klocwork
by Perforce

Klocwork 2020.4

Release Notes

PERFORCE

www.perforce.com

Contents

Release notes

These release notes cover Klocwork 2020.4 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.

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

2019 licenses are not compatible with Klocwork 2020.4. 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 2020.4

Here are the highlights for Klocwork 2020.4. If you're upgrading, also see the Limitations on page 35 for items that affect how you use Klocwork.

Java improvements

- This release includes a significant improvement to our analysis of Java 9 source code.
- We've improved Java analysis on Android 10 and 11.
- We've also added support for a maven wrapper script, kwmavenw.
- New checkers include several new checkers to identify the CWE Top 25 Most Dangerous Software Weaknesses, with new and improved coverage for rules CWE-269, CWE-434, CWE-502, CWE-732, as well as a new checker called JD.CAST.DOWNCAST that flags possible ClassCastExceptions for subtypes.

C# enhancements

- Our C# analysis now runs on multiple cores and threads, resulting in significantly faster analysis times.
- We've added 12 new C# checkers aimed at detecting security, efficiency, and application stability issues
- We've also focused on delivering a set of fixes and improvements that improve analysis accuracy.

C/C++ analysis improvements

- Our entire C/C++ analysis tool chain on Windows and most of Linux is now 64-bit.
- We've made improvements to multiple C/C++ checkers, including preprocessor updates that enhance defect detection for some MISRA rules. We've also added an explicit configuration for essential type checkers for MISRA C 2012 so that you can define custom boolean types.
- We've updated several taxonomies, including AUTOSAR, ISO IEC TS 17961, and have added a new 2020 CWE Top 25 taxonomy.

CWE Top 25

The checkers we've developed for Java and C# add or improve coverage for several 2019 CWE Top 25 Most Dangerous Software Errors. For more information, see 2019 CWE Top 25 Most Dangerous Software Errors mapped to Klocwork checkers.

We've also added three new taxonomies that map Klocwork checkers for C/C++, C#, and Java to the 2020 CWE Top 25 Most Dangerous Software Errors. For more information, see 2020 CWE Top 25 Most Dangerous Software Errors mapped to Klocwork checkers.

Visual Studio Code

Our new Klocwork extension for Visual Studio Code helps you detect and fix issues before check-in. Our extension supports C/C++, Java, C# and mixed projects and solutions. For more information, see Getting started with Klocwork extension for Visual Studio Code.

Security compliance report

Our new compliance report makes it easy to generate PDF and Microsoft Word documents for tracking and long term compliance requirements when following industry standards such as MISRA, CWE, and AUTOSAR. For more information, see Creating a compliance report.

Klocwork checker improvements

From release to release, we improve issue detection to bring state-of-the-art capabilities to our customers. As a result, expect your analysis results to change as accuracy and coverage improve.

New Klocwork checkers

Checker	Description
CS.DBZ.CONST	This C# checker flags situations in which a zero constant value is used explicitly as a divisor of a division or modulo operation.
CS.DBZ.CONST.CALL	This C# checker flags situations in which an explicit zero constant value is passed directly to a function call and might be used as a divisor of a division or modulo operation without checking it for the zero value.
CS.DBZ.GENERAL	This C# checker flags situations in which a variable that has been assigned a zero constant value locally, or as the result of a function call, might subsequently be used explicitly or passed to a function that might use it as a divisor of a division or modulo operation, without checking it for the zero value.
CS.DBZ.ITERATOR	This C# checker flags situations in which a loop iterator that has been assigned a zero constant value in the execution of the loop might subsequently be used explicitly or passed to a function that might use it as a divisor of a division or modulo operation without checking it for the zero value.
CS.RESOURCE.AUTOBOXING	This C# checker reports any instance of automatic conversion of a value type into a reference type object due to auto-boxing, whenever such conversion takes place inside a loop body.
CS.RESOURCE.UNBOXING	This C# checker reports a defect whenever a value is extracted from object inside a loop body due to unboxing.
CS.SV.TAINTED.CALL.LOOP_BOUND.RESOURCE	This C# checker flags code where an unvalidated integer data is passed as a call argument to a function where it is used as a loop boundary while any resource, managed or unmanaged, is allocated within that loop.

Checker	Description
CS.SV.TAINTED.DESERIALIZATION	This C# checker flags code where unvalidated data is passed as an argument to an object constructor call, inside which it is used for any internal members assignment.
CS.SV.TAINTED.FMTSTR	This C# checker flags instances of external input-controlled data used as format strings.
CS.SV.TAINTED.INJECTION	This C# checker flags code where a tainted string can be used as a parameter for routines that invoke external processes.
CS.SV.TAINTED.PATH_TRAVERSAL	This C# checker reports defects when external strings that are used as parts of file paths are not checked properly.
CS.SV.USAGERULES.PERMISSIONS	This C# checker finds instances of calls for privilege elevation and flags them for review.
JD.CAST.DOWNCAST	This Java checker reports a defect when an object of type B is cast to type C, where B and C are sub types of Object A.
SV.DATA.FILE	This Java checker detects files with certain extensions that may be automatically processed under certain environments, allowing attackers to inject malicious files if not checked.
SV.PERMS.HOME	This Java checker reports a defect when a new file or directory is created in a user home directory without setting the correct permissions.
SV.PERMS.WIDE	This Java checker reports an issue when a folder or file is given permissions that are readable/writable/executable by other users or writable by groups.
SV.PRIVILEGE.MISSING	This Java checker reports an issue when the software does not properly assign, modify, track, or check privileges for an actor, creating an unintended sphere of control for that actor.
SV.SERIAL.NOFINAL	This Java checker reporst an issue for a class when this class directly or indirectly implements the 'java.io.Serializable' interface, but the declared method 'readObject' or 'writeObject' is not declared as final.
SV.SERIAL.OVERRIDE	This Java checker reports an issue for a class when the class directly or indirectly implements the java.io.Serializable interface but the declared method 'readObject' is invoking overridable methods.
SV.XSS.COOKIE	This Java checker reports an issue when a cookie that is used to store a session ID for a client's interaction with a website, so that the

Checker	Description
	request made by the client can be validated, is added to the HttpServletResponse object without setting the setHttpOnly flag.

Modified Klocwork checkers

Checker	Description
MISRA C 2012 Essential type checkers	Added the ability to define custom boolean types
ABV and SV.TAINT checkers	New defects detected
CS.NRE.FUNC.MUST	New defects detected
CS.RESOURCE.LOOP	Fewer false positives are expected
CS.RLK	New defects detected
MISRA.ARRAY.VAR_LENGTH.2012	Fewer false positives are expected.
MISRA.ETYPE.ASSIGN.2012	Fewer false positives are expected
MISRA.FUNC.DECL.AFTERUSE	Fewer false positives are expected
MISRA.IF.NO_ELSE	Fewer false positives are expected
MISRA.TOKEN.OCTAL.INT	Fewer false positives are expected
NPD.FUNC.MUST	Fewer false positives are expected
SV.EXEC.PATH	Fewer false positives are expected

Enabled or disabled checkers

The following checkers were added to the default `enabled` field of the checker configuration files for this release:

- CS.SV.TAINTED.CALL.LOOP_BOUND.RESOURCE
- CS.SV.TAINTED.FMTSTR
- CS.SV.TAINTED.INJECTION
- CS.SV.TAINTED.PATH_TRAVERSAL
- JD.CAST.DOWNCAST
- SV.DATA.FILE
- SV.PERMS.HOME
- SV.PRIVILEGE.MISSING
- SV.SERIAL.OVERRIDE
- SV.XSS.COOKIE

Taxonomy improvements

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

Taxonomy	New/Updated
• autosar_cpp_17_10.tconf and autosar_cpp_17_10_ja.tconf • autosar_cpp_18_03.tconf and autosar_cpp_18_03_ja.tconf • autosar_community_cpp14_19_03.tconf and autosar_community_cpp14_19_03_ja.tconf	Deprecated taxonomies that are replaced by autosar_cpp_18_10.tconf and autosar_cpp_18_10_ja.tconf.

Taxonomy	New/Updated
• cwe_2011_top_25_cxx.tconf and cwe_2011_top_25_cxx_ja.tconf • cwe_2011_top_25_java.tconf amd cwe_2011_top_25_java_ja.tconf	Deprecated taxonomies that are replaced by: • cwe_2020_top_25_cs.tconf and cwe_2020_top_25_cs_ja.tconf • cwe_2020_top_25_cxx.tconf and cwe_2020_top_25_cxx_ja.tconf • cwe_2020_top_25_java.tconf and cwe_2020_top_25_java_ja.tconf
cwe_2019_top_25_cs.tconf and cwe_2019_top_25_cs_ja.tconf	Mapped checkers to the following rules: • CWE-20 • CWE-22 • CWE-78 • CWE-269 • CWE-400 • CWE-426 • CWE-502
cwe_2019_top_25_cxx.tconf and cwe_2019_top_25_cxx_ja.tconf	Mapped checkers to the following rules: • CWE-79 • CWE-200
cwe_2019_top_25_java.tconf and cwe_2019_top_25_java_ja.tconf	Mapped checkers to the following rules: • CWE-400 • CWE-434 • CWE-502 • CWE-732 • CWE-798 Removed references to the following rules: • CWE-287
cwe_all_cs.tconf and cwe_all_cs_ja.tconf	Mapped checkers to the following rules: • CWE-20 • CWE-22 • CWE-78 • CWE-94 • CWE-269 • CWE-400 • CWE-426 • CWE-502 • CWE-1235
cwe_all_cxx.tconf and cwe_all_cxx_ja.tconf	Mapped checkers to the following rules: • CWE-79 • CWE-200
cwe_all_java.tconf and cwe_all_java_ja.tconf	Mapped checkers to the following rules: • CWE-259 • CWE-269

Taxonomy	New/Updated
	• CWE-434 • CWE-502 • CWE-732 • CWE-1004 Removed references to CWE-130.
iso_iec_ts_17961_c.tconf and iso_iec_ts_17961_c_ja.tconf	Mapped checkers to the following rules: • 5.07 • 5.11 • 5.13 • 5.17 • 5.19 • 5.39 • 5.40

Improvements to supported compilers

We've improved support for the following compilers:

- ARM
- Clang
- Cosmic
- GNU
- Intel C++
- Microsoft Visual C++
- TI C7000 Optimizing C/C++
- TI tms320c28x
- Wind River Diab

For the full list of supported C/C++ compilers, see C/C++ compilers supported for build integration.

