploaded to Validate.
01160894	Improved the MISRA C and C++ taxonomies to align the categories to rules.
01123589	Fixed an issue with the displayed count of parallel loads to streams in projects.
01131058	Updated the CERT level for INT31-C in the CERT C rules taxonomy.
01131449	Updated the documentation for cross-version support for builds.
01133993, 01133952	Updated the documentation for migrating your <code>projects_root</code> directory.
SUPPORT-23416	Updated the documentation to indicate limitations with certain analyzers in the Visual Studio extension.
01121550	Improved support for running analysis on C++ with Qt6 with the Klocwork compiler.

Number	Description
01142779	Improved support for migrating from previous Validate versions when HTTPS is configured.
01142719	Fixed an issue with running kwcheck in the Eclipse plug-in.
01147947	Updated the documentation for installing the Visual Studio plug-in.
01139771	Updated the Japanese documentation for changing project states.
01148604	Updated the documentation for setting project states with kwadmin.
01154191	Updated the documentation for FOSS to include versions for all components.
01160077	Updated the documentation for the the MISRA C++ 2008 and MISRA C++ 2023 rules mapped to Klocwork checkers.
01164257	Added new taxonomies that map Klocwork Kotlin checkers to the CWE IDs.
01116888, 01142556, 01146011, 01057266	Fixed an issue with builds not uploading to Validate for projects where issues had been cited using the Desktop Tools.

Checker issues

Number	Description
SUPPORT-23416	Reduced false positives for the checker LOCRET.RET.
SUPPORT-27884, 00640812, 00637099, 00800040	Reduced false positives for the checkers NPD.FUNC.MIGHT and NPD.FUNC.MUST.
SUPPORT-29382	Reduced false positives for the checker UNINIT.CTOR.MUST.
SUPPORT-6733, SUPPORT-9064, SUPPORT-35751, 00638508, 00622408, 00640811, 00717377, 00780823	Reduced false positives for the checker UNINIT.STACK.MUST.
SUPPORT-30434	Reduced false positives for the checker UNINIT.HEAP.MUST.
SUPPORT-36017	Reduced false positives for the checker UNINIT.STACK.MIGHT.

Number	Description
00696219	Reduced false positives for the checker MLK.MUST.
00741803	Reduced false positives for the checker FUNCRET.GEN.
00773456	Reduced false positives for the checker VA_UNUSED.GEN.
00863551	Improved defect detection for the checker MISRA.VAR.UNIQUE.STATIC.
00911103	Reduced false positives for the checker NPD.CHECK.MIGHT.
00912237	Updated the mapping for POS02-C in the CERT C and C++ taxonomies.
01123048	Reduced false positives for the checker MISRA.ETYPE.INAPPR.CAST.2012.
01124381	Reduced false positives for the checker MISRA.TOKEN.WRONGESC.
01125802, 01148975	Reduced false positives for the checker NNTS.MUST.
01126327	Reduced false positives for the checker AUTOSAR.ADD.ENUM.OP.

Number	Description
01160164	Updated the code sample for the checker AUTOSAR.ARRAY.CSTYLE in the documentation.
01160595	Updated the code sample for the checker CXX.BITOP.BOOL_OPERAND in the documentation.
01106135	Updated the CERT C++ taxonomies to include level information and separated rules and recommendations categories.

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

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.

Using Debian 10.7 and running the Klocwork Desktop Analysis (kwgcheck)

Run this command to install the missing libraries:

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

Previous dbvalidate and dbvalidate cleanup versions should not be used with previous Validate versions that have CI builds

Using the 2023.1-2023.3 versions of dbvalidate and dbvalidate cleanup with projects-root that have CI builds can result in data loss.

Workaround: To run dbvalidate and dbvalidate cleanup before upgrading from Validate versions 2023.1-2023.3, use the versions included in the 2023.4 Validate installation package.

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.

On Windows, integration build results fail to load to the Server

On Windows, when installing Klocwork, do not choose the **Custom** setup type and select the **Klocwork Servers Only** option, because integration build results will fail to load to the Server.

Workaround: Install Klocwork and choose either the **Complete** setup type that installs all program features. Or, choose the **Custom** setup type and select the **Klocwork Servers and Built Tools** option.

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 kloadadb, 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.

Android N Java analysis with Jack toolchain

When building Android N using the Jack compiler, some jar files required for Klocwork Java analysis are not generated during the build process. Therefore, kwbuildproject encounters "Unresolved import", "Unresolved method", and "Unresolved name" semantic errors that affect the accuracy of the analysis results.

Workaround: Open a ticket with Klocwork customer support. Customer support can provide a script that can generate the jar files required for analysis. Run the script after running the kwinject command and before running the kwbuildproject command.

