

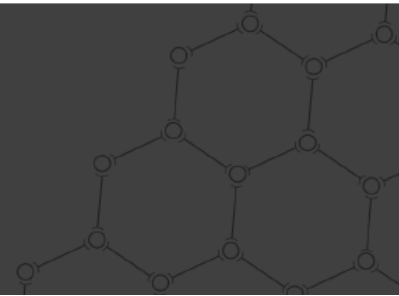


# **Klocwork 2020.1**

Release Notes

PERFORCE

[www.perforce.com](http://www.perforce.com)



# Contents

<b>Release notes.....</b>	<b>3</b>
What's new in Klocwork 2020.1.....	3
Fixed issues in Klocwork 2020.1.....	8
Limitations.....	10

## Release notes

These release notes cover Klocwork 2020.1 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 2020.1. You need a new license to use the latest version of the product. Contact [license@perforce.com](mailto:license@perforce.com) to obtain a new license.

#### Disabled checkers

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

## What's new in Klocwork 2020.1

Here are the highlights for Klocwork 2020.1. If you're upgrading, also see the [Limitations](#) on page 10 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
<a href="#">CWARN.MOVE.CONST</a>	Detects and reports instances of calls to the <code>std::move()</code> method from the C++ Standard Template Libraries (STL) where the argument is <code>const</code> .

### 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.
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
<a href="#">autosar_community_cpp14_19_03</a> and <a href="#">autosar_community_cpp14_19_03_ja</a>	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.

Taxonomy	New/Updated
cert_c_cpp.tconf and cert_c_cpp_ja.tconf	Added references to the following rules: <ul style="list-style-type: none"> <li>• DCL50-CPP</li> <li>• MEM51-CPP</li> </ul>
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: <ul style="list-style-type: none"> <li>• CWE-94</li> <li>• CWE-125</li> <li>• CWE-259</li> <li>• CWE-269</li> <li>• CWE-400</li> <li>• CWE-426</li> <li>• CWE-798</li> </ul> Removed references to the following rules: <ul style="list-style-type: none"> <li>• CWE-195</li> <li>• CWE-247</li> <li>• CWE-366</li> <li>• CWE-466</li> <li>• CWE-479</li> <li>• CWE-488</li> <li>• CWE-562</li> <li>• CWE-587</li> <li>• CWE-684</li> <li>• CWE-764</li> <li>• CWE-770</li> </ul>
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: <ul style="list-style-type: none"> <li>• CWE-200</li> <li>• CWE-259</li> <li>• CWE-295</li> <li>• CWE-426</li> <li>• CWE-772</li> </ul>
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.

Taxonomy	New/Updated
	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.
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 2020.1. You need a new license to use the latest version of the product. Contact [license@perforce.com](mailto:license@perforce.com) to obtain a new license.

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 2020.1

The following issues were fixed in Klocwork 2020.1.

### General issues

Number	Description
SUPPORT-26635	Fixed an issue with the <code>kwprojcopy</code> 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, SUPPORT-27059, SUPPORT-27743	Improved support for nested namespaces.
SUPPORT-7141, SUPPORT-24535, SUPPORT-27096, SUPPORT-26215	Reduced false positives with the checker <code>MISRA.VAR.MIN.VIS</code> .
SUPPORT-25368	Improved support for the Java analysis engine related to null pointer checks.
SUPPORT-27674, SUPPORT-27012	Fixed an issue with the <code>cleanup</code> command in the <code>dbvalidate</code> 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 <code>kwadmin load</code> command.
SUPPORT-8478	Improved support for the Microchip MPLAB pic32 compiler.
SUPPORT-27637	Reduced false positives for the checker <code>UNUSED.FUNC.STL_EMPTY</code> .
SUPPORT-7360	Improved the analysis engine for Java related to static import.



Number	Description
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.
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 this 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.

### Limitations for Mac OS support

- On Mac, clients running Flex Net Publisher version 11.14.0.2 cannot connect to a Klocwork 2020.1 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	<ul style="list-style-type: none"><li><code>kwinject</code> cannot be used to create a build specification for a C# project. Instead, use <code>kwcsprojparser</code>.</li><li><a href="#">Build specification templates</a></li></ul>
Integration build analysis	<ul style="list-style-type: none"><li><a href="#">Mixed-language projects</a> (you need to create one C/C++ project and one C# project)</li><li><a href="#">Parallel analysis</a></li><li><a href="#">Incremental analysis</a></li></ul>
<a href="#">Klocwork Static Code Analysis</a>	<ul style="list-style-type: none"><li>"Show implementation", "Show declaration", and <a href="#">Source Cross-Reference</a></li></ul>
<a href="#">Distributed analysis</a>	<ul style="list-style-type: none"><li>Distributed analysis is not supported for C#.</li></ul>

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

- On-the-fly analysis
- Display of server issues in Visual Studio
- [Parallel analysis](#)
- [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.

## Visual Studio 2017 help

As of Visual Studio 2017, the Help Viewer component is no longer installed by default and must be explicitly selected during installation. If you attempt to install our Klocwork extension for Visual Studio and you do not have this component installed, you will receive an error as our local help is unable to be installed. For more details on this, see [Klocwork Help registration could not acquire the location of the Help Viewer](#).

### 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.

## **Klocwork plug-in for Android Studio installs to unexpected location**

If IntelliJ IDEA 2017 and Android Studio are both installed, and you install the Android Studio plug-in, the IntelliJ IDEA path will be auto-filled instead of the Android Studio path. Klocwork automatically detects your IntelliJ IDEA directory and installs the plug-in to that location.

*Workaround:* If more than one installation directory is detected, you must browse to the preferred location for Android Studio manually.

## **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.

## **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.



Toll-free: 1.800.487.3217

Direct: 1.613.836.8899

[sales@klocwork.com](mailto:sales@klocwork.com)

[support@klocwork.com](mailto:support@klocwork.com)

400 First Avenue North #200, Minneapolis, MN 55401

[www.perforce.com](http://www.perforce.com)

[www.perforce.com/products/klocwork](http://www.perforce.com/products/klocwork)



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 property of Perforce Software, Inc. and is confidential between Perforce Software, Inc. and the client and remains the exclusive property of Perforce Software, Inc. 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. Perforce Software, Inc. does not warrant that this document is error-free.

Klocwork is a registered trademark of Perforce Software, Inc.

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

Copyright notices for third-party software are contained in the file THIRDPARTYLICENSEREADME.txt, located in the Klocwork installation directory.