End of support announcements for 2021

Beginning with Klocwork 2021.1 the following operating systems and installers will not be supported:

- AIX
- Solaris
- Klocwork 32-bit installers

Klocwork 2020.4.1 Service Release

Early in 2021 we will release Klocwork 2020.4.1. This Service Release will upgrade all components of the Linux analysis tool chain to the 64-bit architecture.

Changes to the Path API

In Klocwork 2016, we made a number of changes to the C++ version of our Path API. Chapter 2 of the Klocwork C/C++ Path Analysis API Reference contains a list of deprecated functions and provides a proposed replacement for each. As of Klocwork 2020.3, these functions are fully deprecated and checkers will fail to load if one of the deprecated methods is used. For more information, see Important changes to the Path API in version 11.2.

Solaris and AIX support

Downloads for Solaris and AIX are now by request. To obtain Solaris or AIX product downloads, please contact Klocwork Customer Support.

Solaris installation packages, build tools, and desktop tools are available for Klocwork release 2020.1. Downloads are not available for later releases.

Licensing

2019 licenses are not compatible with Klocwork 2020.4. You need a new license to use the latest version of the product. Contact license@perforce.com to obtain a new license.

We upgraded the version of FlexNet Publisher that we support for Windows and Linux to FlexLM 2019 R3 SP1 (11.16.5.1). If you are using your own FlexNet Publisher license server, ensure you upgrade to this or a newer version. You can also use the license server included with Klocwork 2020.1 and beyond.

Maintenance for Klocwork 2018 ended

Maintenance for all versions of Klocwork 2018 ended February 29, 2020. The end of maintenance (EOM) date and end of sale (EOS) date was also February 29, 2020. For information about the availability of support for any release of Klocwork, see the Klocwork Product Lifecycle.

Portal licensing changes

Klocwork has implemented additional licensing checks related to running the Klocwork Server, which, among other things, underpins the Klocwork portal. We recommend you validate your licensing needs to ensure you have a sufficient number of web service licenses.

Changes to system requirements

This section lists changes to the system requirements. For the complete list of supported versions, see System requirements. We've added support for the following:

- Windows 10 versions above 1809 to 20H2
- Amazon Linux 2 version 2.0.20200904.0
- Debian 9.13 and 10.6
- Red Hat Enterprise Linux version 7.9
- Oracle Linux version 7.9
- CentOS version 7.9
- Ubuntu versions 20.04.1 LTS and 20.10
- SUSE Enterprise 15 SP2
- Eclipse version 4.17
- Android Studio versions above 2.3.2 to 4.0.2
- Visual Studio 2017 versions up to 15.9.28
- Visual Studio 2019 versions up to 16.7.6
- IntelliJ IDEA 2020.1 versions up to 2020.1.4
- IntelliJ IDEA 2020.2 versions up to 2020.2.3
- CLion 2020.1 versions up to 2020.1.3
- CLion 2020.2 versions up to 2020.2.4
- Internet Explorer versions above 11.0.195 to 11.0.210
- Edge versions 84.x and 85.x
- Firefox versions 82.x
- Chrome versions above 83.x to 86.x
- Jenkins versions above 2.243 to 2.263
- Gradle versions above 6.5.1 to 6.7
- Visual Studio Code versions 1.49.3 to 1.51.1

We no longer provide support for the following:

- SUSE Enterprise Leap 15.1
- AIX 7.2 TL2

Changes to commands, tools, and options

In the Visual Studio extension, we added the ability filter issues by owner and by taxonomy. For more information, see Control which issues you see.

We added a check box to both the Eclipse plugin and to the Klocwork Desktop GUI to automatically use a loopback interface when running the kwcheck command. For more information, see Use loopback interface for internal connections.

We added the --force-clr option to kwbuildproject so that you can force the analysis engine to analyze C++/CLI source files containing Managed C++ code.

What's new in Klocwork 2020.3

Here are the highlights for Klocwork 2020.3. If you're upgrading, also see the Limitations on page 35 for items that affect how you use Klocwork.

Java

Building on our improvements to have full support for Java 9 in Klocwork 2020.2, we have added partial support up to Java 11.

We've made improvements to better support for Java constructions such as

- enums
- interfaces
- annotations
- lambda functions
- wildcards

We've added a substantial number of new checkers that map to the 2019 CWE Top 25 Most Dangerous Software Errors:

- CWE-611: SV.XXE.DBF, SV.XXE.SF, SV.XXE.SPF, SV.XXE.TF, SV.XXE.XIF, SV.XXE.XRF
- CWE-426: SV.EXEC.PATH
- CWE-400: JD.INF.ALLOC
- CWE-20: SV.LOADLIB.INJ

We improved the 2019 top-25 CWE taxonomy for Java by simplifying the mapping structure, mapping existing checkers SV.PASSWD.PLAIN and SV.WEAK.CRYPT to CWE-287, and correcting a small number of incorrect mappings.

We've also made improvements to existing checkers:

- split JD.CAST.COL into JD.CAST.COL.MIGHT and JD.CAST.COL.MUST
- reduced false positives for SV.EXPOSE.MUTABLEFIELD

CWE Top 25

The checkers we've developed for Java and C# add coverage for several additional 2019 CWE Top 25 Most Dangerous Software Errors. For more information, see 2019 CWE Top 25 Most Dangerous Software Errors mapped to Klocwork checkers.

C# enhancements

In this release we've continued improving support for C# by

- developing six new security-focused checkers that map to the 2019 CWE Top 25 Most Dangerous Software Errors
- improving analysis accuracy
- adding support for custom Path checkers in C# analysis. For help developing custom Path checkers, contact Static Code Analysis Professional Services to discuss assistance via a services engagement.

C/C++ analysis improvements

We have updated our C/C++ Logical Error Finder to 64-bit on Windows, which enables Klocwork analysis to run to completion on very large and complex compilation units. We also now support 64-bit custom checkers.

We have seen minor performance improvements on some of our OSS test projects.

We've also improved how we handle new and delete keywords and initializer lists.

MISRA C 2012 Amendment 2 (C11)

We've added a new taxonomy that maps Klocwork checkers to MISRA C 2012 Amendment 2 (C11). For more information, see MISRA C 2012 with Amendment 2 (C11).

Option to rebuild Lucene index

We've added an option to the dbvalidate tool that rebuilds the Lucene index for the specified project, which often reduces the size of the index. For more information, see Validate your database (mandatory).

Klocwork checker improvements

From release to release, we improve issue detection to bring state-of-the-art capabilities to our customers. As a result, expect your analysis results to change as accuracy and coverage improve.

New Klocwork checkers

Checker	Description
CS.INFORMATION_EXPOSURE.ALL	This C# checker flags potentially unintended logging or printing to the console of any program data.
CS.INFORMATION_EXPOSURE.ATTR	This C# checker flags potentially unintended logging or printing to the console of data fields specifically marked with the attribute [SecurityCritical] or [SecuritySafeCritical].
CS.RESOURCE.LOOP	This C# checker reports a defect whenever any object is created inside the body of a loop that has no explicit exit condition.
CS.SV.TAINTED.CALL.GLOBAL	This C# checker flags whenever tainted data is used to assign a globally visible data field via a function call.
CS.SV.TAINTED.GLOBAL	This C# checker flags whenever tainted data is used to assign a globally visible data field.
CS.SV.TAINTED.LOOP_BOUND.RESOURCE	This C# checker flags code where an unvalidated argument is used as a loop boundary while any resource, managed or unmanaged, is allocated within the loop.
JD.CAST.COL.MIGHT	This Java checker flags when more than one type is stored in the same collection (Map or List) and at least one of the stored types is related to the type used for a type cast.
JD.CAST.COL.MUST	This Java checker flags when none of the types used to store values in collections is related to the type used in an immediate cast.
JD.INF.ALLOC	This Java checker flags when large sections of memory are consumed within an infinite loop and there is no verification of available memory.

Checker	Description
SV.EXEC.PATH	This Java checker flags when an application searches for critical resources by using a short or relative path that can point to resources that are not under the application's direct control.
SV.LOADLIB.INJ	This Java checker flags the use of 'System.loadLibrary' or 'Runtime.loadLibrary', both of which are vulnerable to environment injection.
SV.TAINTED.XSS.REFLECTED	This C/C++ checker flags potential cross-site scripting issues for CGI scripts (web servers that use a Common Gateway Interface).
SV.XXE.DBF	This Java checker flags when XML input is processed by a weakly-configured XML parser, DocumentBuilderFactory.
SV.XXE.SF	This Java checker flags when XML input is processed by a weakly-configured XML parser, SchemaFactory.
SV.XXE.SPF	This Java checker flags when XML input is processed by a weakly-configured XML parser, SAXParserFactory.
SV.XXE.TF	This Java checker flags when XML input is processed by a weakly-configured XML parser, TransformerFactory.
SV.XXE.XIF	This Java checker flags when XML input is processed by a weakly-configured XML parser, XMLInputFactory.
SV.XXE.XRF	This Java checker flags when XML input is processed by a weakly-configured XML parser, XMLReaderFactory.