Limitations for C# analysis

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

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

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](#)

Limitations for Java analysis

In release 2024.2, the documentation has been updated to reflect the use of kwciagent run and kwcheck run with Java. In particular, differential analysis with kwciagent for a specified subset of files is not supported for Java projects.

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:

0001	F0001	I0001	I0020
E0013	F0002	I0010	I0021
E0014	F0010	I0011	I0022
E0015	F0011	I0013	I0023

In release 2023.2, the following checkers listed in the Python taxonomy are not supported. They are disabled by default and can be enabled manually:

E0001	F0002	I0020	W4902
E0013	F0010	I0021	W4903
E0014	F0011	I0022	W4904
E0015	I0001	I0023	W4905
E0705	I0010	W0246	
E1143	I0011	W3101	
F0001	I0013	W4901	

Limitation for Validate

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.

In Microsoft Edge, some items may not be clickable

Due to a Microsoft Edge issue, some items in the portal may not be clickable. For more information, see <https://developer.microsoft.com/en-us/microsoft-edge/platform/issues/5782378/>

Workaround: Refresh the page.

The 'Cert C Recommendations' section of the Security Report is omitted.

Workaround: You can generate the compliance report for this taxonomy if needed. Contact support for more information.

Cannot run Klocwork Server and Validate Windows services at the same time

On Windows, if you install either of the Validate or Klocwork Server installations, and set the Windows services to start automatically, then choose to additionally install the other installation, the services will not start. You can run the Windows services for only one of these installations at a time.

Additionally, make sure if you're installing both installations, their web ports and database ports must be different.

Workaround: If you want to run both simultaneously, you must start the servers for at least one installation by using the command line.

CI build paths are not shown in the tree structure of the 'Modules' page in Klocwork Static Code Analysis/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.

An error occurs when attempting to 'Edit All' issues in a project with CI Builds only

When citing issues in Validate from the project's issue list, the following error occurs if you attempt to use **Edit All** to manage your issue list if your project only contains CI Builds:

Database error occurred: There is no valid build for the project

Workaround: Select the issues you want to cite and use **Edit Selected**.

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).

For the Microsoft Visual Studio extension, minor performance degradation when working with server issues if connection to server is lost

A lost server connection causes a delay of up to three seconds when working with server issues, for example, when opening or citing a server issue.

Workaround: Work with local issues only by clicking the "Show local issues only" button.

F1 help does not work when you try to open help for an issue from the Klocwork Issues window in Visual Studio for the Klocwork extension for Visual Studio

If you click an issue in the Klocwork Issues window and try to open the help by pressing F1, the shortcut opens the incorrect help in the Help Viewer.

Workaround: Open the help for the checker by right-clicking on the issue and select **View Checker Documentation** from the **Manage <checker name> Checker** menu.

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

Cannot authenticate a connected desktop project from the VS Code extension

To use a connected desktop project with the VS Code extension in release 2024.2, you must authenticate manually from outside the extension.

Workaround: Run the following command:

```
kwauth --url http(s)://<ip-address>:<port-number> --insecure
```

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 buildspec 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 Eclipse plug-in

Eclipse 4.16 requires additional software for Klocwork Eclipse plug-in to run

If you are using the 4.16 version of Eclipse, you must install the following software in order to run the Klocwork Eclipse plug-in:

- **Eclipse CDT** using Java 11; or, post-installation, update the eclipse.ini file to point to a valid Java 11 installation.
- **Eclipse Java Development Tools** by using the Help --> Install New Software command.

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.

WARN: Do not use URL connection as JarURLConnection' message in IntelliJ IDEA log

If you see this warning in IntelliJ IDEA in your idea.log, you can safely ignore it.

Limitations for the CLion plug-in

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.

'java.sql.SQLException: SQLite.Exception: no such column: in_connected' error

This error occurs if you are trying to use a 2022.1 or newer version of the Klocwork Desktop plugin for CLion with a 2021.4 or older version of the Klocwork tools.

Workaround: If you are still using an older version (2021.4 or older) of the Klocwork tools, do not upgrade the Klocwork CLion plugin to 2022.1 or newer to avoid this error.

Limitations for Klocwork extensibility

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.

Checker limitations on Windows

kwcreatechecker.exe creates 64-bit checkers by default. If you want to create 32-bit checkers (that is, checkers compatible with a pre-2020.4 version of Klocwork) you must use the --force-32bit option.

If you use pre-2020.4 KAST / AST checkers or pre-2020.3 PATH checkers, you must add the --force-32bit option to all builds (kwbuildproject, kwcheck, and so on).