Modified Klocwork checkers

Checker	Description
CWARN.BITOP.SIZE	Fewer false positives are expected
LV_UNUSED.GEN	Fewer false positives are expected
MISRA.COMP.WRAPAROUND	Fewer false positives are expected
MISRA.CVALUE.IMPL.CAST	New defects detected and fewer false positives are expected
MISRA.ETYPE.ASSIGN.2012	New defects detected
MISRA.FUNC.NOPROT.CALL	Fewer false positives are expected
NPD.FUNC.MUST	New defects detected and fewer false positives are expected
STRONG.TYPE.ASSIGN.INIT	Fewer false positives are expected
STRONG.TYPE.ASSIGN.RETURN	Fewer false positives are expected

Checker	Description
SV.EXPOSE.MUTABLEFIELD	Fewer false positives are expected
UNINIT.STACK.MUST	Fewer false positives are expected

Enabled or disabled checkers

The following checkers were added to the default `enabled` field of the checker configuration files for this release:

- CS.INFORMATION_EXPOSURE.ATTR
- CS.RESOURCE.LOOP
- CS.SV.TAINTED.LOOP_BOUND.RESOURCE
- JD.CAST.COL.MUST
- JD.INF.ALLOC
- SV.EXEC.PATH
- SV.LOADLIB.INJ
- SV.XXE.DBF
- SV.XXE.SF
- SV.XXE.SPF
- SV.XXE.TF
- SV.XXE.XIF
- SV.XXE.XRF

The checker JD.CAST.COL is deprecated; use JD.CAST.COL.MIGHT and JD.CAST.COL.MUST instead.

The checkers CS.WRONG.CAST and CS.WRONG.CAST.MIGHT are no longer enabled by default.

Taxonomy improvements

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

Taxonomy	New/Updated
• cwe_2019_top_25_cs.tconf and cwe_2019_top_25_cs_ja.tconf • cwe_all_cs.tconf and cwe_all_cs_ja.tconf	We added references to the following checkers: CWE-20 • CS.SV.TAINTED.CALL.GLOBAL • CS.SV.TAINTED.GLOBAL • CS.SV.TAINTED.LOOP_BOUND.RESOURCE CWE-200: • CS.INFORMATION_EXPOSURE.ALL • CS.INFORMATION_EXPOSURE.ATTR CWE-400 • CS.RESOURCE.LOOP • CS.SV.TAINTED.LOOP_BOUND.RESOURCE
• cwe_2019_top_25_java.tconf and cwe_2019_top_25_java_ja.tconf • cwe_all_java.tconf and cwe_all_java_ja.tconf	We added references to the following checkers: CWE-20 • SV.STRUTS.NOTVALID • SV.STRUTS.VALIDMET

Taxonomy	New/Updated
	• SV.DOS.ARRINDEX • SV.LOADLIB.INJ CWE-200 • SV.IL.DEV • SV.IL.FILE CWE-287 • SV.PASSWD.PLAIN • SV.WEAK.CRYPT CWE-400 • JD.INF.ALLOC CWE-426 • SV.EXEC.PATH CWE-611 • SV.XXE.DBF • SV.XXE.SF • SV.XXE.SPF • SV.XXE.TF • SV.XXE.XIF • SV.XXE.XRF Updated the reference for SV.CSRF.GET to map to CWE-352. We updated the reference for RLK.FIELD to map to CWE-772. For cwe_all_java.tconf and cwe_all_java_ja.tconf we also mapped the checker SV.LOADLIB.INJ to CWE-114.
misra_c_2012_with_amd2_c11.tconf and misra_c_2012_with_amd2_c11_ja.tconf	These are new taxonomies that map Klocwork checkers to MISRA C:2012 Amendment 2.

Improvements to supported compilers

We've improved support for the following compilers:

- Clang
- GNU

For the full list of supported C/C++ compilers, see C/C++ compilers supported for build integration.

Changes to the Path API

In Klocwork 2016, we made a number of changes to the C++ version of our Path API. Chapter 2 of the Klocwork C/C++ Path Analysis API Reference contains a list of deprecated functions and provides a proposed replacement for each. As of Klocwork 2020.3, these functions are fully deprecated and checkers will fail to load if one of the deprecated methods is used. For more information, see Important changes to the Path API in version 11.2.

Solaris and AIX support

Downloads for Solaris and AIX are now by request. To obtain Solaris or AIX product downloads, please contact Klocwork Customer Support.

Solaris installation packages, build tools, and desktop tools are available for Klocwork release 2020.1. Downloads are not available for later releases.

Licensing

2019 licenses are not compatible with Klocwork 2020.4. You need a new license to use the latest version of the product. Contact license@perforce.com to obtain a new license.

We upgraded the version of FlexNet Publisher that we support for Windows and Linux to version 2018 R4 (11.16.2). If you are using your own FlexNet Publisher license server, ensure you upgrade to this or a newer version. You can also use the license server included with Klocwork 2020.1 and beyond.

Maintenance for Klocwork 2018 ended

Maintenance for all versions of Klocwork 2018 ended February 29, 2020. The end of maintenance (EOM) date and end of sale (EOS) date was also February 29, 2020. For information about the availability of support for any release of Klocwork, see the Klocwork Product Lifecycle.

Portal licensing changes

Klocwork has implemented additional licensing checks related to running the Klocwork Server, which, among other things, underpins the Klocwork portal. We recommend you validate your licensing needs to ensure you have a sufficient number of web service licenses.

Changes to system requirements

This section lists changes to the system requirements. For the complete list of supported versions, see System requirements. We've added support for the following:

- Windows 10 versions above 1909 to 2004
- Debian 10.4
- Red Hat Enterprise Linux versions 7.8 and 8.2
- Oracle Linux versions 7.8 and 8.2
- CentOS versions 7.8 and 8.2
- Ubuntu 20.04 LTS
- Fedora 32
- OpenSUSE Leap 15.2
- Eclipse version 4.16
- Android Studio versions above 3.6 to 4.0
- Visual Studio 2017 versions up to 15.9.24
- Visual Studio 2019 versions up to 16.6.3
- IntelliJ IDEA 2019 versions above 19.3.3 to 19.3.5
- IntelliJ IDEA 2020.1 versions up to 2020.1.2
- Wind River Workbench 4 SR0640
- CLion 2020.1 version 2020.1.2
- Internet Explorer versions above 11.0.175 to 11.0.195
- Edge versions 81.x and 83.x
- Firefox versions 78.x
- Chrome versions above 80.x to 83.x
- Jenkins versions above 2.204.5 to 2.243
- Gradle versions above 6.2.1 to 6.5.1

We no longer provide support for the following:

- Windows 10, version 1803
- Ubuntu version 19.10
- Fedora 30
- SUSE Enterprise 12 SP1
- Eclipse versions 3.4 to 3.7

What's new in Klocwork 2020.2

Here are the highlights for Klocwork 2020.2. If you're upgrading, also see the Limitations on page 35 for items that affect how you use Klocwork.

New Jenkins plugin

Our new Jenkins plugin provides an easy way for you to automate industry-leading static code analysis as part of your Continuous Integration (CI) or Continuous Delivery (CD) pipeline.

To support the DevOps and CI/CD movement, which requires speed from a static analysis tool, our plugin provides Klocwork's Differential Analysis, which uses system context data from the server to analyze only the files that were changed, while providing a diff analysis as if the entire system were analyzed, resulting in the shortest analysis times. You can also use it to generate periodic full analysis runs.

For more information, see Klocwork Jenkins CI plugin.

CLion plugin

Use our new CLion desktop analysis plugin to quickly and easily detect and fix issues before check-in. For more information, see Getting started with Klocwork Desktop plugin for CLion.

PCI DSS version 3.2.1 taxonomy

We now provide taxonomies that map Payment Card Industry Data Security Standard (PCI DSS) Version 3.2.1 IDs to Klocwork checkers for C and C++, C#, and Java. The PCI DSS was developed to encourage and enhance cardholder data security and facilitate the broad adoption of consistent data security measures globally. For more information, see Payment Card Industry Data Security Standard IDs mapped to Klocwork checkers.

Joint Strike Fighter Air Vehicle C++ taxonomy

Our new Joint Strike Fighter Air Vehicle C++ taxonomy maps the Joint Strike Fighter Air Vehicle C++ coding standard to Klocwork C++ checkers. The Joint Strike Fighter Air Vehicle C++ coding standard, developed by Lockheed Martin, helps programmers develop error-free code for safety-critical systems. For more information, see Joint Strike Fighter Air Vehicle C++ IDs mapped to Klocwork C++ checkers.

Dramatic improvements to C#

We've dramatically improved support for C# in this release:

- We've significantly improved C# build integration using kwinject.
- We now support mixed C/C++ and C# projects.
- We've also added support for more C# VS project types like .Net.

On some Open Source projects that we benchmark against, we've seen up to a 30% increase in defects detected!

We also now have full support for the C# 7.0 language specification. For example, we've added support for the following language features:

- out variables as function arguments and discard out variables
- pattern matching
- tuples, tuple deconstruction, and discards in tuple deconstruction
- local functions
- binary literals and digit separators
- ref locals and returns
- generalized async return types
- expression bodied members for members formally returning void
- throw expressions

Significant improvements to Java

We're pleased to announce that we now have full support for the Java 9 language specification. For example, we've added or improved support for the following:

- Java Platform Module System
- private methods in interfaces
- diamond operator for anonymous inner class
- @SafeVarargs on private instance methods
- Try-with-resources Java 9 enhancement

We've also made significant improvements to our Java analysis engine with regard to lambda functions. For example:

- Lambda return types are now properly analyzed and determined, which allows for better nesting and chaining of lambdas in objects such as java.util.Stream.
- We've improved the type inference of input values such that we understand both lower and upper bounds and perform parameter coercion correctly.
- Method references now properly identify static and non-static methods and match overloaded methods correctly.

Improved Knowledge bases

We've improved our KB related to virtual methods. For more information, see C/C++ knowledge base reference.

Analysis improvements

We now support cases of intraprocedural function pointer resolution in defect detection as well as cases of function pointers that are returned directly or indirectly by function calls (limited interprocedural support). We've also improved support for rvalue references and for override file mechanisms.

Simplified plugin installation

We've simplified the installation of our IntelliJ IDEA and Android Studio plugins by converting our installers into JetBrains-style plugins. For more information, see Installing the IntelliJ IDEA/Android Studio/CLion plugins.

Klocwork checker improvements

From release to release, we improve issue detection to bring state-of-the-art capabilities to our customers. As a result, expect your analysis results to change as accuracy and coverage improve.

New Klocwork checkers

Checker	Description
CS.SV.TAINTED.ALLOC_SIZE	C# checker that flags code that uses tainted data in determining the size of a memory allocation.
CS.SV.TAINTED.BINOP	C# checker that flags code that uses tainted data in arithmetic binary operations, such as addition, subtraction, or multiplication.
CS.SV.TAINTED.CALL.BINOP	C# checker that flags code that uses tainted data, via a function call, in arithmetic binary operations, such as addition, subtraction, or multiplication.
CS.SV.TAINTED.CALL.INDEX_ACCESS	C# checker that flags code that passes tainted data to functions that will use it to access an array.
CS.SV.TAINTED.CALL.LOOP_BOUND	C# checker that flags code when a loop variable is passed as an argument to another function and used as a loop boundary.
CS.SV.TAINTED.INDEX_ACCESS	C# checker that flags code that uses tainted data to access an array.

Checker	Description
CS.SV.TAINTED.LOOP_BOUND	C# checker that flags code when an unvalidated argument is used as a loop boundary.
JD.CAST.SUSP.MIGHT	Related to Java checker JD.CAST.SUSP.MUST. This checker flags code when an object is checked with an instance of operator for type A and then cast to type B, where types A and B may be unrelated.
JD.CAST.SUSP.MUST	Renamed Java checker JD.CAST.SUSP. This checker flags code when an object is checked with an instance of operator for type A and then cast to type B, where types A and B are unrelated.
MISRA.PTR.ARITH.NOT_SAME.2008	Supports MISRA-C++ Rule 5–0–16: A pointer operand and any pointer resulting from pointer arithmetic using that operand shall both address elements of the same array.
MISRA.TYPE.NAMECLASH.C.2004	Deprecated the checker MISRA.TYPE.NAMECLASH and split it into two checkers. This checker supports MISRA-C Rule 5.6 (advisory): No identifier in one name space should have the same spelling as an identifier in another name space, with the exception of structure member and union member names.
MISRA.TYPE.NAMECLASH.CPP.2008	Deprecated the checker MISRA.TYPE.NAMECLASH and split it into two checkers. This checker supports MISRA-C++ Rule 2-10-6 (required): If an identifier refers to a type, it shall not also refer to an object or a function in the same scope.
SV.TAINTED.CALL.GLOBAL	This C/C++ checker flags code whenever tainted data is used to assign a global variable via a function call.
SV.TAINTED.GLOBAL	This C/C++ checker flags code whenever tainted integer data is used to initialize the global variable.

New community checkers

Checker	Description
CXX.BSTR.LITERAL	Do not pass string literal or casted CString to COM function expecting BSTR parameter.
CXX.CWARN.DTOR.NONVIRT	Destructors should be declared as virtual.
CXX.CWARN.HARDCODED_LOOP_BOUND	Hard-coded loop used for array index.
CXX.CWINAPP.INIT	Incorrect or missing InitInstance override for class derived from CWinApp.
CXX.FUNC.CSTRING.FORMAT	CString cannot call CString.Format() on itself.

Checker	Description
CXX.FUNC.MEMSET.BUILTIN	Calls to memset must not pass a reference to a structure containing non-builtin types.

Modified Klocwork checkers

Checker	Description
DBZ.CONST	New defects detected
CS.HIDDEN.MEMBER.LOCAL.CLASS	New defects detected
JD.CAST.KEY	Fewer false positives are expected
MISRA.CTOR.BASE	Fewer false positives are expected
MISRA.FLOAT_EQUAL	New defects detected.
MISRA.FUNC_CAST	New defects detected
MISRA.FLOAT_EQUAL	New defects detected
MISRA.LITERAL.NULL.PTR.CONST.2012	Fewer false positives are expected
MISRA.ONEDEFRULE.VAR	New defects detected
MISRA.PTR.ARITH	Fewer false positives are expected
MISRA.VAR.HIDDEN	New defects detected, and fewer false positives are expected.
RH.LEAK	Fewer false positives are expected
RTC.CALL	Fewer false positives are expected
SV.TAINTED.BINOP	New defects detected
UNINIT.STACK.MIGHT	Fewer false positives are expected

Enabled or disabled checkers

The following checkers were added to the default `enabled` field of the checker configuration files for this release:

- CS.SV.TAINTED.ALLOC_SIZE
- CS.SV.TAINTED.CALL.INDEX_ACCESS
- CS.SV.TAINTED.CALL.LOOP_BOUND
- CS.SV.TAINTED.INDEX_ACCESS
- CS.SV.TAINTED.LOOP_BOUND
- JD.CAST.SUSP.MIGHT
- JD.CAST.SUSP.MUST

Taxonomy improvements

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

Taxonomy	New/Updated
autosar_cpp_17_10.tconf and autosar_cpp_17_10_ja.tconf	Updated rule 6.2 to reference MISRA.TYPE.NAMECLASH.CPP.2008 instead of MISRA.TYPE.NAMECLASH.

Taxonomy	New/Updated
autosar_cpp_18_03.tconf and autosar_cpp_18_03_ja.tconf	Updated rule 6.2 to reference MISRA.TYPE.NAMECLASH.CPP.2008 instead of MISRA.TYPE.NAMECLASH.
• cert_c.tconf and cert_c_ja.tconf • cert_cpp.tconf and cert_cpp_ja.tconf	These are new taxonomies that map Klocwork checkers to CERT C and CERT C++ IDs, respectively. We split the previous cert_c_cpp.tconf and cert_c_cpp_ja.tconf taxonomies into separate C and C++ taxonomies.
• cert_c_community.tconf and cert_c_community_ja.tconf • cert_cpp_community.tconf and cert_cpp_community_ja.tconf	These are new taxonomies that map both Klocwork and community checkers to CERT C and CERT C++ IDs, respectively. We split the previous cert_c_cpp_community.tconf and cert_c_cpp_community_ja.tconf into separate C and C++ taxonomies and added the Klocwork checkers.
• cwe_2019_top_25_cs.tconf and cwe_2019_top_25_cs_ja.tconf • cwe_all_cs.tconf and cwe_all_cs_ja.tconf	Removed the reference to rule CWE-400 for CS.SQL.INJECT.LOCAL. We added references to the following checkers: CWE-20: Improper Input Validation • CS.SV.TAINTED.ALLOC_SIZE • CS.SV.TAINTED.CALL.INDEX_ACCESS • CS.SV.TAINTED.CALL.LOOP_BOUND • CS.SV.TAINTED.INDEX_ACCESS • CS.SV.TAINTED.LOOP_BOUND CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer • CS.SV.TAINTED.INDEX_ACCESS • CS.SV.TAINTED.CALL.INDEX_ACCESS CWE-190: Integer Overflow or Wraparound • CS.SV.TAINTED.BINOP • CS.SV.TAINTED.CALL.BINOP
jsf_av_rev_c_community_cpp.tconf and jsf_av_rev_c_community_cpp_ja.tconf	New community taxonomy that maps the Joint Strike Fighter Air Vehicle C++ coding standard to Klocwork C++ checkers.
kw_quality_std_java.tconf and kw_quality_std_java_ja.tconf	Removed the reference to JD.CAST.SUSP and added references to JD.CAST.SUSP.MIGHT and JD.CAST.SUSP.MUST.
misra_c_2004.tconf and misra_c_2004_ja.tconf	Removed the reference to MISRA.TYPE.NAMECLASH and added a reference to MISRA.TYPE.NAMECLASH.C.2004.
misra_cpp_2008.tconf and misra_cpp_2008_ja.tconf	• Removed the reference to MISRA.TYPE.NAMECLASH and added a reference to MISRA.TYPE.NAMECLASH.CPP.2008.