If you use pre-2020.4 KAST / AST checkers or pre-2020.3 PATH checkers and want to create new checkers, you must do one of the following:

- Rebuild all your old checkers by using the 2020.4 version of kwcreatechecker.exe or by using a 64-bit compiler.
- Build your new checkers in a pre-2020.4 version.

You cannot mix 32-bit and 64-bit checkers:

- If you build any 32-bit checkers, you can only use them in the 2020.4 Visual Studio plugin if the Force 32-bit Analysis option is enabled.
- If you build any 64-bit checkers, you can only use them in the 2020.4 Visual Studio plugin if the "Force 32-bit" Analysis option is disabled.
- If you build any 32-bit checkers, you cannot use them in Eclipse-based plugins.
- When using 64-bit path checkers, you will get the following warning (note that it is safe to ignore this warning):

```
kwcc: Warning: cannot open plugin library 'c:\Klocwork Server\plugins\ix86-pc-win32\&lt;checker>.dll': The specified module could not be found.
```

Checker limitations on Linux

32-bit backward compatibility for custom checkers is no longer supported and the option '--force-32bit' is deprecated. You must rebuild all of your old checkers by using a 64-bit compiler. Contact support for more information.

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 2024.1

These release notes cover Klocwork 2024.1 and include information about what's new in this release, and issues we've fixed since the last release,

What's new in Klocwork 2024.1

Here are the highlights for Klocwork 2024.1. If you're upgrading, see the [Limitations](#) for items that affect how you use Klocwork.

Take advantage of upgrade and migration improvements

You can now:

- Minimize maintenance and downtime by starting the Validate Server before projects have been migrated. Successful migrations will be available immediately, and failed migrations will be automatically disabled.
- Apply fixes and remigrate problematic projects without interrupting the use of successfully migrated projects.
- Save time by migrating individual projects without having to restart the Validate Server.

Manage project states

You can assign [project states](#) using the kwadmin command, to indicate whether a project is active, inactive, or unavailable, and whether it is being migrated or deleted. Project states are visible in Validate and the Web API.

In addition, you can assign project states to restrict edit and view access during maintenance operations or backup creation.

Exclude individual projects from migration

To help speed up migration, you can exclude specific projects. Excluded projects will not be migrated and will appear as disabled in the Validate tools. Later, you can migrate and enable excluded projects individually without restarting the Validate server. For details, see [Migrate your projects_root directory](#).

Prioritize migration of specific projects

To prioritize important projects, you can assign a priority list which specifies the order in which projects will be migrated. Projects not included in the list will be migrated in the order of their project id. To learn more, see [Migrate your projects root directory](#).

Import or duplicate existing projects

You can now import a project from another Validate server even if the project name matches an existing project on the target server. For more information, see [Import your projects and server settings](#).

Explore new capabilities in Validate

You can now:

- Allow specific users to change an issue's owner but not its status by assigning the permission change issue owner.
- Benefit from expanded search capabilities with increased support for the use of * and ** wildcard characters in modules.

Preserve license logs for audit purposes in Klocwork and Validate

You can now preserve logs for auditing purposes, by appending the logs to your license server license.report.log file instead of overwriting the file after the server restarts. To learn about the append.license.logging setting, see [kwservice](#).

Analyze projects that were built with Bazel

You can analyze projects built with Bazel by using the [kwbazel](#) command on Linux or the [--bazel option in kwandroid](#) for Android projects. The kwbazel command supports build integration for C/C++, C#, and Java projects, allowing custom compilation mnemonics handling through configuration. Bazel's virtual_includes are also supported.

Download issue list as CSV

You can now download your list of issues as a CSV file. The CSV download button can be found at the top of the Issues page.

Expanded coverage for coding standards

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

- CWE for C and C++
- MISRA C++:2023

Plugins and extensions

VS Code

To save you time, VS Code will run differential analysis when you open or change a file instead of analyzing the whole build specification.

Visual Studio

The status and info bars are improved to highlight important information and reduce visual distractions.

Checker improvements

New checkers

No checkers were added in this release.

Modified checkers

Checker	Description
Buffer overflow checkers	Finds additional defects
INVARIANT_CONDITION.UNREACH	Finds fewer false positives
MISRA.EXPR.PARENS.REDUNDANT	Finds additional defects
NPD.FUNC.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	New/updated
cwe_all_cxx.tconf and cwe_all_cxx_ja.tconf	<p>Added or modified checker mappings to the following weaknesses:</p> <ul style="list-style-type: none"> ▪ CWE-126

Taxonomy	New/updated
	<ul style="list-style-type: none"> ▪ CWE-192 ▪ CWE-195 ▪ CWE-197 ▪ CWE-704
Helix QAC taxonomies	Updated the Helix QAC taxonomies to Helix QAC version 2024.1.
misra_cpp_2023_certified.tconf and misra_cpp_2023_certified_ja.tconf	Added new taxonomies that map Klocwork checkers to the MISRA C++:2023 standards.