Taxonomy	New/Updated
	• Added a reference to MISRA.PTR.ARITH.NOT_SAME.2008 for rule 5-0-16.
pci_3_2_1_community_cs.tconf and pci_3_2_1_community_cs_ja.tconf	New taxonomy that maps the Payment Card Industry Data Security Standard (PCI DSS) Version 3.2.1 IDs to Klocwork C# checkers.
pci_3_2_1_community_cxx.tconf and pci_3_2_1_community_cxx_ja.tconf	New taxonomy that maps the Payment Card Industry Data Security Standard (PCI DSS) Version 3.2.1 IDs to Klocwork C and C++ checkers.
pci_3_2_1_community_java.tconf and pci_3_2_1_community_java_ja.tconf	New taxonomy that maps the Payment Card Industry Data Security Standard (PCI DSS) Version 3.2.1 IDs to Klocwork Java checkers.
quality_community_cxx.tconf and quality_community_cxx_ja.tconf	Added references to the following checkers: • CXX.FUNC.CSTRING.FORMAT • CXX.CWARN.HARDCODED_LOOP_BOUND • CXX.BSTR.LITERAL • CXX.CWARN.DTOR.NONVIRT • CXX.FUNC.MEMSET.BUILTIN • CXX.CWINAPP.INIT

Improvements to supported compilers

We've improved support for the following compilers:

- Clang
- GNU
- IAR compiler/linker for STM8 Microcontroller family
- Renesas SuperH and RX family
- Wind River Diab

For the full list of supported C/C++ compilers, see C/C++ compilers supported for build integration.

Solaris and AIX support

Downloads for Solaris and AIX are now by request. To obtain Solaris or AIX product downloads, please contact Klocwork Customer Support.

Solaris installation packages, build tools, and desktop tools are available for Klocwork release 2020.1. Downloads are not available for later releases.

Licensing

2019 licenses are not compatible with Klocwork 2020.4. You need a new license to use the latest version of the product. Contact license@perforce.com to obtain a new license.

We upgraded the version of FlexNet Publisher that we support for Windows and Linux to version 2018 R4 (11.16.2). If you are using your own FlexNet Publisher license server, ensure you upgrade to this or a newer version. You can also use the license server included with Klocwork 2020.1 and beyond.

Maintenance for Klocwork 2018 ended

Maintenance for all versions of Klocwork 2018 ended February 29, 2020. The end of maintenance (EOM) date and end of sale (EOS) date was also February 29, 2020. For information about the availability of support for any release of Klocwork, see the Klocwork Product Lifecycle.

Portal licensing changes

Klocwork has implemented additional licensing checks related to running the Klocwork Server, which, among other things, underpins the Klocwork portal. We recommend you validate your licensing needs to ensure you have a sufficient number of web service licenses.

Changes to system requirements

This section lists changes to the system requirements. For the complete list of supported versions, see System requirements. We've added support for the following:

- Windows 10 versions beyond 1903 to 1909
- Debian 9.12 and 10.3
- Red Hat Enterprise Linux 8.1
- Oracle Linux versions beyond 7 to 7.7 and 8 to 8.1
- CentOS versions beyond 8.0 to 8.1
- Ubuntu 18.04.4 LTS
- SUSE Enterprise 12 SP5
- AIX 7.2 (TL4)
- Eclipse versions beyond 4.13 to 4.15
- Android Studio versions beyond 3.5.1 to 3.6
- Visual Studio 2017 versions beyond 15.9.17 to 15.9.20
- Visual Studio 2019 versions beyond 16.3.6 to 16.4.5
- IntelliJ IDEA 2019.2 versions beyond 2019.2.3 to 2019.2.4
- IntelliJ IDEA 2019.3 up to 2019.3.3
- Wind River Workbench 4 SR0630
- Microsoft Internet Explorer versions beyond 11.0.155 to 11.0.175
- Microsoft Edge 79.x, 80.x to 80.0.361
- Mozilla Firefox 73.x
- Chrome versions beyond 78.x to 80.x
- Safari versions beyond 12.1.2 to 13.0
- Jenkins versions 1.658 to 2.204.5
- Gradle versions beyond 5.6.3 to 6.2.1
- CLion versions 2019.2 (up to 2019.2.5), 2019.3 (up to 2019.3.6), and 2020.1 (up to 2020.1.1).

We no longer provide support for the following:

- Fedora 29
- IntelliJ IDEA versions earlier than 12.x
- Android Studio versions earlier than 2.3.2
- Eclipse versions earlier than 4.2

Changes to commands, tools, and options

We've improved support for kwinject so that it is the command of choice for C#.

For more information about Klocwork commands, see Command Reference.

What's new in Klocwork 2020.1

Here are the highlights for Klocwork 2020.1. If you're upgrading, also see the Limitations on page 35 for items that affect how you use Klocwork.

Community checkers and taxonomies

We've leveraged our deep ties with the wider Klocwork community to bring you a series of new taxonomies and community checkers. In doing so, we've added close to 200 community checkers to Klocwork across our supported languages: C/C++, C#, and Java. For more information, see the following:

- AUTOSAR community checker reference

- CERT community C and C++ checker reference
- Klocwork Quality Standard community C and C++ checker reference
- MISRA-C 2012 community checker reference
- Klocwork Quality Standard community C# checker reference
- Klocwork Quality Standard community Java checker reference
- CERT Java IDs mapped to Klocwork Java checkers

HIS metrics configuration file

The 'his_metrics_community.mconf' file may be of interest to you if your projects focus on automotive industry standards. HIS (Hersteller Initiative Software) was created from five working groups in the automotive industry whose goal is the production of agreed standards. For more information, see Adding the community HIS Metrics configuration file.

Performance improvements

We've upgraded several components in our toolchain to leverage 64-bit architecture, so Klocwork can more effectively analyze large, complex code bases, and projects.

Analysis engine enhancement

We've updated our analysis engine for accuracy and defect detection related to nested namespaces, references, and templates. On some Open Source projects that we benchmark against we've seen up to a 28% increase in defects detected.

Improved Knowledge Bases

We've overhauled the Knowledge Bases we include with Klocwork. You'll see better analysis accuracy in code using the standard C++ library related to smart pointers, utilities, and concurrency, to name a few.

MISRA checkers and taxonomies now fully integrated

In this release, we've integrated all of the MISRA taxonomies and checkers into Klocwork by default, so you no longer need to install and deploy MISRA checker packages separately. Configuring your project to validate against a MISRA taxonomy is now as easy as adding any of our other industry-specific taxonomies. For more information, see Configuring industry-specific coding standards and checkers.

Klocwork checker improvements

From release to release, we improve issue detection to bring state-of-the-art capabilities to our customers. As a result, expect your analysis results to change as accuracy and coverage improve.

New Klocwork checkers

Checker	Description
CWARN.MOVE.CONST	Detects and reports instances of calls to the std::move() method from the C++ Standard Template Libraries (STL) where the argument is const.

Modified Klocwork checkers

Checker	Description
INVARIANT_CONDITION.GEN	Fewer false positives are expected.
MISRA.LITERAL.NULL.INT	New defects detected.
MISRA.MEMB.NON_CONST	Fewer false positives are expected.
MISRA.VAR.MIN.VIS	Fewer false positives are expected.
MISRA.VAR.NEEDS.CONST	Fewer false positives are expected.
MLK.MUST	Fewer false positives are expected.

Checker	Description
RH.LEAK	Fewer false positives are expected.
STRONG.TYPE.JOIN.CONST	New defects detected.
STRONG.TYPE.JOIN.ZERO	New defects detected.
UNINIT.CTOR.MUST	Fewer false positives are expected.
UNUSED.FUNC.STL_EMPTY	Fewer false positives are expected.

Enabled or disabled checkers

The following checkers were added to the default `enabled` field of the checker configuration files for this release.

- CWARN.MOVE.CONST

Taxonomy improvements

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

Taxonomy	New/Updated
autosar_community_cpp14_19_03 and autosar_community_cpp14_19_03_ja	New taxonomy of the list of Klocwork community C/C++ checkers that map to the secure coding standard defined by the Automotive Open System Architecture (AUTOSAR), release 19-03.
cert_c_cpp.tconf and cert_c_cpp_ja.tconf	Added references to the following rules: • DCL50-CPP • MEM51-CPP
cert_c_cpp_community.tconf and cert_c_cpp_community_ja.tconf	New taxonomy of the list of Klocwork community C/C++ checkers that map to the secure coding standard defined by the computer emergency response team (CERT).
cert_java_community.tconf and cert_java_community_ja.tconf	New community-developed taxonomy of the list of Klocwork Java checkers that map to the secure coding standard defined by the computer emergency response team (CERT).
cwe_10_cxx.tconf and cwe_10_cxx_ja.tconf were renamed to cwe_all_cxx.tconf and cwe_all_cxx_ja.tconf.	Taxonomy file renamed. Added references to the following rules: • CWE-94 • CWE-125 • CWE-259 • CWE-269 • CWE-400 • CWE-426 • CWE-798 Removed references to the following rules: • CWE-195 • CWE-247 • CWE-366

Taxonomy	New/Updated
	• CWE-466 • CWE-479 • CWE-488 • CWE-562 • CWE-587 • CWE-684 • CWE-764 • CWE-770
cwe_10_java.tconf and cwe_10_java_ja.tconf were renamed to cwe_all_java.tconf and cwe_all_java_ja.tconf.	List of Klocwork Java checkers that map to the Common Weakness Enumeration (CWE) types. Added references to the following rules: • CWE-200 • CWE-259 • CWE-295 • CWE-426 • CWE-772
cwe_cs.tconf and cwe_cs_ja.tconf were renamed to cwe_all_cs.tconf and cwe_all_cs_ja.tconf.	Taxonomy file renamed. Added references to CWE-20 and CWE-94. Removed references to CWE-248, CWE-571, and CWE-783.
cwe_2019_top_25_cs.tconf and cwe_2019_top_25_cs_ja.tconf	New taxonomy of the list of Klocwork C# checkers that map to the 2019 CWE top 25 most dangerous software errors.
cwe_2019_top_25_cxx.tconf and cwe_2019_top_25_cxx_ja.tconf.	New taxonomy of the list of Klocwork C/C++ checkers that map to the 2019 CWE top 25 most dangerous software errors.
cwe_2019_top_25_java.tconf and cwe_2019_top_25_java_ja.tconf	New taxonomy of the list of Klocwork Java checkers that map to the 2019 CWE top 25 most dangerous software errors.
cwe_25_cxx.tconf and cwe_25_cxx_ja.tconf were renamed to cwe_2011_top_25_cxx.tconf and cwe_2011_top_25_cxx_ja.tconf.	Taxonomy file renamed.
cwe_25_java.tconf and cwe_25_java_ja.tconf were renamed to cwe_2011_top_25_java.tconf and cwe_2011_top_25_java_ja.tconf.	Taxonomy file renamed.
misra_c_2012_community.tconf and misra_c_2012_community_ja.tconf	New taxonomy of the list of Klocwork community checkers that map to the MISRA-C 2012 standard.
quality_community_cxx.tconf and quality_community_cxx_ja.tconf	New taxonomy of the list of Klocwork community C/C++ checkers that focus on improving overall code quality.
quality_community_cs.tconf and quality_community_cs_ja.tconf	New taxonomy of the list of Klocwork Community C# checkers that focus on improving overall code quality.

Taxonomy	New/Updated
quality_community_java and quality_community_java_ja	New taxonomy of the list of Klocwork community Java checkers that focus on improving overall code quality.

Improvements to supported compilers

We've improved support for the following compilers:

- Clang
- GNU
- Microchip MPLAB pic32
- Tasking Tricore

For the full list of supported C/C++ compilers, see C/C++ compilers supported for build integration.

Klocwork release numbering

We've updated our release numbering strategy. Going forward, the first release each year will have the year as the major release number and 1 as the minor release number, for example, 2020.1. Subsequent planned releases will increment the minor number, for example, 2020.2, 2020.3, and 2020.4.

Licensing

2019 licenses are not compatible with Klocwork 2020.4. You need a new license to use the latest version of the product. Contact license@perforce.com to obtain a new license.

We upgraded the version of FlexNet Publisher that we support for Windows and Linux to version 2018 R4 (11.16.2). If you are using your own FlexNet Publisher license server, ensure you upgrade to this or a newer version. You can also use the license server included with Klocwork 2020.1.

Checker support levels

With the introduction of community checkers, we've developed a set of support levels that explain the types of support we offer for the checkers we provide. For more information, see checker support levels.

End of support announcements

As of this release we have ended support for the Microsoft Visual Studio addin. Our Microsoft Visual Studio Extension (kw-vsplugin.vsix) contains the complete feature set and supports Visual Studio versions 2012 to 2019.

Klocwork 2019.3 was the last supported release of the Vim plug-in.

Maintenance for Klocwork 2018 ending

Maintenance for all versions of Klocwork 2018 is ending: the end of maintenance (EOM) date is February 29, 2020; the end of sale (EOS) date is also February 29, 2020. For information about the availability of support for any release of Klocwork, see the Klocwork Product Lifecycle.

Developer Network

In October, 2018, our technical Support Center at https://techsupport.roguewave.com/ was upgraded to include Klocwork. As part of that transition, Developer Network is no longer available.

Portal licensing changes

Klocwork has implemented additional licensing checks related to running the Klocwork Server, which, among other things, underpins the Klocwork portal. We recommend you validate your licensing needs to ensure you have a sufficient number of web service licenses.

Changes to system requirements

This section lists changes to the System requirements. We've added support for the following:

- Debian 9.11 and 10.1

- Red Hat Enterprise Linux 7.7
- CentOS 7.7 and 8.0
- Ubuntu 18.04.3 LTS and 19.10
- Oracle Linux 7
- macOS 10.14.6
- Eclipse 4.13
- Android Studio 3.5.1
- Visual Studio 2017 up to 15.9.17
- Visual Studio 2019 up to 16.3.6
- IntelliJ IDEA 2019.4 and 2019.2.3
- QNX Momentics 6.3 SP3
- Wind River Workbench 4 SR0620 Edition 2
- Internet Explorer 11.0.155
- Microsoft Edge 44.18362
- Firefox 68.2.x and 70.x
- Safari 12.1.2
- glibc 2.30
- Gradle 3.x to 5.6.3

We no longer support the following:

- Debian 8.x to 8.11
- Windows 7 SP1
- Windows Server 2008
- Windows 10, version 1709
- Ubuntu 18.10 and 19.04
- Fedora 27 to 28
- openSUSE Leap 15
- SUSE Enterprise 12
- AIX 7.1 TL4 and 7.2 TL1
- macOS 10.12.x
- Visual Studio 2010
- Jenkins continuous integration plug-in
- TeamCity continuous integration plug-in

Changes to commands, tools, and options

We no longer support the kwconan command, or the Jenkins or TeamCity plug-ins for continuous integration. We recommend you use the kwciagent command and our concurrent licensing model instead. We're here to help! If you need assistance making this change, you can contact Static Code Analysis Professional Services to discuss assistance via a services engagement.

For more information about Klocwork commands, see Command Reference.

Fixed issues in Klocwork 2020.4

The following issues were fixed in Klocwork 2020.4.

General issues

Number	Description
SUPPORT-26525	Fixed an intermittent issue with the C/C++ analysis engine on the Windows platform.
SUPPORT-34881	Fixed an issue with the Web API related to inconsistent results with a search query.
SUPPORT-36701	Updated the Klocwork Jenkins plug-in to eliminate a vulnerability.

Number	Description
SUPPORT-35075	Fixed a packaging issue related to a missing .jar file.
SUPPORT-36178	Improved the performance of the Visual Studio extension.
SUPPORT-30315	Improved support for the Clang and GNU compilers.
SUPPORT-36470	Improved analysis performance times related to a specific customer toolchain.
SUPPORT-35071	Modified the build.log files to include the names of any files that produce parse errors when compliance mode is enabled.
SUPPORT-35893	Improved support for the Intel C++ compiler.
SUPPORT-34256	Added a new AUTOSAR C++ taxonomy that maps Klocwork and community checkers to the 18-10 revision.
SUPPORT-31664	Added support for the TI C7000 Optimizing C/C++ compiler.
SUPPORT-32500	Improved support for Android Q.
SUPPORT-35314	Added an option to kwbuildproject to force the analysis engine to analyze C++/CLI source files containing Managed C++ code.
SUPPORT-38022	Fixed an issue with the Visual Studio extension related to issues appearing outside of a taxonomy.
SUPPORT-31063	Fixed an issue with the Visual Studio extension related to issues in system headers.
SUPPORT-6557, SUPPORT-33867, SUPPORT-35034	Improved support for the ARM compiler.
SUPPORT-24968, SUPPORT-31801, SUPPORT-29307, SUPPORT-35201	Improved support for the Clang compiler.
SUPPORT-36635	Improved support for the Wind River Diab compiler.
SUPPORT-38021	Improved support for the Visual Studio extension related to how the plug-in calculates issues when grouping by taxonomy.
SUPPORT-21731, SUPPORT-36837, SUPPORT-34711	Improved support for the Microsoft Visual C++ compiler.
SUPPORT-34918	Fixed an upgrade issue related to customers accidentally overwriting their misra.xml file when running the kwdeploy command.
SUPPORT-35608	Updated checker mappings for the ISO IEC TS 17961 taxonomy.
SUPPORT-35243	Added a check box configuration option in the Eclipse plugin and Klocwork Desktop to automatically set Klocwork to use a loopback interface when running the kwcheck command.

Checker issues

Number	Description
SUPPORT-33818	Reduced false positives with the checker MISRA.ARRAY.VAR_LENGTH.2012.
SUPPORT-22684	Reduced false positives with the checker MISRA.IF.NO_ELSE.
SUPPORT-33126	Reduced false positives with the checker MISRA.ETYPE.ASSIGN.2012.

Number	Description
SUPPORT-26238, SUPPORT-36708, SUPPORT-6934, SUPPORT-36480	Reduced false positives with the checker MISRA.FUNC.DECL.AFTERUSE.
SUPPORT-27328	Improved defect detection for the checker CS.RLK.
SUPPORT-8421	Improved defect detection for the AVB and SV.TAINT checkers.
SUPPORT-35903	Improved defect detection for the checker CS.NRE.FUNC.MUST.
SUPPORT-35821	Reduced false positives with the checker MISRA.TOKEN.OCTAL.INT.

Documentation issues

Number	Description
SUPPORT-31859	Added information about how to use Klocwork in containers.
SUPPORT-37836	Removed a limitation for the checker DBZ.CONST.
SUPPORT-37750, SUPPORT-27994	Corrected the supported versions of Visual Studio in the Klocwork 2019 documentation.
SUPPORT-35089	Added information about build tools dependencies for Windows.
SUPPORT-31760	Corrected references to the BuildTools package in the list of server arguments.

Fixed issues in Klocwork 2020.3

The following issues were fixed in Klocwork 2020.3.

General issues

Number	Description
00038953, SUPPORT-9223	Improved support for C++ 11 for loops.
SUPPORT-33483	Fixed an issue related to garbled messages related to skipped functions during analysis.
SUPPORT-21731, SUPPORT-33498	Improved support for Microsoft Enterprise Windows Development Kits.
SUPPORT-25342, SUPPORT-32777	Improved support for HIS metrics.
SUPPORT-28138	Fixed an issue with analysis related to the _Alignof keyword.
SUPPORT-29985	Improved support for 64-bit analysis on Windows.
SUPPORT-30661	Fixed an issue related to user authentication.
SUPPORT-30449	Fixed an issue related to a garbled message from the kwcheck command.
SUPPORT-32995	Added an parameter to dbvalidate script to rebuild the Lucene index.
SUPPORT-32705	Resolved Java analysis errors and warnings for a project.
SUPPORT-33265	Improved support for SUSE Enterprise Server 12.
SUPPORT-32717	Improved support for the Clang compiler.
SUPPORT-33213	Fixed an issue related to results returned when searching for a specific project owner.