Improvements to supported compilers

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

- Clang
- Clang-cl
- Tasking Tricore

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

- AlmaLinux 9.3
- Amazon Linux 2 (2.0.20240124.0 update)
- Android Studio Giraffe 2022.3.1 Patch 4, Hedgehog 2023.1.1 Patch 4, Iguana 2023.2.1 RC2
- Chrome 111.x to 122.x
- CLion 2023.1.6, 2023.2.3
- Debian 11.9
- Eclipse 4.30 (2023-12)
- Fedora 39
- Firefox 111.x to 122.x, 115.x ESR
- Gradle 8.6
- IntelliJ IDEA 2023.1.6, 2023.2.6
- Jenkins 2.445
- Microsoft Edge 111.x to 121.x
- Oracle Linux 9 to 9.3
- Red Hat Enterprise Linux 9 to 9.3
- Rocky Linux 9.3
- Visual Studio 2017 version 15.9.60, 2019 version 16.11.34, 2022 version 17.9.0
- VS Code 1.76.2 to 1.84.1

In this release, we've ended support for

- Chrome 109.x to 110.x
- Fedora 37
- Firefox 107.x to 110.x
- Microsoft Edge 108.x to 110.x
- VS Code 1.74.3 to 1.76.

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

Fixed issues in Klocwork 2024.1

The following issues were fixed in Klocwork 2024.1.

General issues

Number	Description
00997770, 00862478, 01068007, SUPPORT-29704	Add support for the Bazel build system for Linux.
00623216, SUPPORT-36523	Fixed an issue with displaying groups in Validate user administration.
00610099, SUPPORT-42584	Fixed an issue with user permissions for citing when using the VS Code plugin.
00861782, SUPPORT-44277	Fixed an issue with failures for selected checks by improving support for running QDP with Git Bash on Windows.
00810392	Added a permission in Klocwork that allows users to change the owner of an issue.
00861988, 01046134	Fixed an issue with generating the HIS Metrics taxonomy report.

Number	Description
00873496	Fixed an issue with creating history entries when comment text is cleared in Validate and the web API.
00886800, 01068007	Improved the CWE for C/C++ taxonomies.
00912237	Improved the CERT for C/C++ taxonomies.
00908912, 00969777	Fixed an issue with project builds hanging on upload.
00925046	Fixed an issue with preserving a setting to append license logs when restarting the Validate server.
01006115	Added support for the Validate issue search to suggest all possible keywords.
01008226	Fixed an issue with the project_dir variable target for duplicate projects.
01008237	Made issues accessible in Validate by improving the handling of paths with special characters.
01018384	Updated the documentation for API metrics.
00608650, 00798247	Added support for differential analysis in VSCode on file save and open.

Number	Description
01026629	Fixed an issue with syncing defects between projects with a large number of issues.
01027162	Fixed an issue with stability of the C/C++ analyzer.
01037562	Updated the documentation for the release notes and checker reference pages.
01037891	Improved translation related to CI builds in the Validate portal.
01038605	Updated the documentation for limitations.
00729313	Added support for individual migration of projects when upgrading.
00931333	Fixed an issue with C++ builds that have build specifications generated by kwandroid.
01041496	Fixed an issue with converting build trace files to build specifications.
01045446	Updated the documentation for license servers.
01048036	Updated the documentation for using a secure Validate Server connection.

Number	Description
01044351	Updated the documentation for kwcheck.
01043917	Improved handling of options for selected Clang based compilers with kwinject.
01051199	Fixed an issue with the project configuration report in Validate for Kotlin project.
01060964, 01065304	Updated the documentation for the command line install of the Klocwork desktop tools.
01062612	Updated the description for the ci_issue_details action on the Validate server API page.
01106135	Updated the CERT C and Java taxonomies to include level information for category name and meta data.
01039013	Fixed an issue with C++ checkers generating results for C files.
01065304	Updated the documentation for unattended installation of Validate server.

Checker issues

Number	Description
00696208	Improved defect detection for the NPD.FUNC.MUST checker.
00742013, 01003753	Improved defect detection for the buffer overflow checkers.
00971243	Reduced false positives for the checker INVARIANT_CONDITION.UNREACH.
01071685	Improved defect detection for the checker MISRA.EXPR.PARENS.REDUNDANT.



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.