Number	Description
SUPPORT-8027	Fixed an issue related to Klocwork Server performance on permission lookup operations.
SUPPORT-34123	Improved support for the GNU filter.
SUPPORT-35506	Fixed an issue with the Visual Studio plugin related to having to press the refresh button to populate the Klocwork server project list.
SUPPORT-35213	Fixed an issue related to the locale setting when importing projects.

Checker issues

Number	Description
00027316	Split the checker JD.CAST.COL into two checkers: JD.CAST.COL.MIGHT and JD.CAST.COL.MUST. Deprecated JD.CAST.COL.
00030478	Reduced false positives with the checker SV.EXPOSE.MUTABLEFIELD.
00034224, SUPPORT-6901	Reduced false positives with the checker CWARN.BITOP.SIZE.
00036328, SUPPORT-32702	Reduced false positives with the checker MISRA.COMP.WRAPAROUND.
SUPPORT-11013	Enabled customers to reduce false positives for the checker MISRA.FUNC.NOPROT.CALL by allowing it to be parametrized via configuration.
SUPPORT-26217	Reduced false positives with the checker STRONG.TYPE.ASSIGN.RETURN.
SUPPORT-27190	Reduced false positives with the checker CWARN.BITOP.SIZE.
SUPPORT-28226	Reduced false positives with the checker LV_UNUSED.GEN.
SUPPORT-30697, SUPPORT-34830	Improved defect detection for the checker NPD.FUNC.MUST.
SUPPORT-34845	Improved defect detection for the checker MISRA.ETYPE.ASSIGN.2012.
SUPPORT-34844	Improved defect detection for the checker MISRA.COMP.WRAPAROUND.
SUPPORT-34876	Improved defect detection for the checker MISRA.CVALUE.IMPL.CAST.
SUPPORT-35384	Reduced false positives with the checker STRONG.TYPE.ASSIGN.INIT.
SUPPORT-26319	Reduced false positives with the checker UNINIT.STACK.MUST.

Documentation issues

Number	Description
SUPPORT-35166	Updated the Japanese translation for the section about installing the IntelliJ IDEA, Android Studio, CLion plugins.

Fixed issues in Klocwork 2020.2

The following issues were fixed in Klocwork 2020.2.

General issues

Number	Description
SUPPORT-29567, SUPPORT-29307,	Improved support for the Clang compiler.

Number	Description
SUPPORT-31801, SUPPORT-23319	
SUPPORT-9013, 00034679, SUPPORT-6941, 00029056, 00036343	Improved support for Java.
SUPPORT-6842, SUPPORT-6650, 00030828	Fixed an LDAP issue related to group names that contain a forward slash.
SUPPORT-26265, SUPPORT-24623, SUPPORT-7715, SUPPORT-32881	Improved support for C# 7.0.
SUPPORT-31618	Improved support for the Jenkins plugin.
SUPPORT-28427	Provided a workaround related to slow LDAP searches.
SUPPORT-29897	Improved support for the Renesas SuperH and RX family compiler.
SUPPORT-30787	Fixed an issue related to cleaning up database entries.
SUPPORT-25350, SUPPORT-21734	Improved support for C# that allows larger projects to be analyzed.
SUPPORT-22629	Fixed an issue related to the way the Web API searches LDAP and Klocwork groups.
SUPPORT-6992, 00035126	Fixed an issue related to projects_root permissions.
SUPPORT-21998, SUPPORT-23636, SUPPORT-8364, SUPPORT-17320, SUPPORT-22549, SUPPORT-17252, SUPPORT-32443, SUPPORT-32476, SUPPORT-32069, SUPPORT-23973, SUPPORT-20980, 510085, 00037925	Improved support for C# analysis.
SUPPORT-23501	Fixed an issue related to changing the color of the Issue report.
SUPPORT-7113	Fixed an issue related to how Klocwork matches detected issues across projects in an issue matching group.
SUPPORT-25342	Fixed an issue related to reporting metrics for a child if a parent entity is not present.
SUPPORT-33265	Improved support for SUSE Enterprise Linux 12.
SUPPORT-26525	Fixed an issue related to the C++ analysis.
SUPPORT-9402	Fixed an issue related to support for SNI host verification.
SUPPORT-21150, SUPPORT-10852, 00038772	Improved support for Java 9.

Number	Description
SUPPORT-29569. SUPPORT-29570	Fixed an issue related to the analysis of cast expressions.
SUPPORT-33502	Fixed an issue with Visual Studio related to icons being displayed at the bottom of the window.
SUPPORT-27481, SUPPORT-32091	Improved support for the Wind River Diab compiler.
SUPPORT-33512	Fixed a website issue related to an expired certificate.
SUPPORT-30427	Fixed an issue with importing a project from Klocwork 2018 to Klocwork 2019.
SUPPORT-32389, SUPPORT-32762, SUPPORT-31956	Improved support for the GNU compiler.
SUPPORT-32439	Fixed an issue with analysis related to unrolling loops.
SUPPORT-29371	Fixed an issue with Visual Studio related to system headers.
SUPPORT-21768	Adjusted which issues in C# result in a non-zero exit so that there is consistency with C/C++ non-zero exit codes.
SUPPORT-9864	Added the ability to create a custom knowledge base for a virtual method.
SUPPORT-30458	Fixed an issue related to the time it took to run the defect groups calculation for a project.
SUPPORT-6936, SUPPORT-7031, 00035497, 00035100	Improved Java analysis related to Lambda expressions.
SUPPORT-28500	Improved support for the IAR compiler/linker for the STM8 Microcontroller family.
SUPPORT-30475	Improved the security of the Windows installer.
SUPPORT-27329	Fixed an issue with the Visual Studio plugin related to filtering issues based on a custom taxonomy.
SUPPORT-32784	Fixed an issue with the Visual Studio plugin related to detecting the compiler.
00022093	Added a community taxonomy that maps the Joint Strike Fighter Air Vehicle C++ coding standard to Klocwork C++ checkers.

Checker issues

Number	Description
SUPPORT-29013	Reduced false positives with the checker MISRA.PTR.ARITH.
SUPPORT-26233	Reduced false positives with the checker MISRA.CTOR.BASE.
SUPPORT-7001, SUPPORT-28448	Improved defect detection with the checker MISRA.FLOAT_EQUAL.
SUPPORT-29541, SUPPORT-29529 , SUPPORT-7683, SUPPORT-29541	Reduced false positives and improved defect detection with the checker MISRA.VAR.HIDDEN.
SUPPORT-6904, SUPPORT-33605, SUPPORT-32751,	Reduced false positives with the checker UNINIT.STACK.MIGHT.

Number	Description
SUPPORT-6939, 00034442, 00034649, 00034055, SUPPORT-6882	
00027443	Reduced false positives with the checker JD.CAST.KEY.
SUPPORT-27093, SUPPORT-31888, SUPPORT-33709	Reduced false positives with the checker MISRA.LITERAL.NULL.PTR.CONST.2012.
SUPPORT-6623	Reduced false positives with the checker SV.EXPOSE.MUTABLEFIELD.
SUPPORT-27756, SUPPORT-25125	Reduced false positives with the checker RH.LEAK.
SUPPORT-7505	Increased the number of defects detected with the checker MISRA.ONEDEFRULE.VAR.
SUPPORT-32550, SUPPORT-31348	Increased the number of defects detected with the checker DBZ.CONST.
SUPPORT-30885	Created the checker MISRA.PTR.ARITH.NOT_SAME.2008 to support MISRA-C++ Rule 5–0–16.
SUPPORT-30742	Deprecated the checker MISRA.TYPE.NAMECLASH and split it into two checkers: • MISRA.TYPE.NAMECLASH.C.2004 • MISRA.TYPE.NAMECLASH.CPP.2008
SUPPORT-20867	Increased the number of defects detected with the checker SV.TAINTED.BINOP.

Documentation issues

Number	Description
SUPPORT-28351	Improved the access control API examples.
SUPPORT-33822	Added a section to the system requirements that states we support Java 9.
SUPPORT-32851	Removed information about a long-since deprecated feature related to code annotations.
SUPPORT-31312	Improved the documentation for the new Jenkins plugin.
SUPPORT-31829	Fixed an issue where an English title was showing for Java Quality Community checkers in the list of taxonomies.

Fixed issues in Klocwork 2020.1

The following issues were fixed in Klocwork 2020.1.

General issues

Number	Description
SUPPORT-26635	Fixed an issue with the kwprojcopy command related to non-zero exit codes not always being returned when they should be returned.
SUPPORT-22862	Fixed an issue with metrics related to the calculation of the LOC calculation.
SUPPORT-7060, SUPPORT-7084, SUPPORT-28056,	Improved support for nested namespaces.

Number	Description
SUPPORT-27059, SUPPORT-27743	
SUPPORT-7141, SUPPORT-24535, SUPPORT-27096, SUPPORT-26215	Reduced false positives with the checker MISRA.VAR.MIN.VIS.
SUPPORT-25368	Improved support for the Java analysis engine related to null pointer checks.
SUPPORT-27674, SUPPORT-27012	Fixed an issue with the cleanup command in the dbvalidate tool.
SUPPORT-7035	Created a new taxonomy that maps CWE rules to Klocwork Java checkers.
SUPPORT-28331	Fixed an issue with the WebAPI related to metrics outputs for combined a Java/C project.
SUPPORT-22180	Modified the list of logs that are uploaded to the server by the kwadmin load command.
SUPPORT-8478	Improved support for the Microchip MPLAB pic32 compiler.
SUPPORT-27637	Reduced false positives for the checker UNUSED.FUNC.STL_EMPTY.
SUPPORT-7360	Improved the analysis engine for Java related to static import.
SUPPORT-24759	Improved support for Android Pie.
00023389	Reduced false positives for the checker RH.LEAK.
SUPPORT-29114	Updated the Windows installer to slightly modify the tools included in each installation option.
SUPPORT-6987	Fixed an issue with kwbuildproject related to duplicate compile lines with different output directories.
SUPPORT-6909	Tested a previously implemented fix that improved support for Android M and N.
SUPPORT-7054	Reduced false positives for the checker MLK.MUST.
SUPPORT-6903	Improved the knowledge bases shipped with Klocwork related to smart pointers.
00035734	Reduced false positives for the checker INVARIANT_CONDITION.GEN.
SUPPORT-27990	Created new taxonomies for each C/C++, C#, and Java that map Klocwork checkers to the 2019 CWE Top 25 Most Dangerous Software Errors.
00036936, 00036855	Fixed issues with C# analysis related to intervening white spaces, null coalescing operators at the end of a line followed by a period on the next line.
SUPPORT-29062, SUPPORT-30457	Fixed an issue with Visual Studio that caused warnings to be repeated.
SUPPORT-27993	Fixed an issue with Visual Studio related to issue information not being displayed.
SUPPORT-29170	Increased defects detected with the checker MISRA.LITERAL.NULL.INT.
SUPPORT-28780	Reduced false positives with the checker MISRA.ASSIGN.OVERLAP.
SUPPORT-26234	Reduced false positives with the checker UNINIT.CTOR.MUST.
SUPPORT-6996	Improved support for the Tasking Tricore compiler.
SUPPORT-6816	Fixed an issue with the QDP related to missing includes.
SUPPORT-27633	Upgraded the version of Flexnet Publisher we use on Linux and Windows.

Number	Description
SUPPORT-30818, SUPPORT-25844	Fixed an issue related to the integration of the issue management system JIRA.
SUPPORT-26216	Reduced false positives with the checker MISRA.MEMB.NON_CONST.
SUPPORT-27098	Reduced false positives with the checker STRONG.TYPE.JOIN.CONST.
SUPPORT-27337, 00039026	Fixed an issue related to running an analysis in a VPN environment.
SUPPORT-28268	Added support for the Oracle Linux.
SUPPORT-26449	Reduced false positives for the checker MISRA.VAR.NEEDS.CONST.
SUPPORT-8082	Improved support for Wind River Workbench related to support for nested projects.

Documentation issues

Number	Description
SUPPORT-28150	Updated the documents in a zip file of PDFs related to functional safety as the documents from a different release had been included. Also added a test report to the zip file.
SUPPORT-27477	Updated the Japanese version of the ISO/IEC TS 17961 reference list to correct typographical errors.

Limitations

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

Limitations for installation, upgrade, and deployment

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.

Limitation for importing projects with existing reports

If you attempt to import a project with existing reports that use default metric names, you may see unexpected results.

Workaround: When importing a project, ensure that the reports do not use default metric names. If you encounter an error message, you can either delete and re-create the report or edit the metrics.xml file, ensuring that missing or disabled definitions are enabled.

kwcollect fails on tables generated by new analysis engine

The behavior of kwcollect has changed with the introduction of the Klocwork 2018 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 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 platform.

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>
```

Limitations for Mac OS support

- On Mac, clients running Flex Net Publisher version 11.14.0.2 cannot connect to a Klocwork 2020.4 server running Flex Net Publisher 11.14.1.2. For a workaround, see kwlef error states license is not valid.
- Distributed Analysis is not supported.
- For developers, plug-in support is provided for Eclipse and IntelliJ IDEA. If your developers are not using Eclipse or IntelliJ IDEA, they need to use Klocwork Desktop Command Line for C/C++ or Java (kwcheck) or Klocwork Desktop to analyze their code and view detected issues. See Fixing issues before check-in with Klocwork Desktop Analysis.
- System Integrity Protection (SIP) blocks the kwinject command from running properly on Mac OS X 10.10 and later. Kwinject returns the following warning, with error code 1: "`System Integrity Protection is enabled. kwinject cannot inject to process.`" *Workaround:* Disable SIP on the machine running the Klocwork analysis or see Using kwwrap plus kwinject to generate a build specification.

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.

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

Klocwork's C# analysis is supported only on Windows.

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

Feature	Details
Build integration	• Build specification templates
Integration build analysis	• Incremental analysis for kwciagent • Parallel analysis. Parallel analysis is supported for Klocwork 2020.4. • Incremental analysis
Klocwork Static Code Analysis	• "Show implementation", "Show declaration", and Source Cross-Reference
Distributed analysis	• Distributed analysis is not supported for C#.

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

- On-the-fly analysis
- Display of server issues in Visual Studio
- Parallel analysis. Parallel analysis is supported for Klocwork 2020.4.
- Incremental analysis
- File-level analysis in Visual Studio (only solutions and projects can be analyzed)
- Using metric thresholds and knowledge bases

Using metric thresholds and knowledge bases is not supported for C# server build analysis.

Limitations for Klocwork Static Code Analysis

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.

Limitations for Klocwork Desktop Analysis

Analysis is not supported for 'no-resolve' mode in certain scenarios

The "no-resolve" mode was added to support symbolic links to source files on Linux. Symbolic links to directories are not supported.

The Eclipse plug-in supports the "no-resolve" mode only if project is configured to use an external build specification, and that build specification was created by using kwinject with "--no-resolve" option.

For WindRiver Workbench users, you will receive an error message if you attempt to use a project with exterior sources linked to it.

Limitations for the Visual Studio plug-in

'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 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 to Klocwork 2020.1

If you have previously 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 Klocwork Portal. Online help is also available.

Kwvcprojparser not supported for Visual Studio 2017

The kwvcprojparser command is not supported for Visual Studio 2017 projects built from the command line. The kwvcprojparser command is supported on Visual Studio 2012 and 2015; however, it only provides results based on the previous generation (pre-Klocwork 2018) analysis engine.

Workaround: Use the `kwinject` command to create the build specification.

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).

After uninstalling the Klocwork Microsoft Visual Studio extension, the Klocwork help content is not removed

Due to a limitation of the Microsoft VSIX installer, Klocwork help is not removed after uninstalling the plug-in.

Workaround: You can uninstall the help files manually. Go to **Help > Add and Remove Help Content**; In the **Klocwork Inc.** section, click the **Remove** action next to **Klocwork Desktop Plugin**. You can install a future version of the plug-in without issue.

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 attempt to open help for an issue from the Klocwork Issues window in Visual Studio for the Klocwork extension for Visual Studio

If you click on an issue in the Klocwork Issues window and attempt to open the help for it 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.

Duplicate issues when using taxonomy filter

As of Klocwork 2020.4, we've added the ability to filter detected issues by taxonomy. When you are viewing issues grouped by taxonomy, if a detected issue exists in both a filtered and non-filtered taxonomy, both taxonomies are displayed. There is currently no workaround for this use case.

Limitations for VS Code

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, it should be double quotes (same as a command prompt/terminal).

Workaround: Ensure your quoted parameters use double quotations.

Citing defects only works when using a connected project.

When your project is not connected and you attempt to cite defects, nothing will happen.

Workaround: Ensure that all 'Connection Settings' (host, port, SSL, license host, port, and Klocwork project) are set, and the Portal instance the settings are pointing to is up and running.

Limitations for Help in Android Studio and IntelliJ IDEA 2017+

Offline help is not available for Android Studio and IntelliJ IDEA versions 2017 and newer. Offline help is available by using the Klocwork Portal. Online help is also available.

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.

Limitation 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 Klocwork extensibility

C/C++ Path checker compilation makefile compatibility

The makefile generated by kwcreatechecker on Unix systems requires GNU make to build the checker. The default make installed on non-GNU systems such as AIX or Solaris may not compile Klocwork extensions for C/C++. On Windows, the makefile generated by kwcreatechecker requires nmake to build the checker.

Workaround: None.

Using Path checkers on Windows as of Klocwork 2020.3

Due to updating our analysis engine to 64-bit in Klocwork 2020.3, there are several limitations for using PATH checkers on Windows as follows:

- If you use pre-2020.3 path checkers, you must add the `--force-32bit` option to all builds (kwbuildproject, kwcheck, etc.)
- If you use pre-2020.3 path checkers and want to create new path checkers, you need to rebuild all your old checkers in 2020.3 by using a 64-bit compiler, or build your new checkers in a pre-2020.3 version. You cannot mix 32-bit and 64-bit path checkers.
- If you build any 64-bit checkers, they cannot be used in the Visual Studio plugin.
- If you use any 32-bit checkers, they cannot be used 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 Windows as of Klocwork 2020.4

With 2020.4, 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 as of Klocwork 2020.4

With 2020.4, 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.

64-bit KAST checkers and 64-bit AST checkers are currently not supported.

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

You cannot mix 32-bit and 64-bit PATH checkers. If you use pre-2020.4 PATH checkers and want to create new checkers, you must do one of the following:

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

Toll-free: 1.800.487.3217
Direct: 1.613.836.8899
sales@klocwork.com
support@klocwork.com

400 First Avenue North #200, Minneapolis, MN 55401

www.perforce.com

www.perforce.com/products/klocwork